universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

"Commodity aggregation in a manufacturing company's internal supply network"

verfasst von / submitted by

Ivan Dylyn, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2020 / Vienna, 2020

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 915
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft
Betreut von / Supervisor:	O. Univ.-Prof. Dipl.-Ing. Dr. Richard F. Hartl

Acknowledgements

First of all, I wish to express my deepest gratitude to o. Univ.-Prof. Dipl.-Ing. Dr. Richard F. Hartl and DI Dr. Roland Braune for the opportunity to write this thesis under their supervision at the Department of Business Decisions and Analytics at the University of Vienna. Without their expertise, guidance and assistance, this work would not have been possible.

Secondly, I would like to express my special regards to A.P. for always being by my side when I needed you the most. Your support and encouragement kept me going throughout the process of writing this paper.

Finally, I wish to acknowledge the support and great love of my parents, my mother, Maria, and my father, Dmytro. Thank you for the chance to study at my dream university. You are my role models, and I will always look up to you.

Abstract

This paper studies a capacitated network loading problem of a manufacturer in the automotive industry. The problem is classified as a multiperiod multicommodity capacitated fixed charge network flow problem with integer design variables and intermediate storage. The manufacturer has an internal supply network, where a set of multicommodity demands have to be routed between the nodes represented by production plants, assembly locations and warehouses. Commodities are transported by installing capacitated facilities such as trucks, delivery vans and cars on the arcs (i.e. transport connections). Since there is a large variety of demanded products, commodity aggregation is applied to reduce the problem complexity. The paper analyzes the data generated by Roland Braune's solution of the capacitated network loading problem, where two different aggregation methods are applied, namely, the single-node and unrestricted aggregation procedures. The study points out the inaccuracies of the straightforward aggregation methods and supports the development and improvement of an advanced aggregation procedure that ensures a lower bound.

Abstract German

In dieser Masterarbeit wird ein Netzwerkflussproblem eines Unternehmens im Spezialkraftfahrzeugbau untersucht. Das Problem ist mehrperiodisch und kapazitiv, hat mehrere Produkte, eine Zwischenlagerung, ganzzahlige Design-Variablen, und die Bedienung mehrerer Verbindungen durch ein Transportmittel ist möglich. Der Hersteller verfügt über ein internes Supply-Netzwerk. Es besteht aus Knoten, die Produktionsanlagen, Montagestandorte und Lagerhäuser darstellen, sowie Kanten, die den Transportverbindungen zwischen den Knoten entsprechen. Der Transport von Waren erfolgt durch Installation kapazitiver Einrichtungen wie Lastkraftwagen, Lieferwagen und Autos auf den Kanten. Da es eine große Vielfalt nachgefragter Produkte gibt, wird eine güterbasierte Aggregation angewendet, um die Komplexität des Problems zu verringern. Die Arbeit analysiert die von Roland Braune generierten Daten, wobei zwei verschiedene Aggregationsmethoden angewendet werden, die Einzelknoten- und die uneingeschränkte Aggregation. Die Studie weist auf die Ungenauigkeiten der einfachen Aggregationsmethoden hin und unterstützt die Entwicklung und Verbesserung eines fortschrittlichen Aggregationsverfahrens, das eine Untergrenze gewährleistet.

Table of contents

List of tables

List of figures

1. Introduction

At the end of the 20th century, companies have realized that supply chain management plays a vital role in achieving competitive advantage (Christopher, 2016). Large corporations have started appointing directors to build and oversee the cost-effective supply chain management. Efficient and effective supply chain management increases company's competitiveness and maximizes its economic performance in the long run. It remains to be an essential tool for organizations that aim to successfully face challenges arising from globalization and respond effectively to permanent market changes. A company may have at its disposal a product that delivers the highest value in the eyes of the customer, but if there is no cost-efficient supply chain design behind it, the enterprise will find itself in an inferior business position (Mangan, Lalwani, & Butcher, 2008). Thus, more and more firms encounter a problem, where they need to redesign their supply network, in order to meet company's strategic goals. These goals might be the minimization of the net present value of costs through cost-effective fulfilment of orders, or maximization of the perceived differentiated value through better delivery service (Christopher, 2016). Businesses might want to achieve competitive advantage by providing a better customer service in terms of reduced lead-times, just-in-time delivery and higher reliability. Companies might also want to expand geographically by undertaking mergers and acquisitions or even creating a new operation using a green field investment (Goetschalcks & Fleischmann, 2008). As a result, it has become very common for businesses to have their manufacturing facilities in different geographical locations (Wang, Cheng, & Wang, 2016), which often creates difficult modeling and algorithmic challenges and requires new or upgraded solutions from supply chain managers.

This paper deals with a capacitated network loading problem that is mainly encountered in the telecommunications industry but can also be used for different applications in transportation, distribution and production scheduling. The capacitated network loading is a multiperiod multicommodity fixed charge network design problem with integer design variables and intermediate storage. The fixed charge element means that a fixed cost has to be paid before one can use an arc (Lamar, Sheffi, & Powell, 1990). Although the problem has a very simple structure, which includes a set of nodes, a set of arcs and a demanded flow that is channeled over the network, when capacitated facilities are installed on the arcs, the problem poses significant modeling and algorithmic challenges (Magnanti, Mirchandani, & Vachani, 1995). For instance, in telecommunications private networks are leased to companies that require higher reliability and flexibility for their communications traffic (e.g. audio, video and data transfer) between their offices situated in different geographical locations. With the rapid development of communication technology, the demand for private lines is steadily increasing. Therefore, telecommunications companies are offering digital facilities of different bandwidth to businesses to meet their growing demands.

This paper studies the capacitated network loading problem that is based on a manufacturing company's internal supply network, which operates in a special vehicle automotive industry. Its internal supply network is composed of nodes, which are characterized by production plants, warehouses and assembly locations, and arcs that correspond to transport connections between the nodes (see Figure 1). Both nodes and arcs are capacitated. Nodes have storage and handling capacities, while arcs have space and time-oriented capacities. Another characteristic of this problem is that it is multiperiod. The periods that are considered in the model range from 10 to 18 periods.

Above all, the problem has a large variety of commodities such as different raw materials, small parts and semi-finished products that have to be routed from their origins to their destinations over capacitated edges of the network. The combination of all these factors makes it impossible to obtain the exact solution with CPLEX in reasonable time and even LP relaxation requires excessive computational effort. Therefore, commodity aggregation has to be applied to reduce the complexity of the problem and the amount of data. In this manner, two different aggregation methods, namely, single-node and unrestricted aggregation procedures are examined in this thesis.

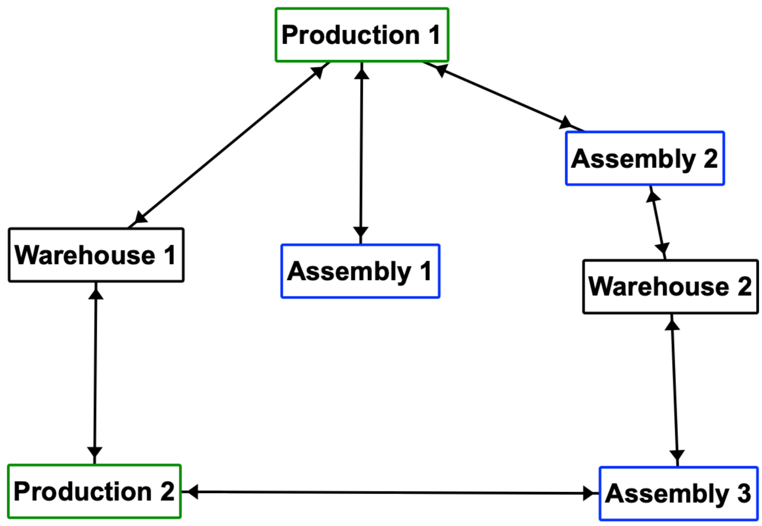

Figure 1: Manufacturing's company internal supply network

To the best of my knowledge commodity aggregation in the context of the internal supply network planning has not been studied yet. It is important to address the effect of the single-node and unrestricted commodity aggregation methods on the capacitated network loading problem. Chouman, Crainic and Gendron (2017) studied a simmilar topic. However, they considered only one ascpect of commodity aggregation, namely, the grouping of commodiites based on their origin node to examine their impact on the strenght of cut-set-based inequaltiies.

The objective of this master thesis is fourfold. First, it aims to present a literature review for the capacitated network loading problem. Second, the commodity aggregation and pegging procedures are introduced. Third, the data generated by Roland Braune's solution of the capacitated network loading problem, where the single-node and unrestricted commodity aggregation procedures were implemented is analyzed. Finally, the thesis aims to support the development of an advanced aggregation procedure that ensures a lower bound.

The remainder of this paper is organized as follows. Section 2 presents the literature review in the domain of capacitated network loading problem and commodity aggregation. Section 3 formulates the problem, and Section 4 covers the implemented commodity aggregation and pegging procedures. Section 5 presents a flow analysis. Finally, in Section 6 all results are summarized, and future research avenues are underlined.

2. Literature review

2.1 Capacitated network loading problem

The thesis considers the capacitated version of the network design models, which is NP-hard and very challenging to solve compared to the uncapacitated network design problems. The latter are easy to solve, because efficient specialized algorithms (e.g. greedy and dual-ascent algorithms) have been developed (Balakrishnan, Magnanti, & Wong, 1989; Magnanti & Wong, 1984). In the capacitated network design problem, some commodities (e.g. products, components, people, data, etc.) are routed over the edges of a network with limited capacitates from their origins to their destinations. The objective is to find a subset of arcs that minimizes the flow costs and fixed costs (Balakrishnan, Magnanti, & Wong, 1989). Consequently, it introduces trade-offs between higher design fixed costs and resulting improvements in the network operation, which decrease flow costs (Magnanti & Wong, 1984). Since higher design fixed costs mean more available capacity on the arcs, which in turn decreases the flow costs, a more efficient network operation can be achieved. Capacitated network generates competition among commodities for the limited quantity that is allowed to pass through the edges. However, there are some heuristics that have been shown to be useful for capacitated problems of reasonable size. For example, drop and add heuristic based on reduced-cost calculations, which determine whether an arc has to be included or excluded from the network after its marginal value had been calculated, while a more complicated procedure applies marginal values of paths in the network (Crainic, Gendreau, & Farvolden, 2000).

One of the prominent examples in the domain of capacitated network loading problems is the model developed by Mirchandani (1989) for the telecommunications industry. The author considers two levels of digital facilities with different capacities, such as DS0 (Digital Signal Level 0) and DS1 (Digital Signal Level 1). DS0 has a throughput capacity of 1 unit and DS1 has a bandwidth, which is equivalent to 24 DS0 facilities. For each of these facilities there is a complex cost structure: for a typical DS1 a provider charges a price that is equal to approximately 10 DS0s. Nevertheless, it is possible to achieve economies of scale. It is therefore of interest to find out what arrangement of capacitated digital facilities between the offices should be rented to satisfy the forecasted demand for traffic at minimum cost. There are usually two types of costs: a per unit flow cost and a fixed cost, which is charged when an edge is used (i.e. also known as design fixed cost). However, in this model the user does not pay for the routing cost. There is only the design fixed cost, which is charged, when a digital transmission facility is loaded. Thereby, there are no additional variable flow costs, and transmission facilities have to be installed on the arcs in a way that the whole demand is met at minimum cost (Mirchandani, 1989).

Magnanti, Mirchandani and Vachani (1995) enhanced and modeled the problem with an additional transmission facility DS3 (Digital Signal Level 3), which was already in use, when their paper was being written. In their study DS3 is able to transmit at the rate of 28 DS1 facilities. In this manner, they modeled the problem with two types of digital transmission facilities – DS0 and DS1 or DS1 and DS3. The authors named this problem the two-facility loading problem (TFLP). In TFLP $\boldsymbol{N}$ is a set of nodes, $\boldsymbol{A}$ is a set of arcs and $\boldsymbol{K}$ is a set of commodities. A commodity $\boldsymbol{k}$ has an origin $\boldsymbol{O(k)}$ and a destination $\boldsymbol{D(k)}$ with demand $\boldsymbol{d_k}$. $\boldsymbol{C}$ denotes the capacity of facilities. There are two types of decision variables: $\boldsymbol{x_{ij}}$ and $\boldsymbol{y_{ij}}$. The former design variable $\boldsymbol{x_{ij}}$ defines the number of low capacity facilities installed on the arcs. A low capacity facility has a capacity of 1. The latter decision variable $\boldsymbol{y_{ij}}$ corresponds to the number of loaded high capacity digital transmission facilities on the arcs. A high capacity facility has a capacity of $\boldsymbol{C}$, where $C = 24$, when DS0 and DS1 are leased, and $C = 28$, when DS1 and DS2 are leased. To the flow variables belongs $\boldsymbol{f^k_{ij}}$ which determines the flow of commodity $\boldsymbol{k}$ on arc $\boldsymbol{\{i, j\}}$. Finally, $\boldsymbol{a_{ij}}$ and $\boldsymbol{b_{ij}}$ are cost coefficients that are incurred when the facilities are being installed.

The following objective function is formulated to minimize the total cost for installing all the required facilities to meet the traffic demand (Magnanti, Mirchandani, & Vachani, 1995):

$$\min \sum_{(i,j)\in A} (a_{ij} * x_{ij} + b_{ij} y_{ij})$$

It is subjected to the following constraints:

$$\sum_{j\in N} f^k_{ji} - \sum_{j\in N} f^k_{ij} = \begin{cases} -d_k \; if \; i = O(k) \\ d_k \; if \; i = D(k) \; \forall \, i \in N, k \in K \\ 0 \; otherwise \end{cases} \quad (1)$$

$$\sum_{k\in K}\left(f_{ji}^k + f_{ij}^k\right) \leq x_{ij} + Cy_{ij}, \forall\,(i,j) \in A \quad (2)$$

$$x_{ij}, y_{ij} \geq 0 \; and\; integer\;, \forall\,(i,j) \in A \quad (3)$$

$$f_{ij}^k, f_{ji}^k \geq 0, \forall\,(i,j) \in A, k \in K \quad (4)$$

Constraint (1) is the flow conservation constraint for each commodity at each node, and constraint (2) ensures that the total flow, which occurs in both directions, does not exceed the capacity of the installed facility. Constraint (3) implies that the design variables x_{ij} and y_{ij} are of integer type and non-negative. Finally, constraint (4) ensures that the flow variable f^k_{ij} is a non-negative continuous variable. The authors propose a Lagrangian relaxation strategy and a cutting plane approach to solve the mixed integer problem (Gavish & Neuman, 1989). Magnanti, Mirchandani and Vachani (1995) identified a set of inequalities (i.e. the arc residual capacity, the cutset and the 3-partition inequalities) that ensure a lower bound that is equal to the Lagrangian lower bound and reduce the limits of linear relaxation in approximating the mixed integer program by an average of 8% to 25%. The study concluded that the cutting plane approach is an effective method for capacitated network design problems and can be certainly applied in the telecommunications industry.

The above-mentioned formulation is based on the telecommunications industry, where no additional variable flow costs and no flow variables are considered. However, the network loading problem can be also used for different applications in transportation planning. In this case, data traffic corresponds to the flow of commodities and the capacitated digital transmission facilities are represented by different modes of transportation with different capacities (e.g. trucks, delivery vans and cars). In addition to this, one should also account for the assignment of transportation facilities to routes, handling and storing of commodities. Thereby, Gendron, Crainic and Frangioni (1999) introduce arc-based, path-based and cut-based formulations for the multicommodity capacitated fixed charge network design problem. They present three solution methods which include simplex-based cutting plane approach, Lagrangean relaxation approach and heuristics (Crainic, Gendreau, & Farvolden, 2000). The authors focus on the arc-based formulation and their results show that a precise combination of cutting planes, Lagrangean relaxation methods and sophisticated heuristics is necessary to solve the problem (Balas, Ceria, & Cornuejols, 1993).

Gendron, Crainic and Frangioni (1999) consider a directed graph $G = (N, A)$, where N is a set of nodes and E is a set of design arcs in the network. K denotes the set of commodities with its origins $O(k)$, destinations $D(k)$ and transshipment nodes $T(k)$. Each origin and demand node should have a demand and supply that is larger than zero for commodity $\boldsymbol{k}$. They have all fixed capacity $\boldsymbol{u_{ij}}$. Two types of costs are associated with this problem: the flow and the design costs. The flow cost for each commodity $\boldsymbol{k}$ on arc ***(i, j)*** is represented as $\boldsymbol{c^k_{ij}}$. The design cost $\boldsymbol{f_{ij}}$ has to be paid when flow goes through a design arc ***(i, j)***. In addition to this, $\boldsymbol{b^k_{ij}}$ is an upper bound on the amount of commodity $\boldsymbol{k}$ that can flow through arc ***(i, j)***. Moreover, every commodity $\boldsymbol{k}$ has a weight $\boldsymbol{e^k_{ij}}$, and there is a capacity $\boldsymbol{v_{ij}}$ on each arc. In the model the authors use continuous flow variables $\boldsymbol{x^k_{ij}}$ for every commodity $\boldsymbol{k}$ on every arc ***(i, j)*** and integer design variables $\boldsymbol{y_{ij}}$, which indicate the number of facilities built on each arc ***(i, j)***.

The objective of the arc-based model is to minimize the sum of flow and design costs, and it is formulated for the multicommodity capacitated fixed-charge problem in the following way (Gendron, Crainic, & Frangioni, 1999):

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c^k_{ij} x^k_{ij} + \sum_{(i,j) \in E} f_{ij}\, y_{ij}$$

It is subjected to the following constraints:

$$\sum_{j \in N^{+}+(i)} x^k_{ij} - \sum_{j \in N^{-}(i)} x^k_{ji} = \begin{cases} o^k_i \text{ , } \forall i \in O(k) \\ -d^k_i \text{ , } \forall i \in D(K), i \in N, k \in K \\ 0, \ \forall \in T(k) \end{cases} \quad (1)$$

$$0 \leq x^k_{ij} \leq b^k_{ij} \text{ , } \forall (i,j) \in A, k \in K \quad (2)$$

$$\sum_{k \in K} x^k_{ij} \leq u_{ij}, \forall (i,j) \in A \quad (3)$$

$$\sum_{k \in K} x^k_{ij} \leq u_{ij} y_{ij}, \forall (i,j) \in E \quad (4)$$

$$x^k_{ij} \leq b^k_{ij} y_{ij}, \forall (i,j) \in E, k \in K \quad (5)$$

$$0 \leq y_{ij} \leq 1, \forall (i,j) \in E \quad (6)$$

$$y_{ij} \ integer, \forall (i,j) \in E \quad (7)$$

Constraint (1) is the flow conservation constraint for each commodity at each node. Constraint (2) ensures that the continuous total flow variable is non-negative and does not exceed the capacity of arc ***(i, j)***. Constraint (3) guarantees that the sum of flows of all commodities does not surpass the capacity of the installed facility on design arc ***(i, j)***.

Constraint (4) ensures that the sum of flows of all commodities does not exceed the capacity of the installed capacity and makes the flow equal to 0, if the facility is not installed on design arc ***(i, j)***. Constraint (5) is equivalent to constraint (4) for relaxation purposes, because they improve considerably the lower bounds. It also ensures that the flow of commodity ***k*** does not surpass the arc capacity, and it forces the flow to be equal to 0, if the arc is not used for commodity routing. Constraint (6) restraints the design variables to assume the values between 0 and 1. Finally, constraint (7) ensures that the design variables are of integer type. Gendron, Crainic and Frangioni (1999) apply two continuous and five Lagrangean relaxations. The continuous relaxations are responsible for lifting the integrality constraints, whereas the Lagrangean relaxations are obtained by dualization of constraints, while following the goal of minimization of the number of dualized constraints and production of an easily solvable Lagrangean subproblem (Gavish, 1986).

Aforementioned, there are three different formulations for the capacitated network loading problem. The arc-based formulation was shown in the previous section. The path-based formulation is applied when an optimizer wants the flow from origin to destination to have a single path (Barahona, 1996). On the other hand, when there are no additional variable flow costs to account for, which is often the case in telecommunications, the cut-based formulation is used. Although path-based and arc-based formulations of the multicommodity fixed-charge problem provide similar results in terms of their strong LP relaxations, the arc-based LP relaxation delivers a tighter bound (Gendron, Crainic, & Frangioni, 1999). In the case of cut-based formulations, survival networks have to be designed, with a prespecified number of edge-disjoint paths between each pair of nodes (Grötschel, Monma, & Stoer, 1995).

Lee, Medhi and Strand (1989) introduce a nonlinear programming path-based formulation for the multiperiod multicommodity network loading problem and implement a nonlinear optimization algorithm – a variant of the Frank-Wolfe algorithm. This algorithm allowed to generate quick near optimal solutions. It is very practical, because under uncertain demand and cost forecasts, the Frank-Wolfe algorithm delivers good solutions (Lee, Medhi, & Strand, 1989). Crainic, Gendreau and Farvolden (2000) develop a new efficient Tabu Search procedure, which examines the space of the continuous flow variables by linking pivot moves with column generation, while assessing the actual mixed integer objective, to solve large realistic multicommodity capaciated fixed charge network flow problems. The authors state that their method delivers good feasible solutions for problems with many commodities (Crainic, Gendreau, & Farvolden, 2000). Avella, Mattia and Sassano (2007) presented the new class of tight metric inequalities and implemented separation algorithms and a cutting plane algorithm. Their results for the LP-based upper bound heuristic appeared to be effective for symmetic and assymetic Norwegian instances. Babonneau and Vial (2010) suggest a new algorithm to solve the network loading problem, which is similar to the Benders decomposition scheme. Unlike other approaches, their method does not use the original mixed integer programming formulation, and the integer programming solver does not generate the best integer point. On the contrary, it is an iterative method, which applies metric inequalities to approximate the set of feasible capacities and uses the integer programming solver to select an integer solution that is closest to the best-known integer solution (Babonneau & Vial, 2010). Furthermore, this solver generates integer solutions in the space of the capacity variables. Metric inequalities are obtained by solving nonlinear multicommodity flow problems. The authors claim that their algorithm produces good solutions, which is unsophisticated and can be implemented for many mixed integer programming problems (Babonneau & Vial, 2010). Ljubic, Putz and Gonzalez (2012) study the single source variant of the network loading problem, for which they develop a Multi-Cabletype Multi-Commodity Flow (MMCF) formulation incorporated into a branch-and-cut framework. The authors show that Benders' inequalities, used to project out the flow variables while keeping strong lower bounds, are able to improve gaps. Fragkos, Cordeau and Jans (2017) study a multiperiod extension of the traditional single period multicommodity network design problem and consider its capacitated and uncapacitated variants. They show that for capacitated problems the Lagrange relaxation, which incorporates a series of local improvement heuristics, scales very well with problem size. On the other hand, for uncapacitated problems, a variety of Benders decomposition schemes is used, which breaks down the multiperiod multicommodity network design model into single-period shortest path subproblem per period and per commodity (Fragkos, Cordeau, & Jans, 2017). Mejri, Layeb and Mansour (2019) apllied a path-based formulation to develop a tailored Benders decomposition scheme in combination with column generation procedure. Furthermore, through a max-cut-like integer programming model the authors use efficient cutset inequalitites, which are dynamically derived and accelerate the solution of the problem. Their results suggest that the imroved procedure performs better than the basic version of the algorithm (Mejri, Layeb, & Zeghal, 2019).

Adequate formulations and efficient approaches have to be developed, in order to determine desirable supply chain configurations that meet strategic goals (Goetschalcks & Fleischmann, 2008). Companies should be able to make decisions about the assignment of manufacturing and distribution capacities, and the allocation of products to established facilities to meet the customer demand. It is suggested to incorporate into the network design model two planning levels: strategic structural decisions and master planning decisions (see Figure 2). The first level deals with network configuration such as locations of the facilities, their capacities and product allocation, and it encompasses binary structural variables. The second planning level concerns flows of goods in the network and includes continuous flow variables. Master planning decisions are constrained by the strategic structural decisions, and the organizational objectives (e.g. net present value of costs or profit, customer service, etc.) are influenced by master planning's financial variables and structural decisions on investment (Goetschalcks & Fleischmann, 2008).

Goetschalcks and Fleischmann (2008) claim that it is almost impossible to determine the optimal solution for a supply chain configuration. Namely, the forecasted data which is used in the model is highly uncertain. Further, the supply chain configuration entails multiple objectives, and the objectives such as customer service, risk and flexibility are hard to quantify. A solution to this problem may lie in the generation of different scenarios (e.g. risk could be quantified, if there are known probabilities for scenarios, but it is almost always not the case). In addition to this, it might not be possible to consider all aspects of the problem. For example, one could quantify flexibility by estimating variable production volume for different demand levels of a certain product. However, it is not yet clear how to quantify the tradeoff between efficiency and flexibility, when taking into account an installment of machine for general and dedicated purpose.

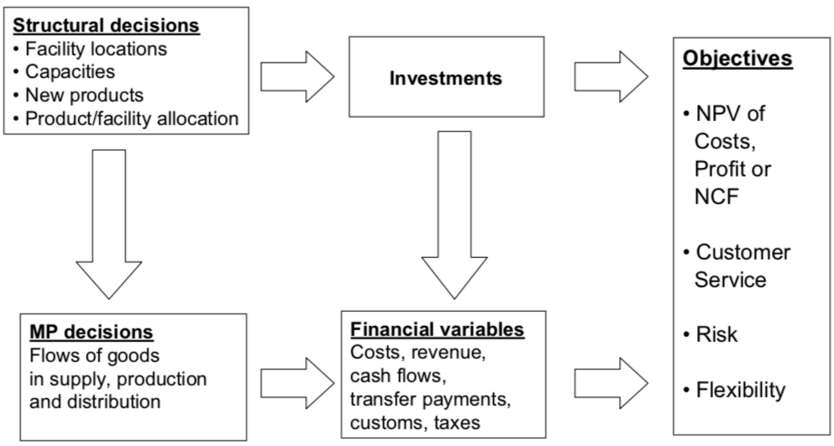

Figure 2: Strategic and master planning decisions (Goetschalcks & Fleischmann, 2008)

Chauhan, Frayret and LeBel (2011) study the supply network planning in the forest industry, namely, the procurement planning, where multiple origins of raw material and multiple demand destinations are considered. The origins are represented by the stands, while the destinations are represented by the mills. The stands produce different kinds of logs, which are demanded by the mills. The authors also solve the bucking problem, which they integrate into procurement planning model. The objective of the procurement model is to minimize the harvesting and transportation costs, while satisfying the demand from several mills (Chauhan, Frayret, & LeBel, 2011). This is a non-trivial problem, due to the fact that harvesting production cost should account for the felling, bucking, sorting and transportation to roadside operations. In order to account for these characteristics, the authors use a decomposition approach, based on column generation, which can be started with a few feasible production vectors, that indicate a harvesting and transportation plan for each stand. The columns can be generated dynamically by applying the latest computation results. In order to generate new columns, a pricing problem has to be solved iteratively for each stand until there is no more new columns to add, which would improve the solution. Since the pricing problem is an integer programming problem and has several possible production scenarios, the authors propose to fix the production scenario and use a heuristic algorithm, which incorporates the bucking pattern algorithm and the pattern generation algorithm to solve the pricing problem iteratively (Chauhan, Frayret, & LeBel, 2011).

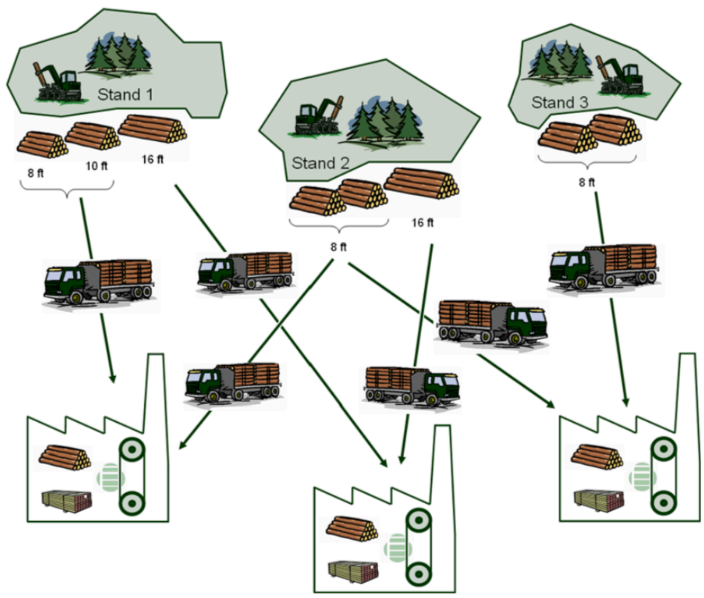

Figure 3: The procurement problem (Chauhan, Frayret, & LeBel, 2011)

Wang, Cheng and Wang (2016) deal with a similar supply network planning problem. They suggest a flexible supply network planning (FSNP) model based on integer linear programming to improve service level in the memory module industry. The model aims to improve order-to-delivery time through order allocation among multiple production facilities and through direct shipments. The problem considers many aspects of the memory module industry. For example, it accounts for multilevel manufacturing process, multisite order allocation and multiple-to-multiple product substitution structures. The network is considered to be flexible, because normal and direct shipments are employed. To develop the FSNP model, memory module industry is divided into three main production stages (see Figure 4).

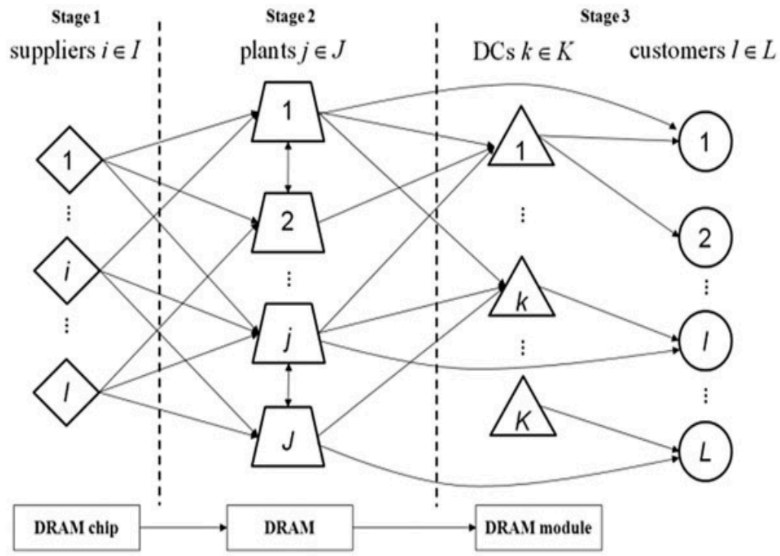

Figure 4: Memory module industry's supply network (Wang, Cheng, & Wang, 2016)

In the first stage, raw materials are supplied to the production sites. In the second stage, supplied raw materials are transformed into semi-finished products. At this stage, in order to decrease the order-to-delivery time, producers can manufacture semi-ready goods based on the demand forecasts. During the last stage, semi-finished products are supplied to the corresponding production sites, where they are used to assemble the final product, which are then normally shipped to the customer (Wang, Cheng, & Wang, 2016). As it can be seen in Figure 4, the manufacturing facilities are able to employ direct shipments, in order to decrease inventory holing costs at DCs and to be able to react to demand changes, while taking into account capacity and inventory constraints and delivery due dates. The FSNP is objected to minimize the holding, transportation, reallocation and penalty costs for the manufacturing plants and the distribution centers respectively. The problem is solved using

LINGO 10.0 extend, which improves the order allocation plans for all manufacturing sites and distribution centers. Compared to the planning method involving just normal shipments in terms of product shortages and cost, the authors show that the FSNP model provides better order allocation, improves order due date, and decreases inventory and transportation costs. In addition to this, the study shows that the companies choose routes of order fulfilment not only based on transportation cost but also based on the unit holding cost at the manufacturing sites and distribution centers (Wang, Cheng, & Wang, 2016).

2.2 Aggregation methods

A strong correlation exists between model complexity and computational run time. The more decisions need to be mapped and the more accurate the model is, the longer it takes for a computer to solve it. Not only run time increases, but also the costs for collecting additional data. Usually, the exact solution is not possible within a reasonable time, especially, when a manager aims to construct an accurate model. In this manner, there is a trade-off between accuracy and computational effort (Rohde & Wagner, 2008). To tackle this challenge, one could consider quantity decisions such as production and transportation quantities and keep them continuous by prohibiting integer values. Similarly, decisions about capacity can be avoided, if it is only allowed to utilize the whole capacity, excluding possibilities for underusage and excessive usage, for example, increasing capacity by working more hours (Rohde & Wagner, 2008).

Another way to decrease complexity of the model is to apply aggregation. Aggregation is used to reduce problem complexity and diminish uncertainty in demand forecasts. “Aggregation is the reasonable grouping and consolidation of time, decision variables and data to achieve complexity reduction for the model and the amount of data” (Rohde & Wagner, 2008, p. 172; Stadtler, 1988). One can aggregate products, resources and time (Stadtler, 2008). For example, weeks could be considered instead of days, end products instead of their variants and capacity groups instead of resources (Fleischmann, Meyr, & Wagner, 2008). It can also be distinguished between three aggregation alternatives, which are usually applied at the same time: aggregation of time, decision variables and data (Shapiro, 1998). The distinction between aggregation of products, customers and suppliers will be underlined later.

Aggregation of time occurs when smaller periods are grouped together into one big period. Usually, it is not reasonable to consider daily time frames for Master Planning, since it can be almost impossible to manage the data if it is to be collected, for example, for one year. In this manner, the smaller time buckets are aggregated into weeks or months. There might be a problem, when different time intervals are used on different planning levels (Rohde & Wagner, 2008). To resolve it, planning horizons on lower planning levels may be selected (see Figure 5).

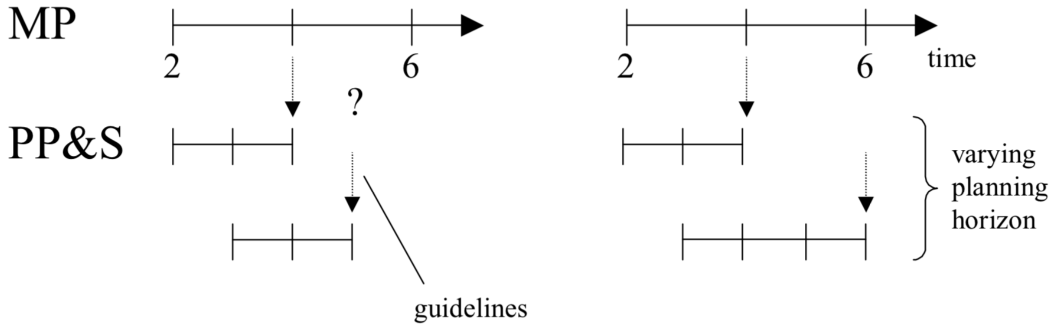

Figure 5: Aggregation of time resources (Rohde & Wagner, 2008)

Aggregation of decision variables occurs when production quantitates are merged together. It is the most difficult type of aggregation, because it requires a proficient understanding of company's product lines (Shapiro, 1998). In the production planning, for example, it is suggested to aggregate finished products based on their similar holding cost per unit per period, direct production costs, seasonality and productivity (i.e. number of end products that can be produced per unit of time). In addition to this, products that share similar set up costs and require the same quantity of the same intermediate assemblies (i.e. bill of materials (BOM)) form together one product family (Bitran, Haas, & Hax, 1982).

Another method of product aggregation can be seen in Figure 6. In the beverage industry final products can be grouped together based on their size (e.g. glass of 16 oz., 32 oz. and 48 oz.), packaging (e.g. glass, can), lifestyle (e.g. regular, diet), taste (e.g. cola, ginger ale, root beer) and group (e.g. soft drinks, ice teas, juices) (Kilger & Wagner, 2008).

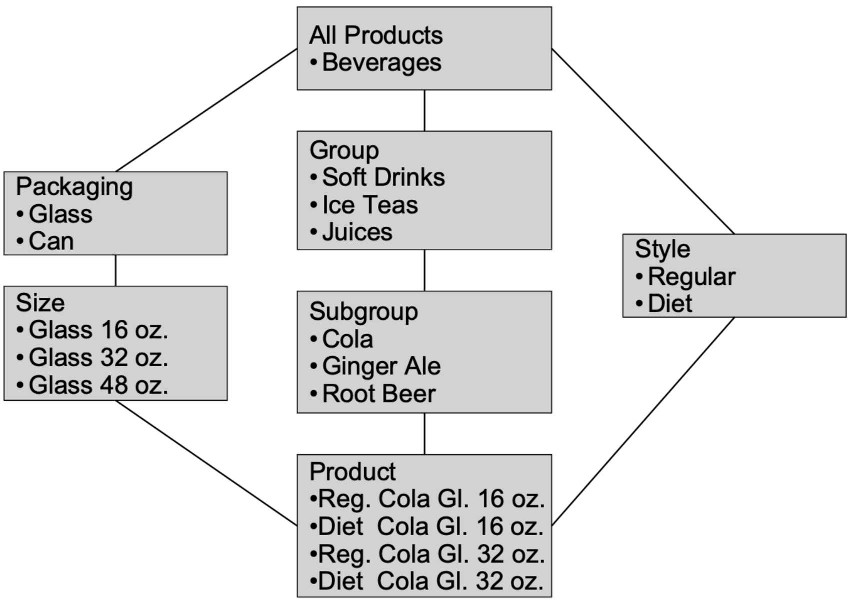

Figure 6: Product aggregation in the beverage industry (Kilger & Wagner, 2008)

There are two drawbacks that can be underlined here. The first one is that it is not clear what is meant by similarity, and the second is what happens when two end products with dissimilar sub-products are grouped together. Often, supply chain managers undertake aggregation of commodities for demand analysis. Commodity aggregation may be used at the data collection or estimation level. An implicit assumption is made that the collected data of different commodities can be perfectly substituted. In other words, it is assumed that perfect substitutability is given. For example, when customers view one liter of petrol the same way as one liter of milk, hence, both items can be added together. However, special methods are required to ensure perfect substitutability for the analysis of demand systems (Park & Thurman, 2013). Figure 7 illustrates how aggregation of intermediate products poses some challenges for the supply chain manager. There are three finished products P1, P2 and P3. P1 requires 1 unit of part A, 2 units of part B and 1 unit of part C. P2 requires 1 unit of part A and 2 units of part B. Lastly, P3 requires 1 unit of part A and 1 unit of part D. When we integrate P1 and P2, we create a new aggregated commodity P 1/2, which is composed of part types AB (i.e. parts A and B are aggregated to component AB) and C. Let’s assume that P1 and P2 have a demand share of 1/4 and 3/4 respectively. BOM aggregation illustrates that P 1/2 requires 1 unit of part type AB (i.e. 3/4 + 1/4 = 1) and 1/4 of component C. Production of part type AB requires 1 unit of part A and 2 units of part B. Therefore, a question arises, how much of the aggregated part type AB is required to manufacture P3 (Rohde & Wagner, 2008). An answer to this question may lie in the 80/20 rule developed by Jeremy Shapiro, which implies that 20% of products with the lowest revenue are responsible for most of product variety, and therefore they should be aggregated into smaller number of product families, while the other products that make up 80% of revenue should be consolidated very judiciously (Shapiro, 1998).

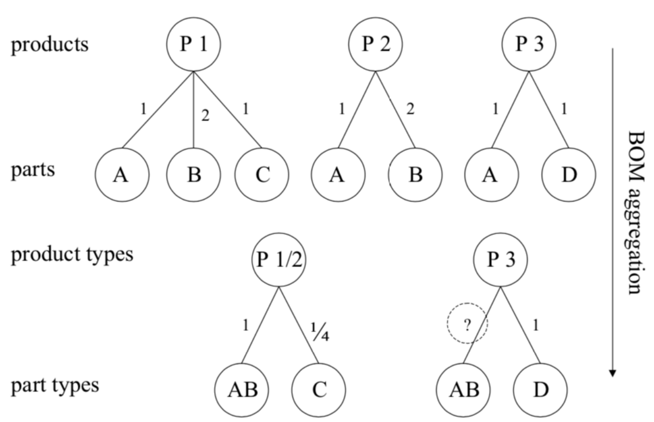

Figure 7: Aggregation of intermediate products (Rohde & Wagner, 2008)

Aggregation of data occurs when, for instance, transport capacities, production capacities, inventory capacities, purchasing bounds and demand data are consolidated (Rohde & Wagner, 2008). Aggregation of time, decision variables and data should be executed concurrently, since there are many interconnections between them. An alternative way to lower the complexity of the model is to perform supplier and customer aggregation. Both aggregations are implemented in the similar manner. Depending on the company, the industry and the supply chain analysis itself, different aggregation methods for customers and suppliers can be used. However, at large, companies should group their small customers or suppliers with respect to their geographical locations together, while keeping their large customers or suppliers as distinct entities. In Figure 8 it can be seen that customers can be grouped together based on their geographical location. For example, customers can be aggregated, when they come from the same region, area or they receive their orders from the same distribution center, manufacturing facility, etc. (Kilger & Wagner, 2008).

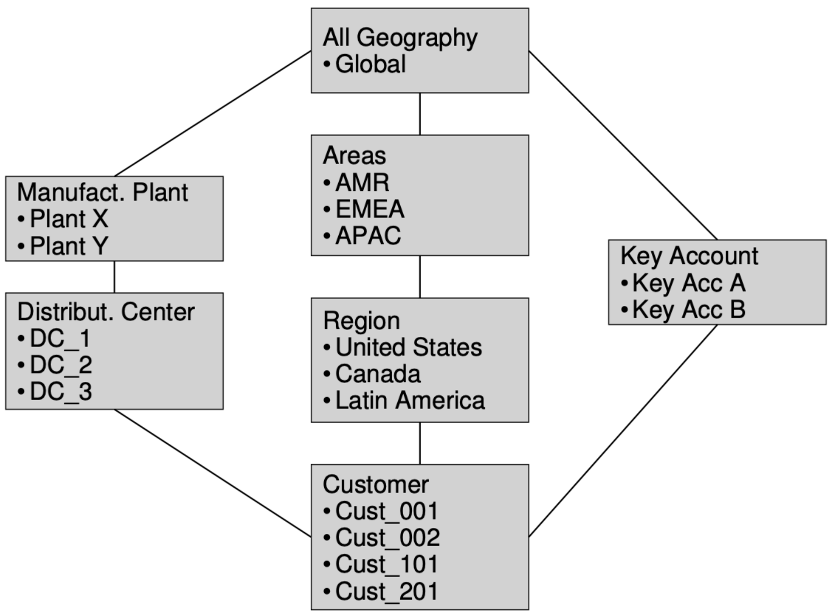

Figure 8: Aggregation of suppliers and customers (Kilger & Wagner, 2008)

To the best of my knowledge, not much research has been done on commodity aggregation. Therefore, it is important to introduce some influential papers that have effectively implemented this technique. Chouman, Crainic and Gendron (2017) improve the mixed-integer programming formulation of the multicommodity capacitated fixed-charge network design problem (MCND) by incorporating valid inequalities into a cutting-plane algorithm and implementing commodity aggregation. The objective function of the problem is to satisfy the demand at minimum cost, where costs are represented by transportation and fixed-charge design costs. Their cutting-plane method is based on five classes of known valid inequalities: the strong, cover, minimum cardinality, flow cover, and flow pack inequalities (Chouman, Crainic, & Gendron, 2017). The first valid inequality is especially useful when a disaggregated representation of the commodities is chosen, and the last four inequalities are expressed in terms of network cut sets, where the set of nodes is partitioned into two subsets. The authors state that commodity aggregation facilitates the performance of the implemented inequalities. The study applies aggregated commodity representation, where all commodities with the same origin are grouped together. Commodity aggregation delivers the same sets of feasible and optimal solutions as the disaggregated commodity representation, because in the case of MCND there are no commodities that depend on costs. The advantage of the aggregated commodity representation is the reduction in the size of the model (i.e. it is reduced by a factor of $|V|$). On the other hand, it is claimed that some valid inequalities are stronger in the case of the disaggregated commodity representation (Chouman, Crainic, & Gendron, 2017).

The authors show in their computational study the strength of the disaggregated commodity representation, when combined with dynamic generation of strong inequalities. Furthermore, the disaggregated commodity representation is more suitable for large-scale instances (i.e. more than 100 commodities) than the aggregated commodity representation. However, the latter performs better with few commodities (i.e. approximately 10) (Chouman, Crainic, & Gendron, 2017).

Burchett and Richard (2015) implement commodity aggregation, in order to calculate strong valid inequalities for the single-commodity variable upper bound flow models (Burchett & Richard, 2015). They generalize their model by establishing some relations between facets of single-and multi-commodity models and introducing a new family of inequalities (Crainic, Frangioni, & Gendron, 2001). Newman and Kuchta (2007) implement a heuristic, where they simplify the formulation of iron ore production schedule that has many binary variables, by aggregating time periods. The authors then use this information to solve the original model. Their results show that this procedure is capable to deliver good soulutions in an acceptable computation time (Newman & Kuchta, 2007). Boland et al. (2013) implement time aggregation for the network design model of coal delivery with time constraints, where they dynamically adapt aggregation to solve complex problems. Furthermore, an iterative disaggregation method is developed to find a time aggregation that ensures an optimal solution. In this manner, the authors reduce the complexity of the problem and calculate the lower bound for the original problem.

Another paper applies commodity aggregation to solve the Minimum Cost Capacity Installation (MCCI) problem, where the objective is to minimize the total cost of installing the required capacity on the arcs, and where a set of traffic demands is shipped between node pairs (Bienstock, Chopra, Guenluek, & Tsai, 1998). The authors introduce two approaches. The first approach is based on metric inequalities, which has in its problem formulation $|A|$ variables. The second approach is based on the idea of aggregated multicommodity flow formulation with $|V||A|$ variables. In the paper, commodities are aggregated based on their source node: all commodities that are transported from the same source form a new commodity. A subset of products becomes one commodity, whose supply equates to the total supply of its comprised products. As a result, the problem size can be reduced with at most $|V|$commodities, while keeping the value of linear programming relaxation. Although the disaggregated version delivers stronger cutting plane, the aggregation facilitates the solution of the problem. Furthermore, strong valid inequalities for the multicommodity variable upper bound design flow model are calculated with the help of commodity aggregation (Bienstock, Chopra, Guenluek, & Tsai, 1998).

In the end, it is important to underline that no matter which aggregation type is used, it should be strictly avoided that some data characteristics get lost in the aggregation process. In the following section the mixed-integer programming formulation for the company's internal supply network will be presented, and in Section 4 the implemented commodity aggregation and pegging procedures in the context of capacitated network loading problem will be portrayed.

3. The company's supply network – a formal statement

Aforementioned, in this paper, we study the multiperiod multicommodity fixed charge network design problem with integer design variables and intermediate storage. The internal supply network of the manufacturing company can be depicted by a directed graph $\boldsymbol{G(V, E)}$, where $\boldsymbol{V}$ denotes the set of nodes and $\boldsymbol{E}$ is the set of arcs in the flow network. The nodes represent production plants, warehouses and assembly locations, while arcs correspond to transport connections between the nodes. Furthermore, $\boldsymbol{V^{in}(u)}$ denotes the set of nodes that are heads of entering edges of node $\boldsymbol{u}$ and $\boldsymbol{V^{out}(u)}$ denotes the set of nodes that are tails of leaving edges of node $\boldsymbol{u}$. Both, nodes and arcs are capacitated. Nodes have storage and handling capacity, while arcs have space and time-oriented capacities. Namely, $\boldsymbol{a_u}$ denotes the minimum requirement/inventory level at node $\boldsymbol{u}$ in shipping or handling units, whereas $\boldsymbol{b_u}$ is the maximum storage capacity at node $\boldsymbol{u}$ in shipping or handling units. The initial inventories are denoted by $\boldsymbol{F^0_{uc}}$ for commodity $\boldsymbol{c}$ at node $\boldsymbol{u}$. In addition to this $\boldsymbol{p_u}$ applies the processing capacity of node $\boldsymbol{u}$ in shipping or handling units. The space-oriented capacity restricts the number of European pallets that fit into a vehicle. For example, the model utilizes commodity weights $\boldsymbol{e_c}$ that determine the size requirements, where weight $\boldsymbol{e_c}$ of a commodity $\boldsymbol{c}$ gives the portion of a pallet that it occupies. In this manner, $\boldsymbol{K_{uvm}}$ denotes the space capacity of one transportation facility unit of mode $\boldsymbol{m}$ between node $\boldsymbol{u}$ and node $\boldsymbol{v}$. As for the time-oriented capacities, they specify the daily availability of facilities. $\boldsymbol{ZK_{mt}}$ indicates the maximum time-based capacity of mode $\boldsymbol{m}$ at time $\boldsymbol{t}$ and $\boldsymbol{ZB_{uvm}}$ is the time required for one tour on arc $\boldsymbol{(u, v)}$ in mode $\boldsymbol{m}$. For instance, the operation time of vehicles can be limited to 8 hours per day.

The model considers intermediate storage. It is possible to store commodities in production plants, warehouses and assembly locations. One can distinguish between three types of commodities: raw materials, small parts and semi-finished products, and there is a large variety of these demanded commodities. Products can be transported from the source nodes $\boldsymbol{q_c}$ to their sink nodes $\boldsymbol{o_c}$ by one or more trucks, delivery vans and cars, where the travel times are in maximum hourly range. Moreover, commodities are produced (> 0) or consumed (< 0) $\boldsymbol{s_{uct}}$, at node $\boldsymbol{u}$ at time $\boldsymbol{t}$.

Another characteristic of this problem is that it is multiperiod. It ranges from 10 to 18 periods. The problem incurs the holding cost per shipping or handling unit at node $\boldsymbol{u}$, which is represented by $\boldsymbol{\alpha_u}$. Furthermore, $\boldsymbol{\beta_u}$ denotes the processing cost per shipping or handling unit at node $\boldsymbol{u}$ for outbound flow, while $\boldsymbol{\gamma_m^U}$ indicates the variable transportation cost of mode $\boldsymbol{m}$, and $\boldsymbol{\gamma_m^F}$ denotes the time dependent renting cost for a transportation mode $\boldsymbol{m}$.

The goal of the model is to determine material flows on strategic and tactical level. Therefore, it is of interest to calculate the number of tours required to channel all commodities from their origin to their destination, the number of facilities (i.e. vehicles) needed in each period to complete the tours, and the inventory levels for all commodities and at each node in every period. Thereby, the model accounts for holding, handling, transportation and renting costs, and the objective is to minimize the overall logics costs while satisfying the demand.

Hence, given the directed graph ***G (V, E)***, a set of commodities ***C*** and a time horizon length ***T***, the multicommodity capacitated fixed charge network flow model is objected to minimize the total cost and the objective function is formulated as:

$$min \sum_{u} \sum_{c} \sum_{t} f_{uct} * e_c * \alpha_u + \sum_{(u,v)\in E} \sum_{m} \sum_{c} \sum_{t} x_{uvmct} * e_c * (\beta_u + \gamma_m^U)$$
$$+ \sum_{(u,v)\in E} \sum_{m} \sum_{t} y_{uvmt} * ZB_{uvm} * \gamma_m^F$$

where $\boldsymbol{f_{uct}}$ denotes the stock level of commodity ***c*** at node ***u*** in time period ***t***; $\boldsymbol{x_{uvmct}}$ the amount of flow (continuous) of commodity ***c*** that is routed from node ***u*** to node ***v*** in mode ***m*** at time ***t***; $\boldsymbol{y_{uvmt}}$ the number of transport units or facilities (discrete) used on arc ***(u, v)*** in mode ***m*** at time ***t.***

The objective function is subjected to the following constraints:

$$f_{uct} = f_{uc,t-1} + s_{uct} + \sum_{v\in V^{in}(u)} \sum_{m} x_{vumct} - \sum_{v\in V^{out}(u)} \sum_{m} x_{uvmct} \text{ ,} \forall\, u, c, t \quad (1)$$

$$f_{uc0} = F_{uc}^0 \text{ ,} \forall\, u, c \quad (2)$$

$$\sum_{c} e_c * x_{uvmct} \leq K_{uvm} * y_{uvmt} \text{ ,} \forall\, (u, v) \in E, m, t \quad (3)$$

$$a_u \leq \sum_{c} f_{uct} * e_c \leq b_u \text{ ,} \forall\, u, t \quad (4)$$

$$f_{uct} \geq 0, \forall u, c, t \quad (5)$$

$$\sum_{v \in V^{out}(u)} \sum_{m} \sum_{c} e_c * x_{uvmct} \leq p_u, \forall u, t \quad (6)$$

$$x_{uvmct} \geq 0, \forall (u, v) \in E, m, c, t \quad (7)$$

$$\sum_{(u,v) \in E} y_{uvmt} * ZB_{uvm} \leq ZK_{mt}, \forall m, t \quad (8)$$

$$y_{uvmt} \in N, \forall (u, v) \in E, m, t \quad (9)$$

Constraint (1) is the flow conservation constraint for each commodity ***c*** at each node ***u*** and for every period ***t***. Constraint (2) implies that the stock level of commodity ***c*** at node ***u*** at time period 0 has to be equal to the initial stock level for commodity ***c*** at node ***u*** at time period 0. Constraint (3) ensures that the total flow does not exceed the space capacity of one transportation facility (unit) of mode ***m*** of the arc ***(u, v)***. Constraint (4) guarantees that the minimum storage requirement and maximum storage capacity at node ***u*** are respected. In addition to this, Constraint (5) makes sure that the stock level of commodity ***c*** at node ***u*** in time period ***t*** is a continuous variable and does not assume negative values. Constraint (6) ensures that weight of flows does not exceed the processing capacity of node ***u***. Constraint (7) guarantees that the flow variable is a continuous non-negative variable. Constraint (8) implies that the maximum time-based capacity of mode ***m*** at time ***t*** should not be exceeded by the time required for one tour on arc ***(u, v)*** in mode ***m*** to take place by the required number of transport units. Finally, Constraint (9) ensures that design variables are of integer type.

The problem poses many modeling and algorithmic challenges. The first complexity is the fact that CPLEX does not deliver an exact solution in reasonable time, and even LP relaxation requires excessive computational effort. The second difficulty concerns the tackling of the real-world setting, which is characterized by a large number of commodities (e.g. between 500 and several thousand), multiperiod setting and loose coupling between supply and demand, where each demand node may be satisfied by multiple different supply nodes and each supply node may serve multiple different demand nodes.

4. Implemented pegging and aggregation procedures

4.1 Pegging procedure

As already mentioned, commodity aggregation is applied to reduce the model complexity and the amount of data. However, before aggregation can be started, the pegging procedure has to be performed. Pegging is matching of supply with demand over time. The idea is to fix the loose coupling between supply and demand to identify potentially common sources or destinations of commodity flows. In the next paragraph the implemented pegging procedure is going to be described in detail.

Consider a supply network consisting of four nodes *i, j, k* and *l*, as shown in Figure 9. Node *i* and Node *l* are supply nodes, whereas Node *j* and Node *k* are demand nodes. As it can be seen in Figure 10, the corresponding supply and demand nodes are arranged in a grid-like fashion, adding time as an additional dimension. Node *i* is supplied in Period 0, 3 and 5, while Node *l* has a receipt of goods in Period 1, 2 and 5. The supply nodes need to satisfy the demand of Node *j* in Period 3 and 5, and Node *k* in Period 2, 3 and 5.

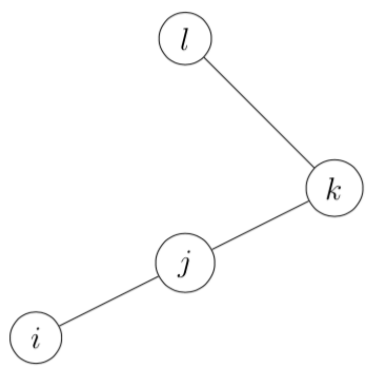

Figure 9: Supply/demand network with 4 nodes

Note that the geographical dimension is no more directly visible in that figure. In fact, in its current version, the pegging procedure does not take into account the distance between nodes. It is purely geared towards matching supply with demand over time, in a chronological fashion. In this example it is assumed that all supply and demand nodes correspond to the same commodity. The procedure starts by first sorting all demand nodes according to their time index, in non-decreasing order. A demand or supply node can be described by a pair *(x, t)* where *x* is the original node index and *t* is the time at which the demand or supply occurs. For the sake of simplicity, it is assumed that only unit quantities are associated with each demand and supply. Otherwise, each demand and supply node would have to be represented by a triple, including the quantity of the respective demand or supply. Let *S* and *D* denote the sorted list of supply and demand nodes, respectively. For the nodes in Figure 10, one obtains *D = ((k, 2), (j, 3), (k, 3), (k, 5), (j, 6)*) and *S = ((i, 0), (l, 1), (l, 2), (i, 3), (i, 5), (l, 5))*, assuming that ties are broken by a lexicographical comparison of the node IDs.

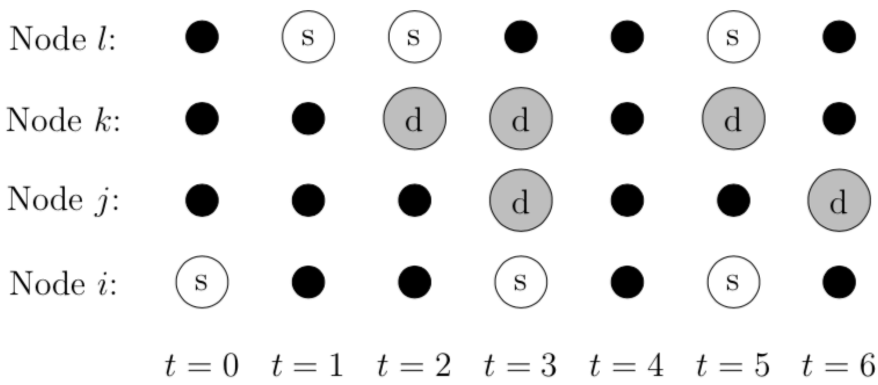

Figure 10: Supply and demand nodes over time: original situation

A strictly chronological pegging would lead to the following matched supply/demand pairs (see Figure 11):

- (i,0) → (k,2)
- (l,1) → (j,3)
- (l,2) → (k,3)
- (i,3) → (k,5)
- (i,5) → (j,6)

Note that geographical information is ignored during this step: Node *i* supplies Node *k*, although Node *l* would be closer and still feasible from a timing-oriented perspective. This in turn applies to the demand *(j,3)*, which is satisfied by *(l, 1)*, although Node *i* would be closer. At least from a distance-oriented point of view, the following partial matching would be better:

- *(i,0) → (j,3)*
- *(l,1) → (k,2)*

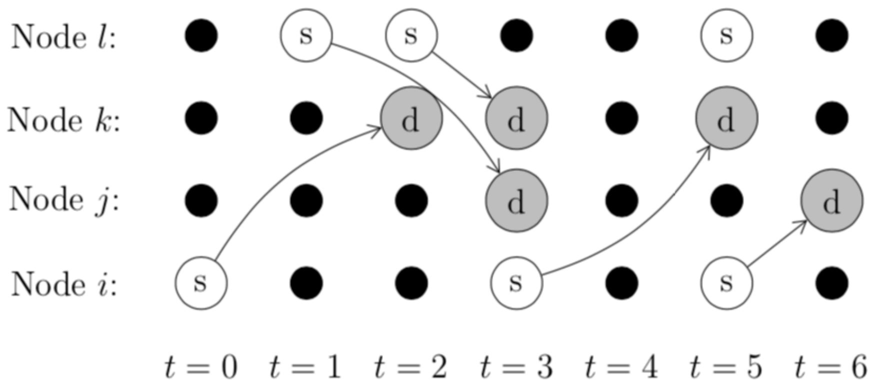

Figure 11: Supply and demand nodes over time: after chronological pegging

Although there are many different ways to connect supply with demand, two distinctive pegging procedures are introduced in this section. In this manner, we consider a more simplified version of a supply network, which consists of only three nodes *i, j* and *k*, where Node *i* is a supply node, and Node *j* and Node *k* are the demand nodes (see Figure 12).

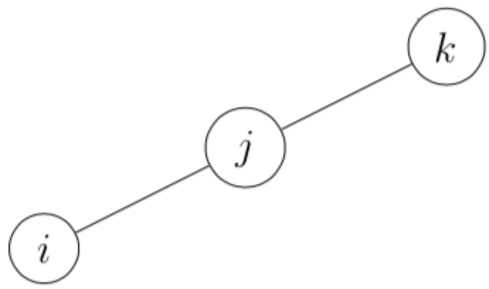

Figure 12: Supply/demand network with 3 nodes

As it can be seen in Figure 13, Node *i* has a receipt of goods in Period 0, 1, 3 and 5. It needs to satisfy the demand of Node *j* in Period 3 and 6, and Node *k* in Period 2, 3 and 6. It has to be also noted that there are more occasions when goods are demanded, than when they are produced. For the nodes in Figure 13, one obtains *D = ((k, 2), (j, 3), (k, 3), (j, 6), (k, 6))* and *S = ((i, 0), (i, 1), (i, 3), (i, 5))*, assuming that ties are broken by a lexicographical comparison of the node IDs.

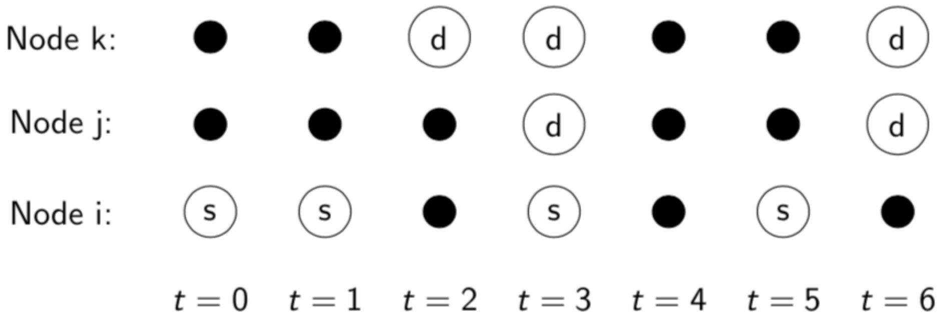

Figure 13: Coupling of supply and demand

In Figure 14 one can see different potential material flows. If we assume that Node *i* with its commodities from Period 0 wants to satisfy the demand of Node *k* in Period 2. The following material flow is possible: Node *i* can store commodities for one period and then in Period 1 it can deliver to Node *j*, which in turn stores commodities till Period 2 and then sends them in Period 2 to Node *k*.

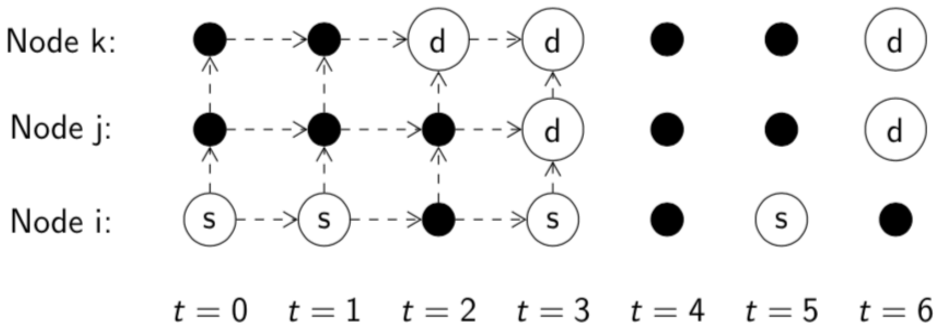

Figure 14: Potential material flow

In the first version, the chronological pegging leads to the following matched supply/demand pairs (see Figure 15):

- (i,0) → (k,2)
- (i,1) → (j,3)
- (i,1) → (k,3)
- (i,3) → (k,6)
- (i,5) → (j,6)

- (i,5) → (k,6)

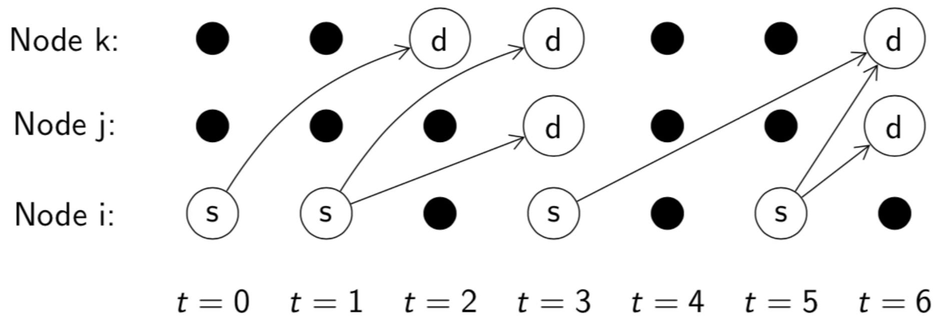

Figure 15: Pegging variant 1

On the other hand, in the non-chronological pegging procedure, one would get the following supply/demand matches (see Figure 16):

- (i,0) → (k,3)
- (i,1) → (k,2)
- (i,1) → (j,3)
- (i,3) → (j,6)
- (i,3) → (k,6)
- (i,5) → (j,6)

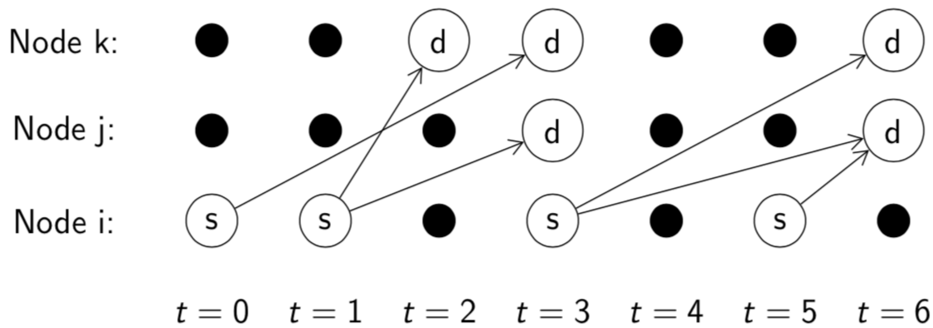

Figure 16: Pegging variant 2

The optimizer has full flexibility in choosing the most suitable final pegging procedure.

4.2 Commodity aggregation procedure

After the pegging has been established, the final step consists of aggregating supply/demand pairs of different commodities into a single one. The actual aggregation can be done in different ways. It might be determined by the geographical location of the supply and demand node, by their respective time indices, or by a combination of these two. The main idea is that one should aggregate commodities that have the same supply or demand node, are produced or consumed in the same time period and have the same weight or space requirement. When all three conditions are met, products can be coupled into a single new commodity. The problem with the aforementioned method is that each supplier can serve multiple demands and each demand can be satisfied by multiple suppliers. Therefore, the optimizer should determine the final pegging first. In the following paragraph, the most important aggregation principles are described based on examples. Besides the respective formalized aggregation rule, supply/demand pairs of three different commodities are given that can in fact be combined into a new, aggregated commodity.

The first type of aggregation is based on supply/demand pairs with the same source node. Commodities also have to be produced in the same time period. Two supply/demand pairs $((x, t_1), (y, t_2))$ and $((u, t_3), (v, t_4))$, arising from different commodities, can be aggregated if

$$x = u \wedge t_1 = t_3$$

Commodities 1, 2 and 3 are all produced in Node *i* and in the time period 1. Hence, in the following example, commodities 1, 2 and 3 can be grouped into a new aggregated commodity:

Commodity 1: $(i, 1) \rightarrow (k, 3)$

Commodity 2: $(i, 1) \rightarrow (l, 1)$

Commodity 3: $(i, 1) \rightarrow (i, 2)$

~~Commodity 4: $(i, 2) \rightarrow (j, 4)$~~

~~Commodity 5: $(k, 1) \rightarrow (j, 4)$~~

The second type of aggregation is based on supply/demand pairs with the same source base node. It is a relaxed version of the previous aggregation rule in the sense that only the geographical location of the source node has to be the same for two supply/demand pairs $((x, t_1), (y, t_2))$ and $((u, t_3), (v, t_4))$, hence, the following condition should be satisfied:

$$x = u$$

In this aggregation method, time indices do not play any role. It can also be seen in the example below:

Commodity 1: *(i, 1) → (k, 3)*
Commodity 2: *(i, 2) → (l, 4)*
~~Commodity 3: *(j, 1) → (k, 3)*~~
~~Commodity 4: *(l, 2) → (i, 2)*~~
Commodity 5: *(i, 4) → (j, 5)*

Here, commodities 1, 2 and 5 are grouped together based on their common source node, forming a new aggregated commodity.

Moreover, one can further the previous aggregation method, by adding the requirement that both supply/demand pairs are routed to the same destination node. In this manner, two pairs $((x, t_1), (y, t_2))$ and $((u, t_3), (v, t_4)$ are allowed to be aggregated if

$$x = u \land y = v$$

Hence, two commodities are grouped together when their origin and destination locations match. Although this type of aggregation delivers a potentially higher accuracy with regard to costs and a higher geographical consistency of the resulting aggregated commodities, this method does not resolve the geographical inaccuracies caused by the pegging procedure. It should be noted that all aggregation schemes are independent of the associated supply/demand quantities. However, in a real-world scenario, the size, volume or weight of the commodities may be an issue and impose restrictions on the aggregation process.

Aforementioned, two different aggregation procedures are implemented for the capacitated network loading problem, namely the single-node and unrestricted commodity aggregation. In the single-node aggregation variant, products are aggregated when they:

- share the same supply or demand node;

- are produced or consumed in the same time period;
- have the same weight or space requirement.

On the other hand, the second type of aggregation – the unrestricted commodity aggregation is concerned with grouping goods together when they:

- are produced or consumed in the same time period;
- have the same weight or space requirement.

The figures below depict, in a step-by-step manner, how the single commodity aggregation is performed. In Figure 17 there are two different commodities: commodity *c* and commodity *b*. It is assumed that both commodities have the same weight and space requirement. The pegging procedure has been already performed in a chronological fashion, and supply has been matched with demand. If we go period by period, we can see that both commodities neither have common supply nor demand node in Period 0. Furthermore, they are neither produced nor consumed in the same time period.

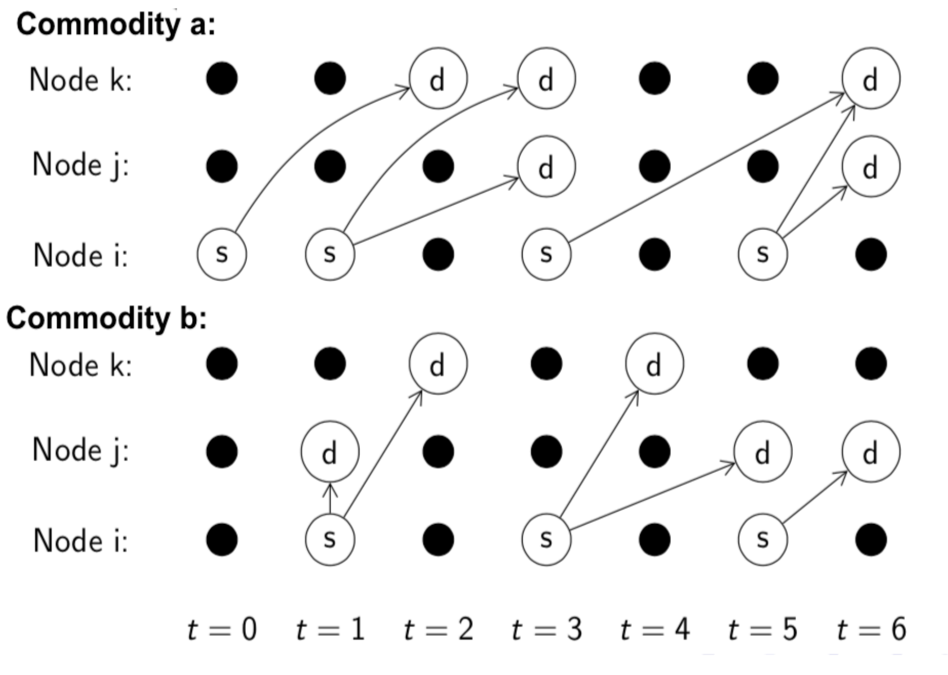

Figure 17: Single-node commodity aggregation in Period 0

However, in Period 1 commodities *a* and *b* share the same supply node and both are produced in the same time period (see Figure 18). Therefore, they can be aggregated into one new commodity. In this manner, the commodities are also grouped together in Period 3 and Period 5 (see Figure 19 and Figure 20). In contrast, in Period 2 and 4 there are no common source nodes and the conditions imposed by the single-node aggregation procedure are not met.

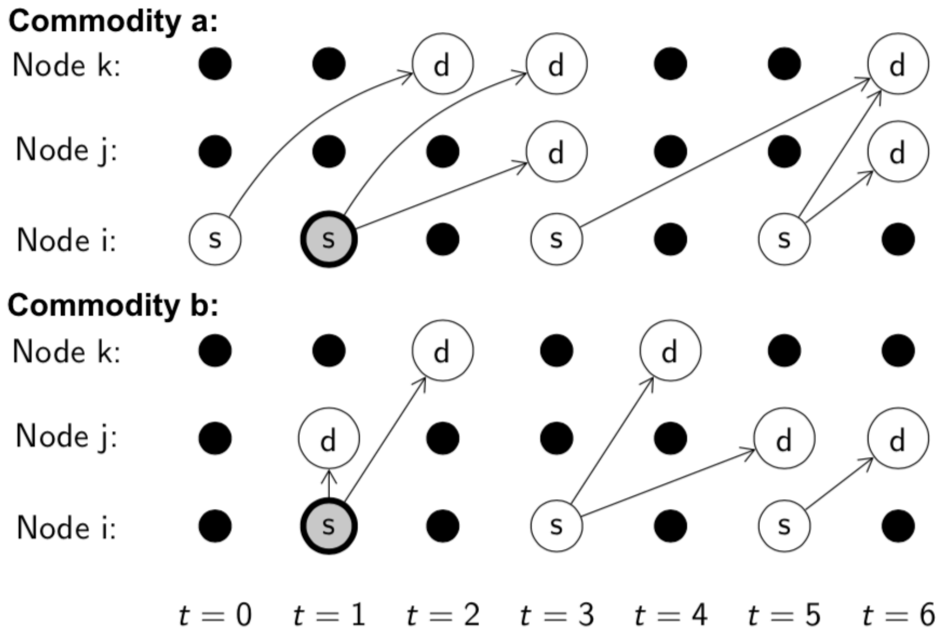

Figure 18: Single-node commodity aggregation in Period 1

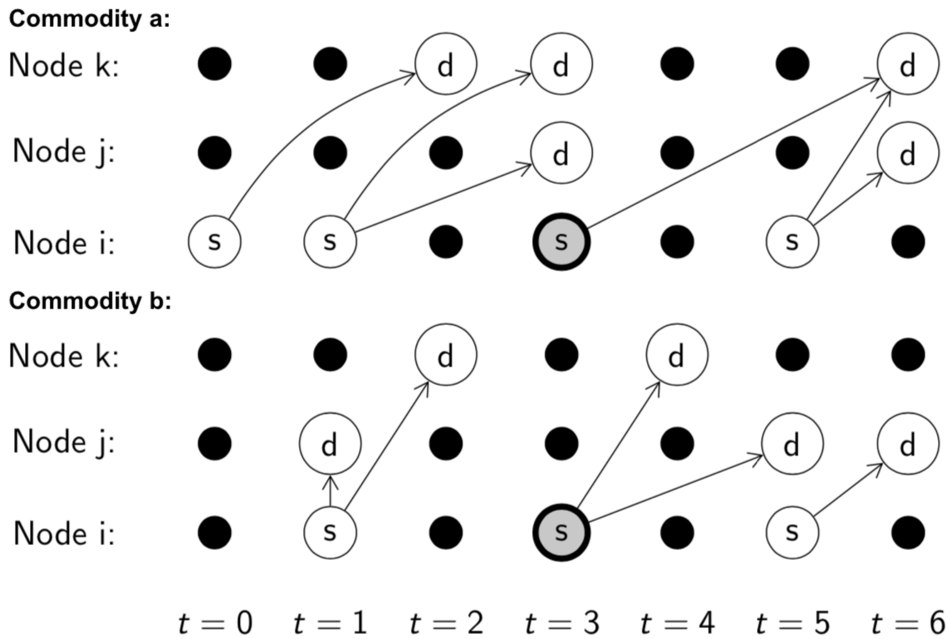

Figure 19: Single-node commodity aggregation in Period 3

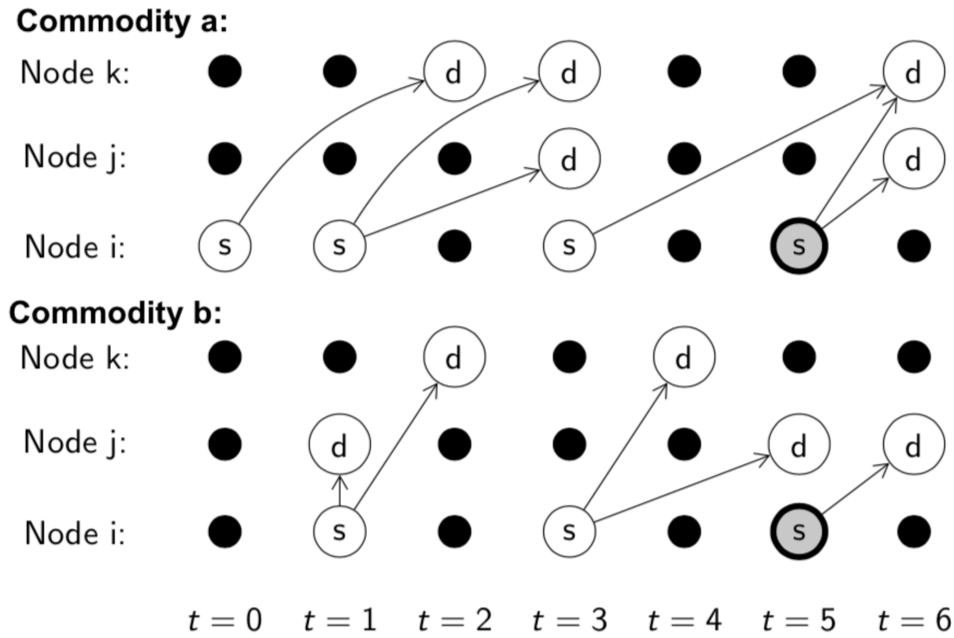

Figure 20: Single-node commodity aggregation in Period 5

5. Data analysis

Current chapter introduces the analysis of the data generated by Roland Braune's solution of the capacitated network loading problem. In the first data file set, single-node commodity aggregation with 100 commodities and 18 time periods is implemented. The second data file set deals with unrestricted commodity aggregation with 50 commodities and 10 time periods. The capacitated network loading problem considers the directed graph, which consists of four destination nodes (i.e. nodes 3, 4, 5, 9), two supply nodes (i.e. nodes 6, 7) and four transition nodes (i.e. nodes 0, 1, 2, 8). In total there are 10 nodes and 26 edges (see Figure 21).

It has to be noted that only one transportation mode is considered. Commodities are transported exclusively by trucks. Delivery vans and cars are not included here. Further, the actual objective function is reduced to fixed cost minimization. It should also be noted that both data sets have files that describe the mapping between supply/demand pairs and new commodities. The mapping provides information about commodities, source nodes, source time periods, destination nodes and destination time periods. Hence, the aggregation is in fact not applied to overall commodities, but to concrete material flows. Furthermore, both data sets include information about flows, inventory levels, tour counts, vehicle counts, edge time demands, initial inventories, supply and demand.

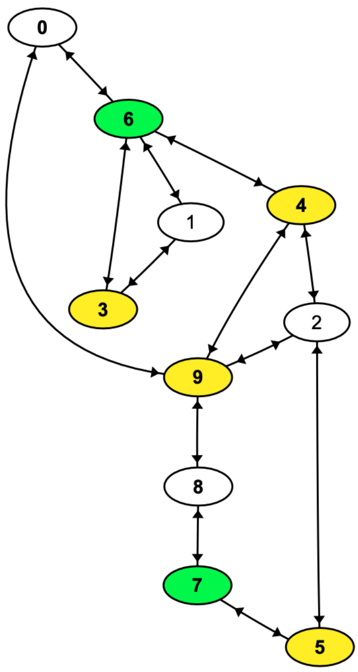

Figure 21: Internal supply network

5.1 Analysis of the single-node commodity aggregation

In the case of the single-node commodity aggregation, 100 commodities are aggregated into 44 commodities (see Figure 22). Although aggregateion results in problem size reduction, the objective function, the tour counts, the transportation time and the flows increase. At this point it can be assumed that the difference in the flows and other parameters results from the pegging and commodity aggregation procedures. Therefore, it is of interest to conduct an analysis to examine the effect of the single-node commodity aggregation method on the capacitated network loading problem.

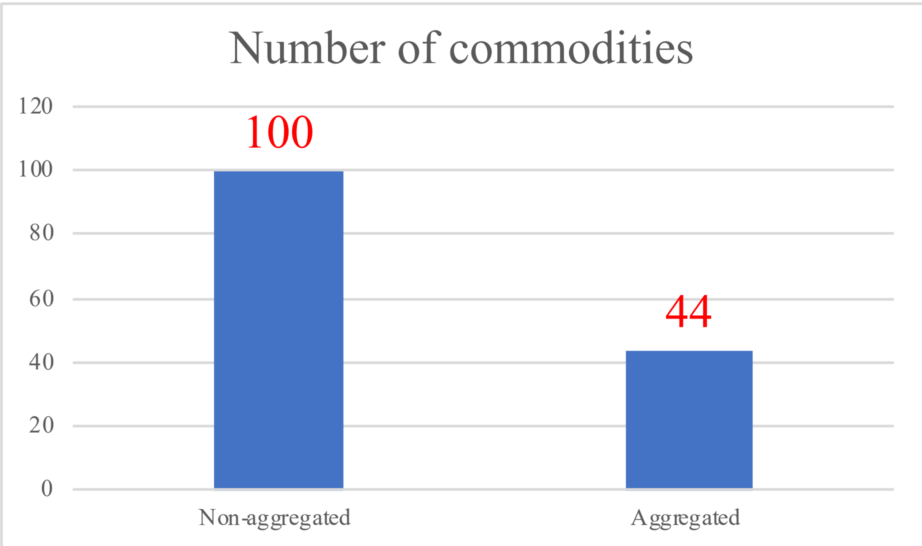

Figure 22: Number of commodities

After aggregation, objective function rose from 36131,33 EUR to 36713,16 EUR, which corresponds to an increase of 581,83 EUR. Since the objective function is a multiplication of transportation fixed cost with number of tours made and the time required to complete these tours, it is of interest to examine how the number of tours was affected by the single-node aggregation.

Time required to complete a tour is not influenced by aggregation, because the same time is needed to pass an arc, no matter how much the product range changes. However, what does change is the number of tours taken to transport all demanded commodities. The more commodities are needed to be moved from one location to another, the more tours are required. In the original version the total number of tours amounts to 126,33. In contrast, in the single-node aggregation, the number of tours increases to 130,19. When we multiple the required number of tours with the time necessary to complete these tours, we obtain the transportation time for all periods and all arcs of 582,76 hours (before aggregation). On the other hand, when commodities are aggregated, transportation time increases by 9,38 hours – from 582,76 to 592,15 hours. Consequently, one can calculate the objective function by multiplying 582,76 hours by 62 EUR (i.e. transportation fixed cost per hour) for the non-aggregated version and, similarly, multiplying 592,15 by 62 EUR for the aggregated version. As a result, one obtains the objective function of 36131,33 EUR before aggregation and 36713,16 EUR after aggregation. We can see that the objective function is influenced by the number of tours, which in turn is influenced by the commodity flows. In this manner, let's examine the flows in the internal supply network.

The total flow generated by the solution of the original problem amounts to 1263,33, whereas in the aggregated problem it amounts to 1301,88 (see Figure 23). It should also be noted that the total demand and the total supply are equal, and each of them amounts to 736 units.

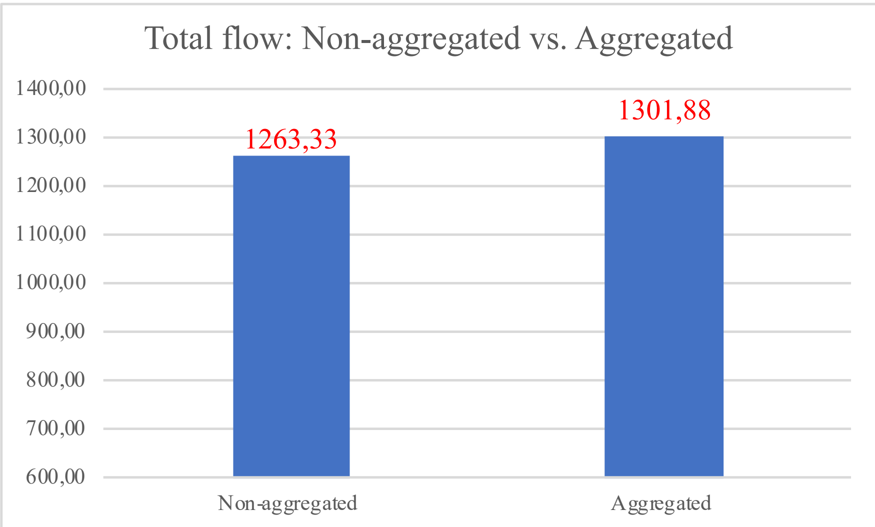

Figure 23: Total flow: Non-aggregated vs. Aggregated

As it can be seen in Figure 23, after aggregation the total flow increases by 38,54 units, and it is of interest to inspect the arcs that contribute to this difference. If we look at the flows of different arcs, we will be able to see that the flows in the aggregated version are not predominantly larger (see Figure 24). For example, in the non-aggregated version of the problem, each of the arcs (0, 9), (8, 9), (7, 8) and (9, 0) lets through larger flows compared to the aggregated model. On the contrary, when commodities are aggregated, the arcs (0, 6), (4, 9), (6, 4) and (6, 0) pass larger flows compared to the original problem. In addition to this, the edges (6, 3), (2, 5) and (4, 2) channel the same flow in both versions (see Table 9 and Table 10).

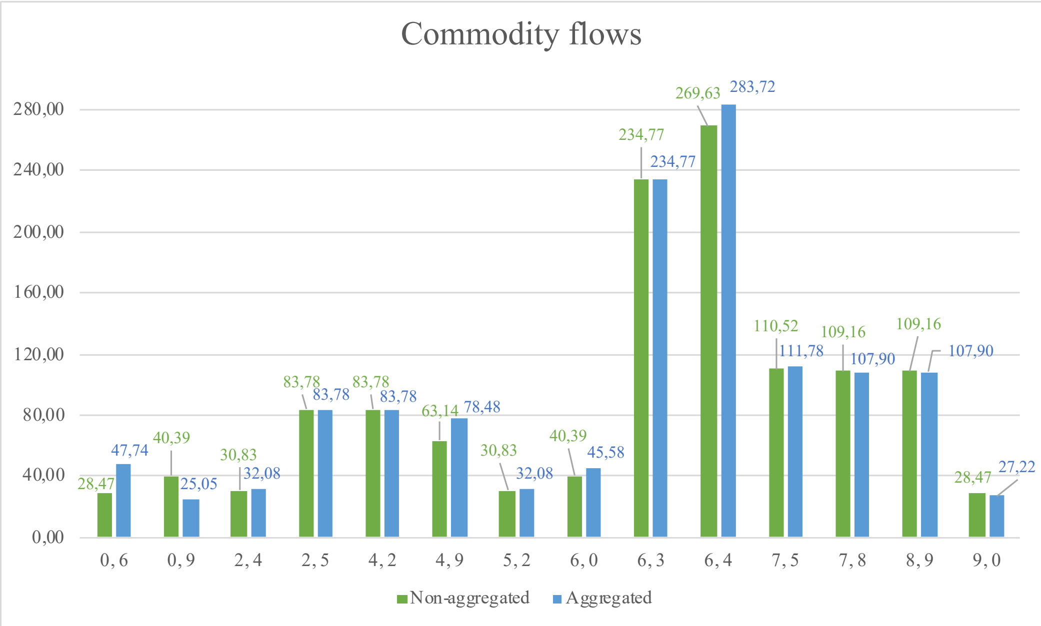

Figure 24: Commodity flows

In Figure 25, it can be seen that the edge (0, 6) is responsible for the biggest difference in flows, adding extra 19,27 units to the total flow of the aggregated problem. It is followed by the arc (4, 9) with a flow of additional 15,34 units. Nevertheless, the latter flow is compensated by the arc (0, 9), letting through a flow of 15,34 units fewer after aggregation. The same balancing characteristic can be observed for the arcs (7, 5), (2, 4) and (5, 2) – each of them increasing the flow by 1,25 units – and the arcs (8, 9), (7, 8), (9, 0) – each of them decreasing the flow by 1,25 units. In this manner, we can cross out these two groups of arcs. Thus, we are left with the arcs (6, 4) and (6, 0), which increase the overall flow difference by 14,08 and 5,19 units respectively. To sum it up, after all simplifications are performed, only three arcs – (0, 6), (6, 4) and (6, 0) – make up the total flow difference (see Figure 38).

In Figure 26, we can observe the percentage difference of each edge's aggregated flow in relation to the original problem. It can be seen that for the edge (0, 6) the flow increases by more than 50%. It is a very unusual difference, because in comparison to other arcs, the next biggest percentage flow difference is only 37,91%, contributed by the edge (0, 9), which in turn is followed by the arc (4, 9) with 24,29% difference. Therefore, later in this section, the observed deviation will be examined in more detail. Concerning the edges that comprise the total flow difference, in the aggregated version the flow on the edge (6, 4) increases by 5,22%, whereas on the edge (6, 0), it increases by 5,19%.

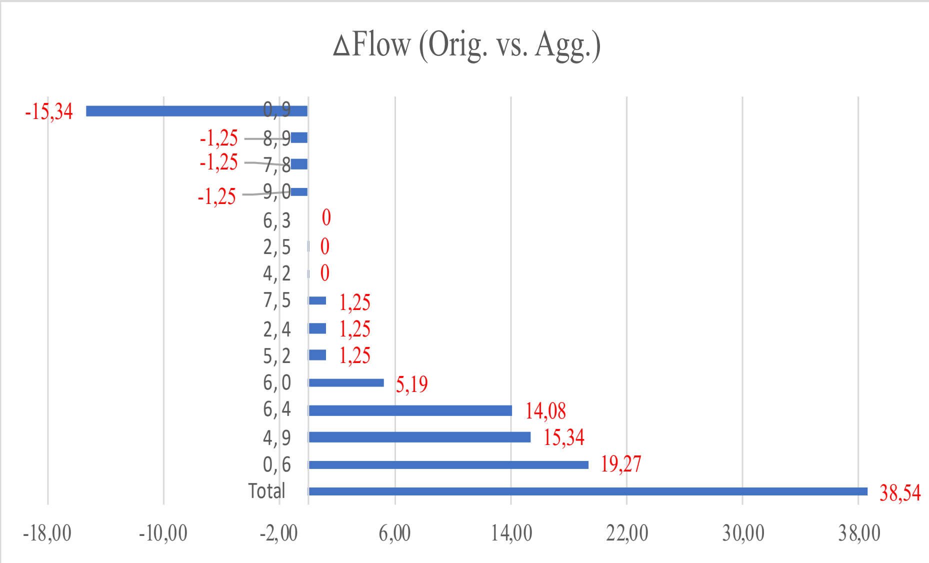

Figure 25: Flow difference (Non-aggregated vs. Aggregated)

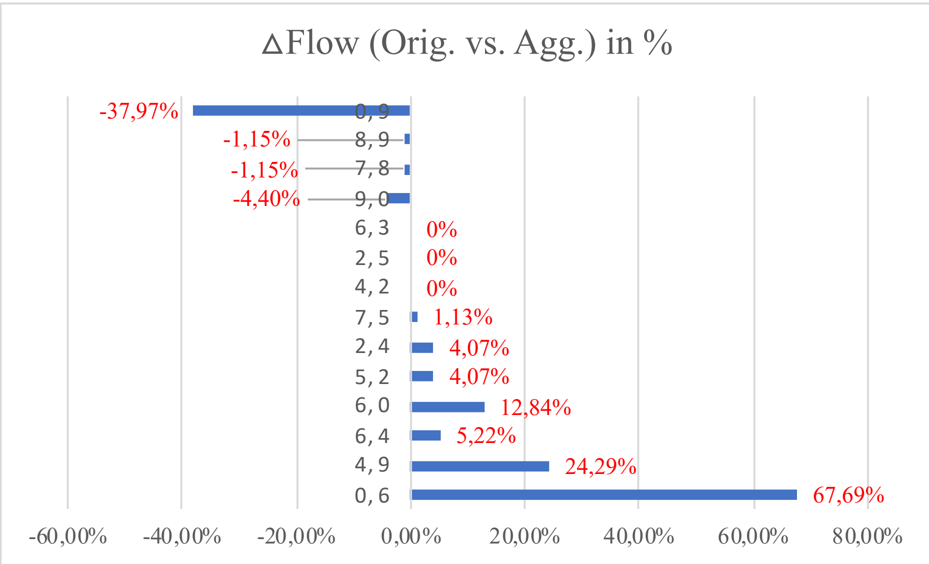

Figure 26: Flow difference in % (Non-aggregated vs. Aggregated)

Since, as previously mentioned, only three edges – (0, 6), (6, 4) and (6, 0) – play a role in increasing the total flow difference between the non-aggregated and aggregated problem, let's examine them in more detail. Namely, two paths are responsible for the resulted flow difference – (6-9) and (7-3). When we first take a look at the supply-demand pair 6-9, we can observe that it takes two different routes to satisfy the demand of Node 9 (see Figure 27). The first route is composed of nodes 6-0-9, and the second route consists of nodes 6-4-9. In the original problem, the former route channels 40,39 commodities and the latter route channels 63,14 of those. In the aggregated version of the problem, the path 6-0-9 has a detour: Node 6 sends 45,58 commodities to Node 0, only 25,05 of which are transported further to Node 9, the rest of 20,53 commodities will be sent back to the source node. As a result, due to aggregation, the flow between Node 6 and Node 0 increases by 5,19. Furthermore, the route 6-4-9 channels 78,48 commodities to the destination node, which is 15,34 more compared to the original problem. This increase in flow was caused by the fact that 15,34 commodities less were passed in the aggregated version in the first route between Node 0 and Node 9. In this manner, the route 6-4-9 equalizes the lacking flow. However, it should be noted that the flow of 15,34 passes two edges on route 6-4-9, namely, the edges (6, 4) and (4, 9). Therefore, by looking at the composition of the flow of commodities on edge (6, 4), it can be seen that in the aggregated version 15,34 commodities more are sent to Node 9, the same flow of 83,78 was sent to Node 5, and 1,25 commodities less are sent to Node 4. Thereby, if we sum the flows up, we obtain the flow difference of 14,08. Thus, the total flow difference increases by 5,19 (i.e. flow of 40,39 (non-aggregated) vs. 45,58 (aggregated)) because of the detour, and by 14,08 pieces (i.e. 269,63 vs. 283,72) because of the flow equalization. Since there are more commodity flows to channel in the network, the total transport time for the supply-demand pair (6, 9) increases by 8,35 hours, from 149,73 hours in the original version to 158,08 hours in the aggregated version (see Figure 39).

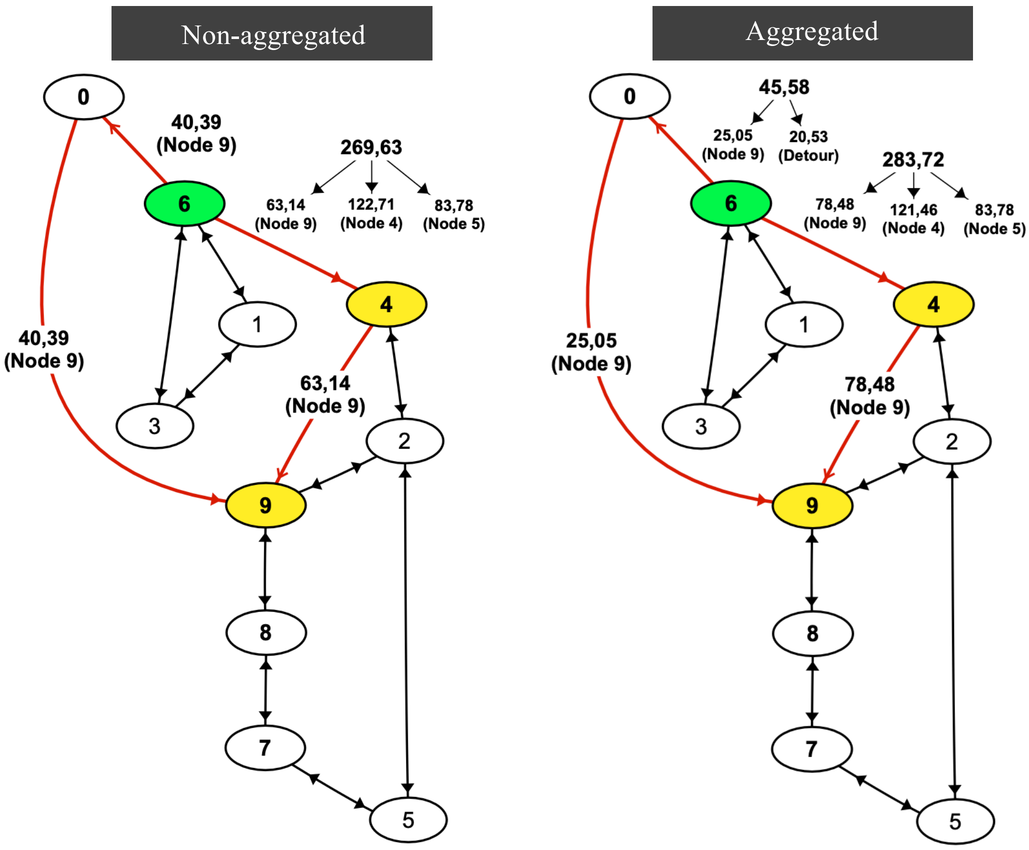

Figure 27: Flows [Path: 6-9] (Non-aggregated vs. Aggregated)

The second supply-demand pair that is responsible for the flow difference is 7-3 (see Figure 28). In order to satisfy the demand of Node 3, commodities take the following path 7-8-9-0-6-3. In the non-aggregated and aggregated versions of the problem, the flows are unremarkable up until Node 0. If we look at the flow between Node 0 and Node 6 for the non-aggregated problem and compare it with the aggregated problem, we will see that the flow difference increases substantially – from 1,25 to 19,27. As mentioned above, the reason for this unexpected jump is the detour that commodities take in supply-demand pair (6, 9). Specifically, we can see that Node 6 sends 20,53 units to Node 0 and then later it receives back from Node 0 the same number of commodities, as a result the flow of 20,53 happens twice, and there is a detour in the model. The commodities that are part of the detour, are then transported either to Node 3, where they are consumed, or Node 4, where they are consumed or forwarded further to other demand or transition nodes. The detour is caused by capacity constraints at all nodes, but especially at Node 6, where it operates at maximum capacity (i.e. 101,25) for five consecutive periods, between Period 1 and Period 5 (see Table 11). Thereby, 20,53 commodities are sent to Node 0, so that the supply Node 6 is able to produce and store the demanded commodities.

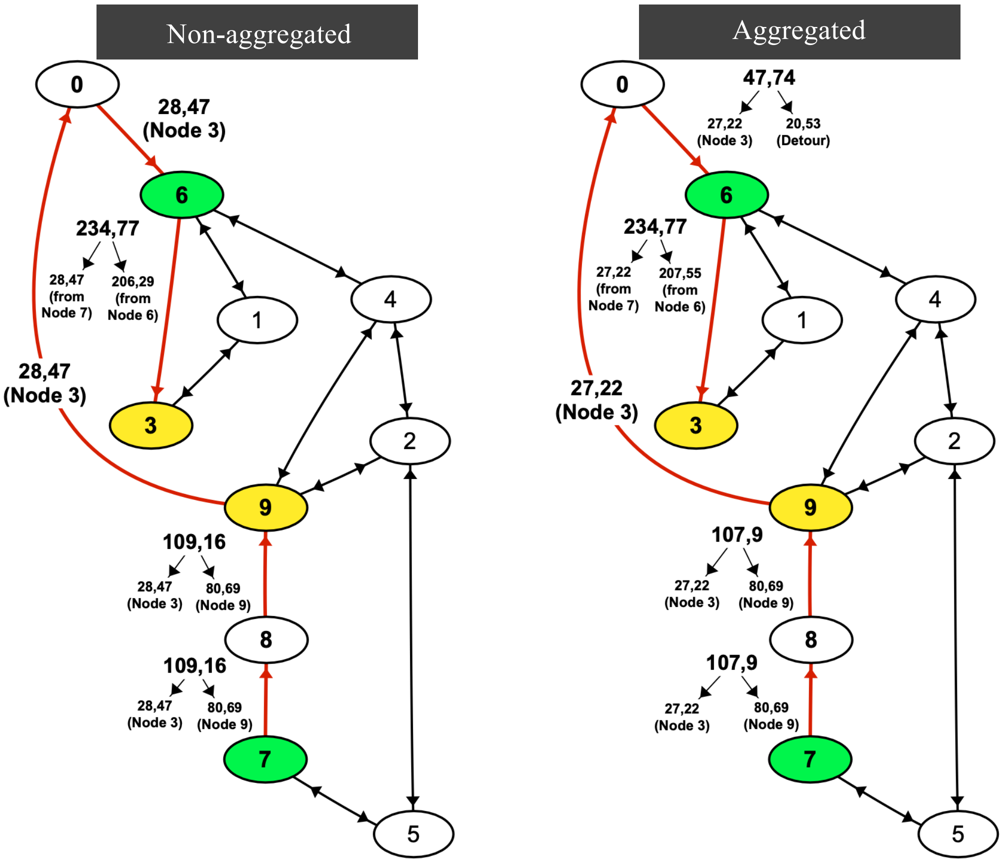

Figure 28: Flows [Path: 7-3] (Non-aggregated vs. Aggregated)

Finally, if we take a look at how transportation time changes after the aggregation, we can see that increased flows require more tours and more tours mean that more time will be spent transporting commodities and, hence, total costs will increase. For example, for the first supply-demand pair (6, 9) transportation time increases from 149,73 to 158,09 hours, while its total flow increases from 413,56 to 432,83. On the other hand, for the second supply-demand pair (7, 3) transportation time decreases from 244,3 to 243,54 hours, although its total flow increases from 510,03 to 525,53 (see Figure 40). In this case, the statement – the more flow the larger transportation time – does not apply, since the largest increase in flow happens on the edge (0, 6), which has the shortest distance among all edges in the internal supply network (see Table 1).

Table 1: Path's total flow and transportation time (Single-node aggregation)

	Non-aggregated		**Aggregated**	
Path	**Flow**	**Transport time**	**Flow**	**Transport time**
6-9	413,56	149,73	432,83	158,09
7-3	510,03	244,30	525,53	243,54

5.2 Analysis of the unrestricted commodity aggregation

In the case of the unrestricted commodity aggregation, 50 commodities are aggregated into 14 commodities. As with the single-node aggregated version, the unrestricted commodity aggregation reduces the problem complexity and the amount of data, but the flows, the tour counts, the transportation time and the objective function increase. Thereby, like for the single-node aggregation version, the flow analysis is conducted to find the causes for the above-mentioned effects.

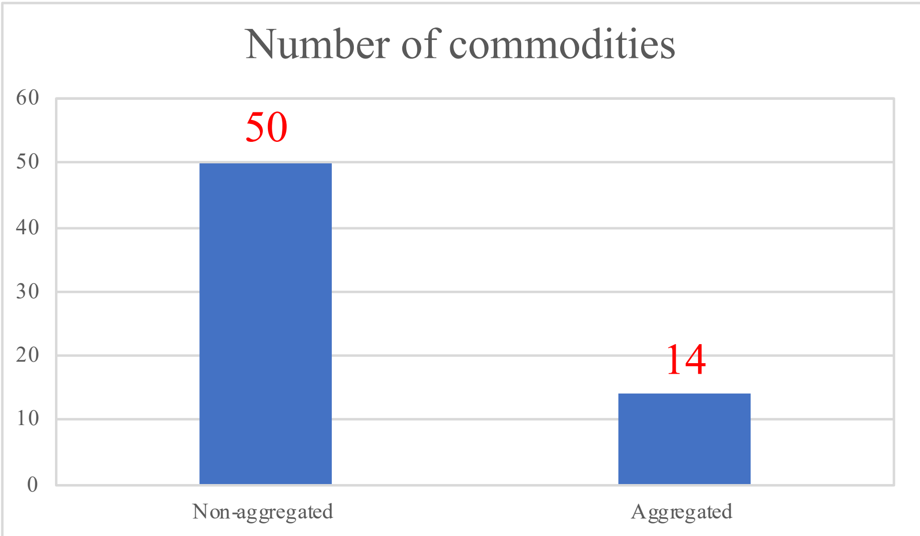

Figure 29: Number of commodities

If we look at the total flows before and after aggregation, we can see that the unrestricted commodity aggregation causes an increase of 265,94 units of flow, from 1106,23 to 1372,17 (see Table 12 and Table 13). Consequently, since there are more commodities to transport, more vehicles are required and the total costs for renting the vehicles increases by 2650,10 EUR. Figure 30 shows that after aggregation flows increase on all edges except the transport connections (6, 3) and (7, 5). The arc (6, 3) lets through the same amount of flow (i.e. 144,41) in the original and in the aggregated problem, and the arc (7, 5) has a slightly lower flow (i.e. $\Delta = 6{,}75$) in the aggregated version. The largest difference in flows is contributed by the edges (0, 6), (9, 0) and (6, 4). When commodities are aggregated, each of these edges increases the flow by 39,16 units. They are followed by the transportation connection (4, 9), which generates an additional flow of 32,71 units. The next largest deviation in flow is caused by the edges (4, 2) and (2, 5). Each of the arcs creates an extra flow of 30, 71 (see Figure 31).

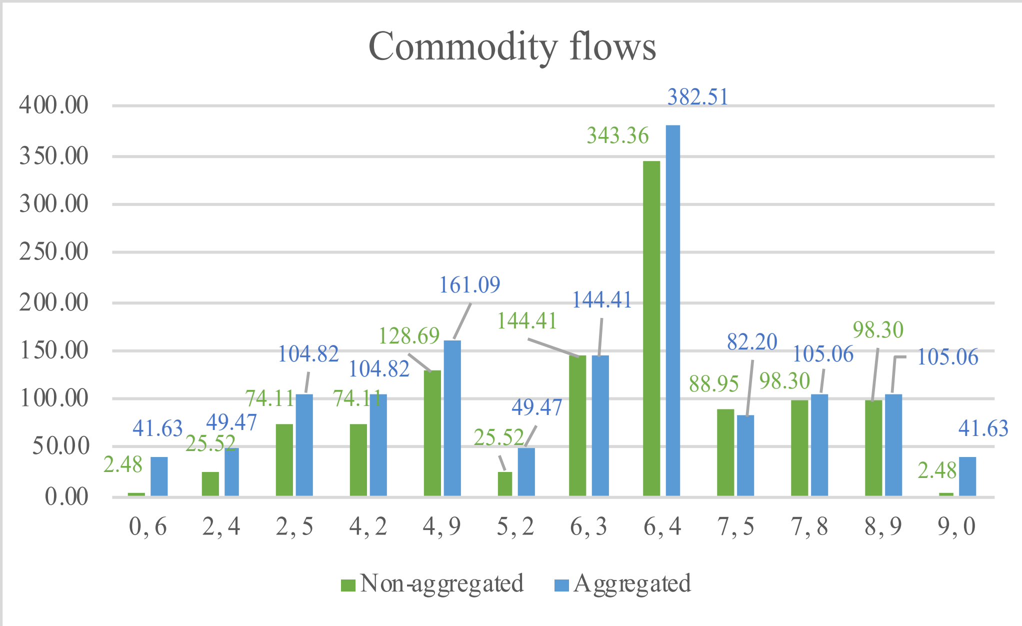

Figure 30: Commodity flows

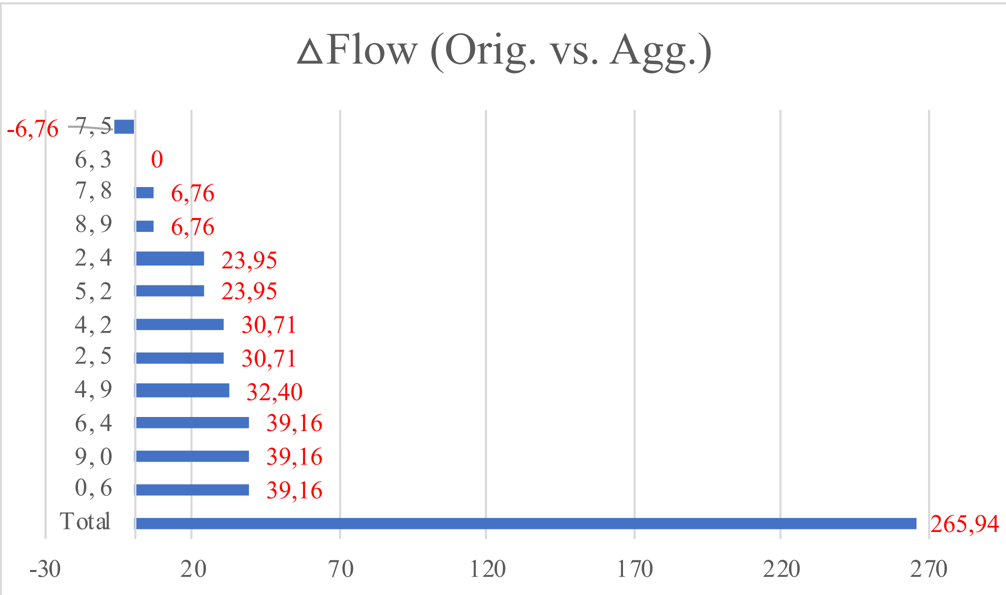

Figure 31: Flow difference (Non-aggregated vs. Aggregated)

Identical pattern can be observed for the tour counts, where the number of tours increases by 26,6 units after aggregation. In order to be able to see the implications of the unrestricted commodity aggregation more clearly, it is advised to take a look at Figure 32. The same can also be said about the arcs, which contribute the most to the tour count difference. As mentioned above these arcs are (0, 6), (9, 0) and (6, 4). Each of these edges increase the tour count value by 3,92. Similarly, the only negative difference can be seen on the arc (7, 5) and the same amount of tours in the original and aggregated version is made on the arc (6, 3). It is shown that the path 9-0-6-4 accounts for 44,23% of the tour count difference. Furthermore, the path 4-2-5 is responsible for an increase of 6,14 tours, which accounts for 23,09%. Moreover, the path 5-2-4 increases the tour count by 4, 78 or 17,98%. Finally, the path 7-8-9 is responsible for 1,36 tours or 5,11%.

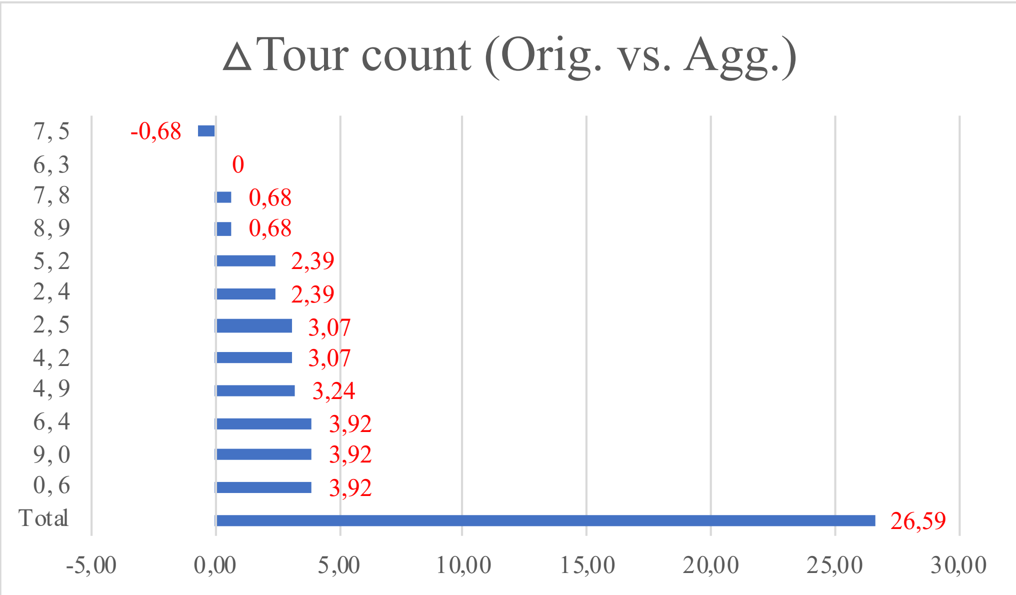

Figure 32: Tour count difference (Non-aggregated vs. Aggregated)

When we look at the difference in transportation time between the original and the aggregated versions, we can see that the largest transportation time occurs on the edge (9, 0), amounting to19,94 hours. The second largest difference is observed on the edge (4, 2), adding additional 19,68 hours. It is followed by the arcs (2, 4) and (6, 4) causing extra 13,89 and 13,42 hours respectively. In addition to this, when commodities are aggregated, on the edges (2, 5) and (4, 9) transportation time increases by 12,79 and 12,29 hours respectively. Finally, the arcs (8, 9), (5, 2), (0, 6) and (7, 8) are responsible only for 10,9 hours in total, which corresponds to 11,11% of the total transportation time difference. We can observe that even though the arc (0, 6) in the aggregated version is responsible for the largest difference in flows and the number of tours, it contributes only 1,94% to the total transportation time difference. The reason for this, is that it is the shortest transportation connection in the internal supply network.

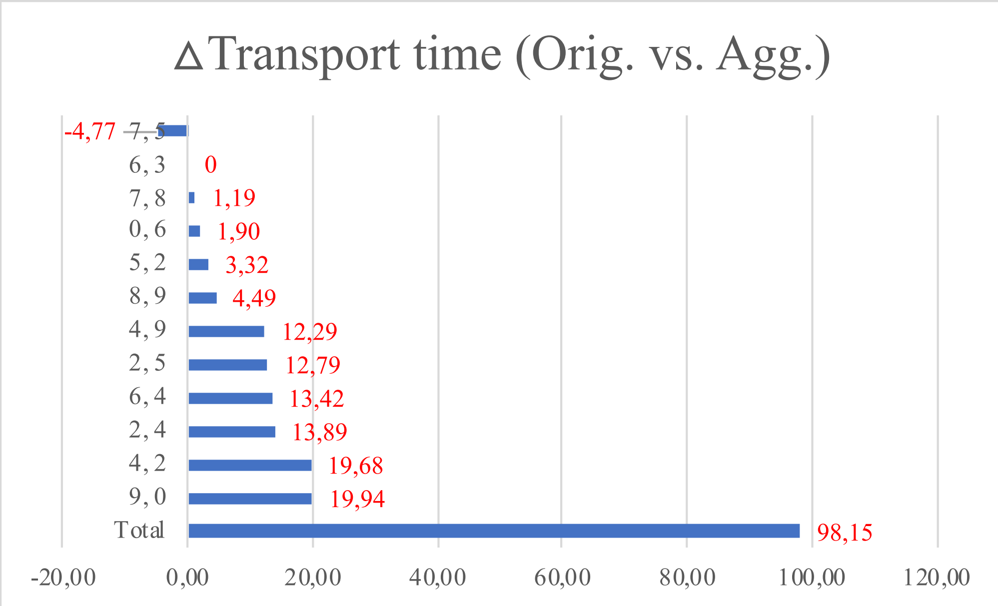

Figure 33: Transport time difference (Non-aggregated vs. Aggregated)

If we look at the product portfolio of Node 6 and Node 7 in the non-aggregated version, which consists in total of 50 different products, one can observe that Node 6 produces all products that Node 7 produces except commodity 16 and commodity 37 (see Table 2). In contrast, if we take a look at the product portfolio in the aggregated version, which consists of 14 different types of products, it can be noted that there are no common commodities produced in both plants (see Table 3). Therefore, when Node 5 requests a large quantity of commodity that is only supplied by the Node 6, the transportation time and, hence, the rental costs increase significantly, due to the fact that at least three edges have to be passed through, in order to get to Node 5 from Node 6. On the other hand, if the supply Node 7 had that commodity in its production portfolio, it would substantially decrease the costs, since Node 7 is situated right next to Node 5. In this manner, it limits greatly the ability of a supply chain manager to respond efficiently to demand requests in the internal supply network.

Hence, it can be concluded that the difference in flows comes from the pegging and aggregation procedures, which do not account for geographical aspects when matching supply with demand. In the non-aggregated version, supply nodes have a wider range of manufactured products, compared to the aggregated problem. In addition to this, in the non-aggregated model the same 20 types of commodities are produced in both sites making it much cheaper to satisfy demands from various locations. In contrast to that, the aggregated version of the problem does not feature any common manufactured products, which results in larger flows.

Table 2: Product portfolio of Node 6 and Node 7 for the non-aggregated commodities

Node 6		Node 7
0	25	1
1	26	2
2	27	6
3	28	7
4	29	8
5	30	9
6	31	14
7	32	15
8	33	16
9	34	17
10	35	22
11	36	23
12	38	24
13	39	25
14	40	26
15	41	27
17	42	28
18	43	31
19	44	35
20	45	36
21	46	37
22	47	40
23	48	46
24	49	

Table 3: Product portfolio of Node 6 and Node 7 for the aggregated commodities

Node 6	Node 7	
1	0	8
2	3	9
4	5	10
11	6	13
12	7	

Now, that the differences between the aggregated and non-aggregated models in terms of flows, number of tours, transportation time and product portfolios have been established, it is of interest to analyze four specific paths that cause the formation of these dissimilarities and that are responsible for the total flow difference. These supply-demand pairs are (7, 3), (6, 5), (7, 4) and (6, 9). The path analysis is focused only on flows, commodities and transportation time. Specifically, the amount of flow that occurs between the nodes, what commodities are shipped and how much time does it take in total to ship goods from the origin to destination. It should be noted, that supply equals demand, and they amount to 672,54.

First, let's analyze the supply-demand pair (7, 3), since its edges are responsible for the biggest flow difference. The path includes the following nodes: 7-8-9-0-6-3. We should compare the non-aggregated and aggregated graphs of the internal supply network to observe how the aggregation of commodities affects the objective function. If we look at Figure 34, we can see that the flows in the non-aggregated version are smaller compared to the aggregated version. When commodities are not aggregated, the demand of Node 3 of 144,41 units is satisfied by Node 7 with a flow of 2,48 and Node 6 with a flow of 141,93. It is important to underline that 98% of demand of Node 3 is satisfied by its nearest supplier – Node 6. In contrast, in the aggregated version, Node 6 meets only 71% of demand of Node 3. It is satisfied by Node 7 with a flow of 41,63 units, which is comprised of commodities 1, 2, 4, 11, 12, and Node 6 with a flow of 102,77 units, which is comprised of commodities 0, 3, 5, 6, 7, 8, 10. A more detailed information about the kinds of commodities that are transferred to their destination node can be seen in Figure 41. It is evident that aggregation makes the internal supply network less flexible, because there are more different types of commodities flowing from Node 7 to Node 6 in the aggregated version than in the non-aggregated version. When commodities are not aggregated, the edges (7, 8) and (8, 9) each have a flow of 98,3, where 95,82 is demanded by Node 9 and 2,48 is demanded by Node 3. The latter demand can be seen on the edge (9, 0), where the initial flow from Node 7 has decreased from 98, 3 to 2,48, after demand of Node 9 was satisfied. Furthermore, it is known that the edge (9, 4) is not used in the model at all. Therefore, Node 3 is the only plausible destination node. On the other hand, when commodities are aggregated, a flow of 105,06 passes through edges (7, 8) and (8, 9) until 63,43 units are consumed by Node 9. As a result, a large flow of 41,63 is transported through the edges (9, 0), (0, 6) and finally arrives at Node 3 through arc (6, 3). To sum it up, the supply from Node 7 to Node 3 is larger by 39,15 units compared to the same supply in the non-aggregated version, and it needs to pass as many as five edges to reach the final node. Hence, due to commodity aggregation, the total flow of the path (7, 3) is larger by 91,82 units (i.e. 345,97 vs. 437,79), which is already responsible for 35% of the total flow difference of the model. In addition to this, since there are more commodities to move from one location to another, the total transportation time for the supply-demand pair (7, 3) increases by 27,52 hours – from 168,08 to 195,60 hours, corresponding to an increase of 16,38% (see Table 4 and Figure 42).

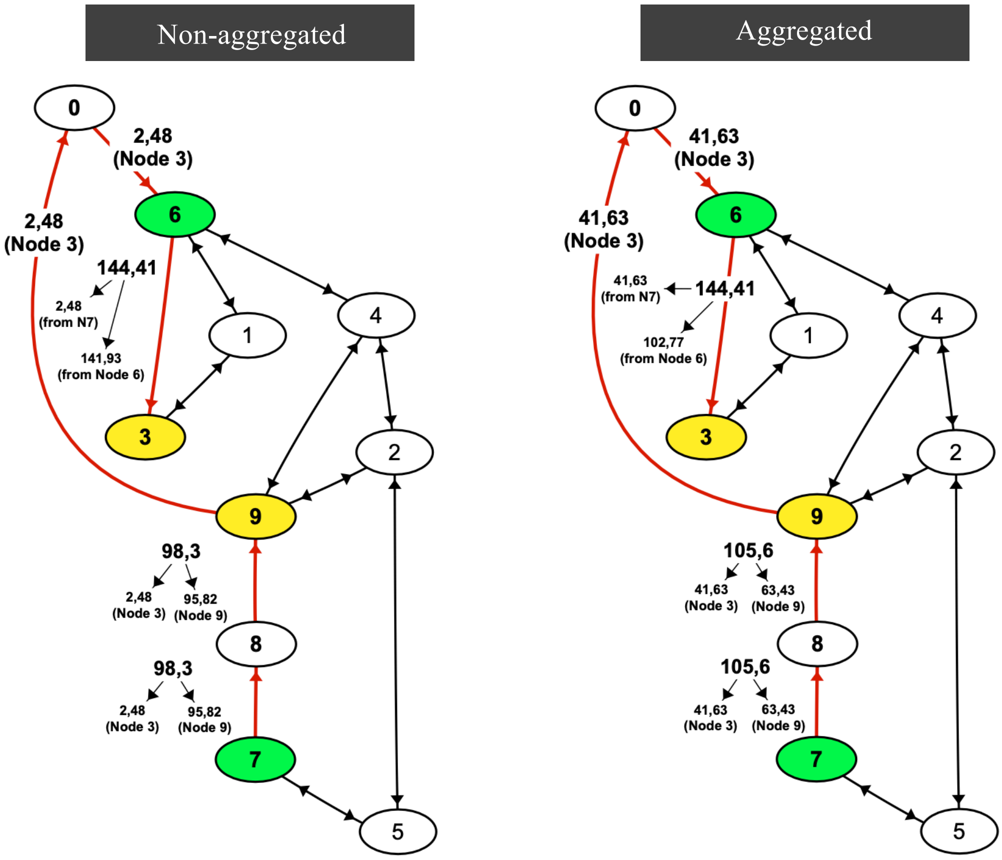

Figure 34: Flows [Path: 7-3] (Non-aggregated vs. Aggregated)

The second path that is going to be analyzed is the supply-demand pair (6, 5). The tour includes the following nodes: 6-4-2-5. When we look at the flow of aggregated commodities and the products that form this flow, we can see that destination Node 5 requires every product type that Node 6 produces. The path's total flow increases by 100,57 units. It is an increase of 20,46% compared to the non-aggregated version, and it is responsible for 37,82% of the total flow difference of the model. Furthermore, due to commodity aggregation, the path's transport time increases by 45,89 hours (see Figure 43).

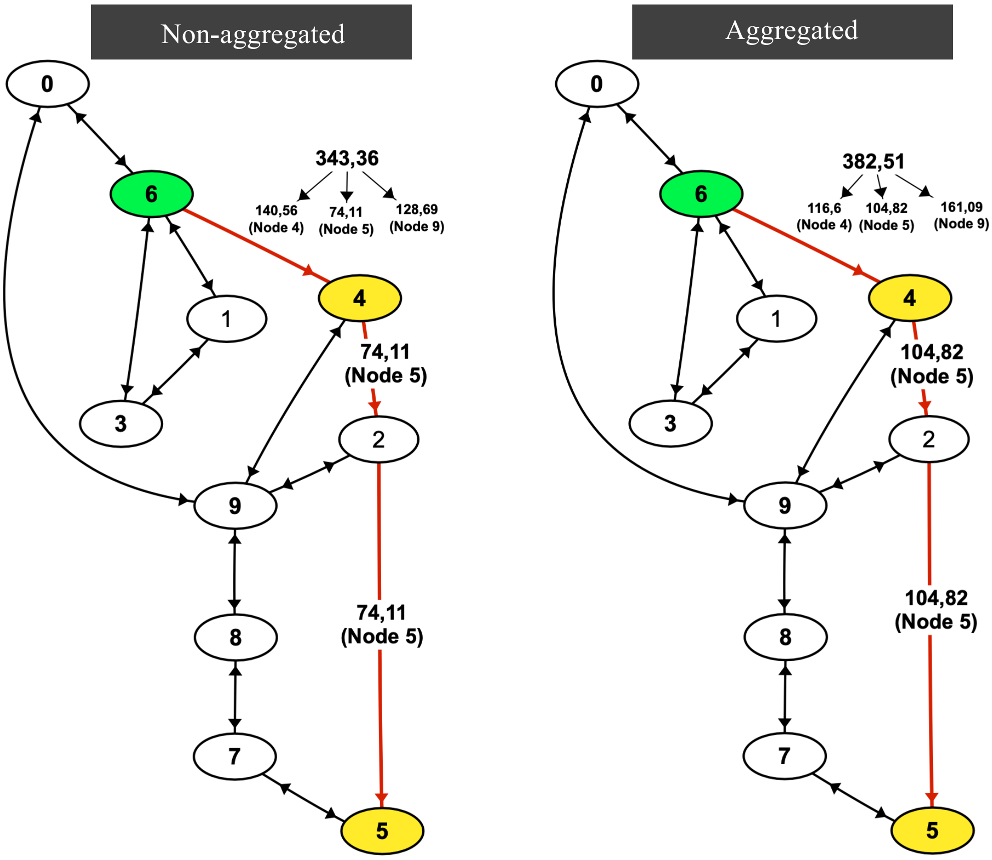

Figure 35: Flows [Path: 6-5] (Non-aggregated vs. Aggregated)

The next supply-demand pair that is going to be analyzed in this section is (7, 4). For commodities to arrive at Node 4 from Node 7, the following path is driven: 7-5-2-4. It has to be mentioned that Node 6 is the nearest supplier and Node 7 is the furthest supplier. Node 4 has a total demand of 166,07, which is satisfied differently depending on whether commodities are aggregated or not. In the non-aggregated version 25,52 commodities are transported from Node 7 to Node 4, whereas in the aggregated version 49,47 commodities are shipped form the furthest supplier. As it can be seen, the same pattern that was observed in the aforementioned supply-demand pairs is happening here, where an increased number of products is transported from a remote supplier (i.e. Node 7). Node 4 is highly dependent on Node 7, since there are no common commodities produced in both plants and it requires all types of products manufactured in Node 7. Consequently, the total transport time for the supply-demand pair (7, 4) increases by 12,44 hours (see Figure 44).

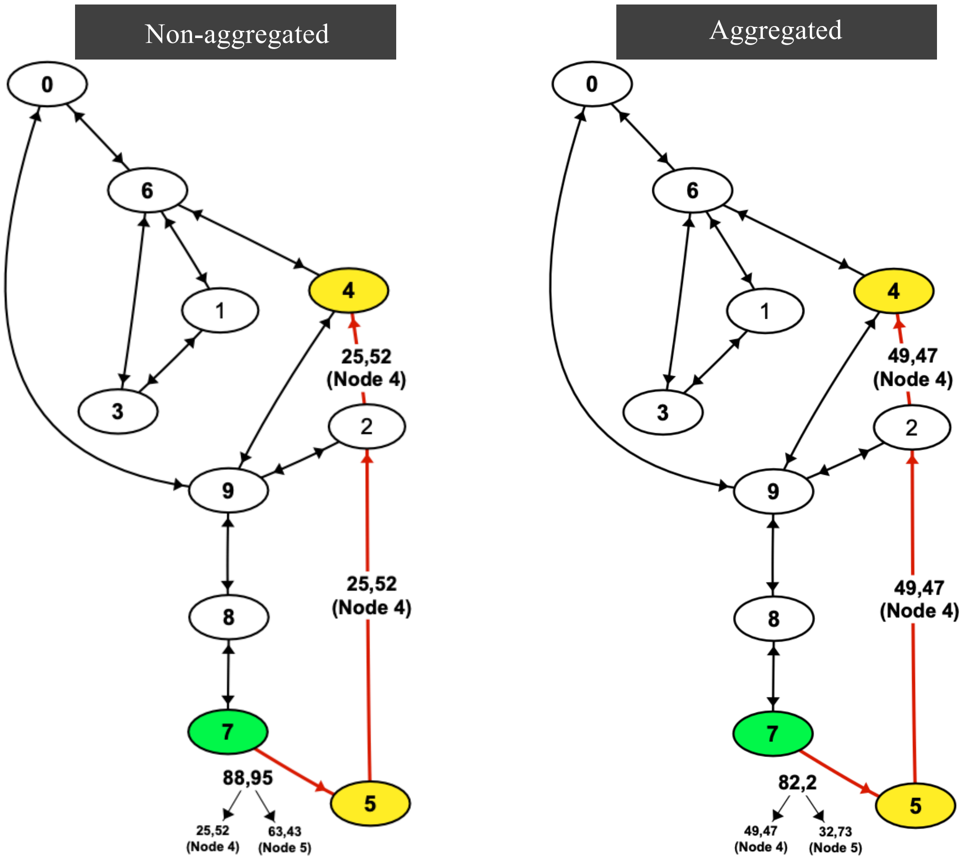

Figure 36: Flows [Path: 7-4] (Non-aggregated vs. Aggregated)

Finally, the last path that is going to be analyzed, is (6, 9). It starts at Node 6, passes Node 4 and arrives at Node 9, meeting 57% of its demand in the non-aggregated version and almost 72% in the aggregated version. Hence, commodities are channeled through the following path: 6-4-9. It has to be noted that although it may seem that the distance from Node 6 to Node 9 is similar to the distance from Node 7 to Node 9, it is not the case. It takes in total 83,87 hours more time to transport non-aggregated commodities from Node 6 to Node 9. The difference in transport time becomes even larger, when the aggregated version is considered. It increases from 83,87 to 103,91 hours. When we look at the flow analysis graphs in Figure 37, we can see that the total flow of aggregated commodities of the path 6-4-9 rises by 71,56 units, or 15,16%. It repeats the same pattern of all the previous paths, where the furthest supplier (i.e. in this case it is Node 6) increases its deliveries by a larger rate than the nearest supplier (i.e. Node 7), resulting in higher total costs. A closer look at composition of deliveries shows that in the aggregated version Node 9 demands all types of commodities that Node 6 produces, whereas in the non-aggregated version Node 9 is supplied by Node 6 with commodities that either are not produced in Node 7 or are manufactured in insufficient quantities. Lastly, as it can be seen in Figure 45, commodity aggregation causes an increase of 25,72 hours for the supply-demand pair (6, 9).

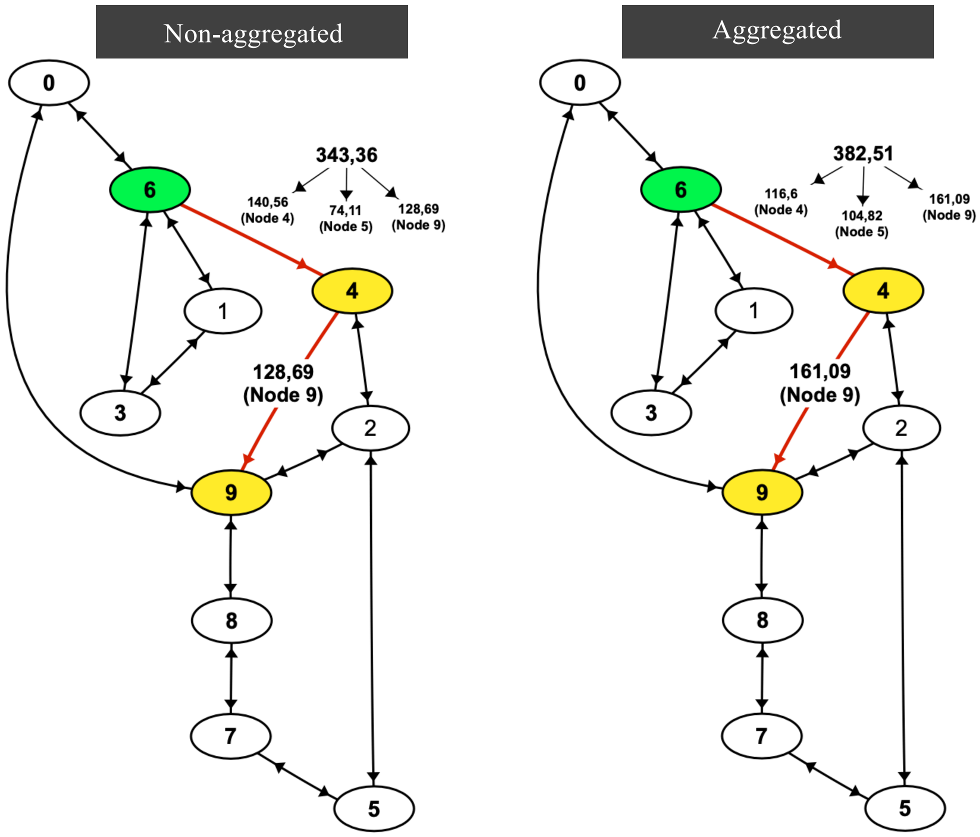

Figure 37: Flows [Path: 6-9] (Non-aggregated vs. Aggregated)

Table 4: Path's total flow and transportation time (Unrestricted aggregation)

	Non-aggregated		Aggregated	
Path	**Flow**	**Transport time**	**Flow**	**Transport time**
7-3	345,97	168,08	437,80	195,6
6-5	491,58	196,07	592,15	241,97
7-4	139,99	81,11	181,13	93,55
6-9	472,05	166,52	543,61	192,24

5.3 Analysis of the improved pegging procedure

The findings from Section 5.1 and 5.2 indicate where the inaccuracies come from. It is revealed that the straightforward pegging procedure causes unnecessary additional flows from the furthest supplier, because it does not account for the distances between the supply and demand vertices. Further, the analysis shows that the capacitated inventories with aggregated commodities result in the detour, which increases the flows.

In this manner, Roland Braune, improved the pegging procedure to resolve the geographical inconsistencies. The new implemented pegging procedure includes the computation of the shortest paths between all nodes using the Floyd-Warshall algorithm, where the edge time demands are regarded as edge weights. Consequently, each demand node must be served by its closest supply nodes according to the shortest path results. In order for this principle to work a simple network flow problem has to be solved. This linear programming model has to meet the following properties: *i)* flows can be computed separately for each commodity; *ii)* for each pair of supply and demand, an arc is created; *iii)* the weight (i.e. cost coefficient) of each arc corresponds to the shortest path between the origin node and the destination node; *iv)* the total inflow of a demand node has to be equal to the demand quantity; *v)* the total outflow of a supply node cannot exceed its supply quantity. The objective is to minimize the overall flow cost, obtained by multiplying the flow quantity on each arc with the arc weight (i.e. the shortest path value). Since there are no intermediate or transshipment nodes, the model corresponds to the classical transportation problem.

In addition to the improved pegging procedure, Roland Braune relaxed the problem by removing the inventory capacity constraints, to eliminate the detour that existed between Node 0 and Node 6, after the single-node aggregation had been implemented. Since the inventory capacity was fully loaded at Node 6 between Period 1 and Period 5, some of the commodities were transferred to Node 0, in order to free some space for the newly produced commodities.

The analysis of the new data file set reveals that the improved pegging procedure and the relaxation of the inventory capacity constraints, indeed, fix the issue with uneven objective function values, while holding the lower bound property. Since the flows become the same for the non-aggregated and aggregated versions, the number of tours, the transportation time and the transportation costs also became equal. Thereby, the implemented and improved pegging procedure does not possess the inaccuracies present in the previous version of the model. Commodity aggregation effectively reduces the amount of data and the problem complexity.

If we look at the figures below, we can see that the total flows of the original and the single-node aggregated problem are identical and each of them amounts to 1260,83. Consequently, their objective function values are equal to 35847,46 EUR. It has to be noted that for the instance with 100 commodities, the LP solution of the non-aggregated version is different now, due to the missing inventory capacity limitations. However, for the instance with 50 commodities, the LP solution of the non-aggregated problem stays the same. Regarding the model with unrestricted commodity aggregation, the LP solution of the aggregated problem yields exactly the same total flows and the objective function as for the full LP. Here, their total flows amount to 1106,23 with an equal objective function value of 13339,98 EUR.

Table 5: Commodity flows (Non-aggregated)

Edge \ Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Total
0, 6	4,01	13,61	6,22	1,47												0,02	1,89	**27,22**
2, 4	8,13	12,35	5,45	2,93	0,86	1,29			0,58	0,18	0,00						0,31	**32,08**
2, 5	3,22	9,32	29,14	10,15	14,82	5,99	4,94	2,53	0,32	0,22	0,33	1,27			0,02	0,23	1,28	**83,78**
4, 2	3,22	12,14	28,07	8,59	16,19	5,59	5,14	1,42	0,32	0,22	0,38	1,27		0,10	0,02	0,23	0,87	**83,78**
4, 9	7,05	26,28	35,79	11,46	11,97	4,61	1,78	2,54					0,30		0,05	0,39	1,31	**103,54**
5, 2	9,09	14,58	3,25	2,93	0,77	1,14											0,31	**32,08**
6, 3	24,07	55,77	90,36	25,66	13,63	3,82	7,26	5,71	1,15	1,38		1,43	0,97			0,02	3,53	**234,77**
6, 4	25,75	54,65	103,39	47,82	31,64	14,33	8,77	7,62	0,11	2,59		5,39	0,61			0,69	5,40	**308,77**
7, 5	15,36	37,88	39,83	9,88	4,42	3,39											1,02	**111,78**
7, 8	15,02	40,64	34,36	11,03	3,91	2,56				0,14			0,00				0,24	**107,90**
8, 9	15,02	36,49	36,83	11,03	3,91	2,56							0,72				1,34	**107,90**
9, 0	7,85	10,41	7,49	1,47														**27,22**
Total	**137,80**	**324,11**	**420,18**	**144,44**	**102,11**	**45,29**	**27,89**	**19,82**	**2,48**	**4,73**	**0,72**	**9,38**	**2,61**	**0,10**	**0,08**	**1,59**	**17,50**	**1260,83**

Table 6: Commodity flows (Single-node aggregation)

Edge \ Period	1	2	3	4	5	6	7	8	9	10	11	16	17	Total
0, 6	1,88	14,84	8,79	1,47									0,25	**27,22**
2, 4	3,02	17,49	6,63	2,65	0,77	0,90						0,04	0,59	**32,08**
2, 5			71,75	3,80	2,90	2,92	1,48	0,93			0,00			**83,78**
4, 2		40,33	31,42	3,80	2,90	2,92	1,48	0,93						**83,78**
4, 9		48,25	21,55	1,38	7,93	19,59		4,84						**103,54**
5, 2	3,92	22,35	2,70	0,82	0,77	0,90						0,04	0,59	**32,08**
6, 3	17,64	71,33	68,79	28,30	5,15	4,91	28,57	6,24		0,36	2,58		0,90	**234,77**
6, 4	0,95	158,53	70,94	23,95	14,71	3,51	27,79	6,38		0,63	0,99		0,39	**308,77**
7, 5	17,24	34,82	41,51	9,72	2,61	4,95							0,93	**111,78**
7, 8	23,82	39,71	30,51	6,87	1,92	4,55			0,34				0,18	**107,90**
8, 9	20,91	39,16	28,52	6,87	1,92	4,55					5,45		0,52	**107,90**
9, 0	2,38	21,48	1,89	1,47										**27,22**
Total	**91,78**	**508,27**	**385,00**	**91,09**	**41,58**	**49,70**	**59,32**	**19,32**	**0,34**	**0,99**	**9,01**	**0,08**	**4,35**	**1260,83**

Table 7: Commodity flows (Non-aggregated)

Edge \ Period	1	2	3	4	5	6	7	8	9	Total
0, 6		0,26	1,81			0,12	0,11	0,10	0,08	**2,48**
2, 4	1,25	1,35	19,37	2,32	0,96		0,26	0,02		**25,52**
2, 5	6,89	5,81	47,92	5,46	6,80		0,55		0,69	**74,11**
4, 2	7,04	5,81	47,94	5,46	6,80		0,55		0,53	**74,11**
4, 9	5,79	16,84	70,58	19,52	11,90		2,57		1,48	**128,69**
5, 2	1,25	8,99	13,37	1,63			0,26	0,02		**25,52**
6, 3	9,90	31,54	64,56	24,21	7,81	3,24	2,12	0,45	0,58	**144,41**
6, 4	18,89	48,52	180,84	45,39	38,03	3,88	5,40		2,40	**343,36**
7, 5	2,43	22,74	45,15	15,99	2,30				0,35	**88,95**
7, 8	1,33	25,76	55,33	14,44	0,70				0,73	**98,30**
8, 9	1,33	18,28	62,59	14,44	0,70				0,96	**98,30**
9, 0	0,41	0,26	1,81							**2,48**
Total	**56,51**	**186,15**	**611,26**	**148,87**	**76,01**	**7,24**	**11,81**	**0,59**	**7,80**	**1106,23**

Table 8: Commodity flows (Unrestricted commodity aggregation)

Edge \ Period	1	2	3	4	5	6	7	8	9	Total
0, 6			2,04					0,44		**2,48**
2, 4			23,72	0,83	0,96					**25,52**
2, 5	25,97		41,04	1,76	4,66	0,69				**74,11**
4, 2	25,97	23,93	17,10	1,76	4,66	0,69				**74,11**
4, 9	44,31		44,61	12,55	0,34	24,97		1,92		**128,69**
5, 2			23,72	1,79						**25,52**
6, 3			124,39	2,59	11,28	4,82	0,65	0,44	0,23	**144,41**
6, 4	70,28	23,93	191,03	31,70	19,44	2,06	4,31	0,61		**343,36**
7, 5	24,32	37,42	15,83	7,03	4,36					**88,95**
7, 8	28,68	43,57	21,52	4,54						**98,30**
8, 9		28,68	65,09	2,00	2,54					**98,30**
9, 0			2,48							**2,48**
Total	**219,53**	**157,53**	**572,57**	**66,54**	**48,23**	**33,24**	**4,96**	**3,41**	**0,23**	**1106,23**

6. Conclusion

To sum it up, this thesis explored the capacitated network loading problem with its modeling and algorithmic challenges. Commodity aggregation and pegging procedures were described in detail. The main focus of the thesis was on analyzing the data generated by Roland Braune's solution of the multiperiod multicommodity capacitated fixed charge network flow problem with integer design variables and intermediate storage, where single-node and unrestricted commodity aggregation procedures were applied.

The aggregation resulted in problem size reduction and improvement of LP calculation time, enabling the solution of large-scale problems with more than hundred commodities. However, in the aggregated variant, the objective function, the flows, the tour counts and the transportation times increased. Flow analysis was conducted to find out why this was the case and what factors caused the deviations. The analysis revealed that the inventory capacity limitations were responsible for the detours and the increased flows, while the difference between the non-aggregated and aggregated variants resulted from the implemented pegging and aggregation procedures. To be more specific, commodity aggregation altered the range of products produced by the manufacturers in a negative way. Some of the demand nodes could not be served by their closest supply nodes, since the nearest supply nodes did not have the demanded products in their product portfolio anymore. Furthermore, the inventory capacity constraints restricted the commodity flow in the network. They caused some unnecessary detours for a small number of flows that exceeded the storage constraints. It is safe to say that the analysis has helped to improve the aggregation and pegging procedure, and after the elimination of the inventory capacity constraints, the LP solution of the aggregated problem yielded exactly the same objective function value as for the original LP, and the lower bound property held. The improved pegging procedure embodied the Floyd-Warshall algorithm to compute the shortest paths between all nodes. In this manner, each demand node could be served by its closest supply nodes according to the shortest path algorithm. In addition to this, a LP model represented by a simple network flow model, which corresponds to the classical transportation problem, was solved to minimize the overall flow costs in the pegging procedure.

In conclusion, the thesis provides some interesting avenues for future research. Since pegging and aggregation methods have only been tested under a fixed cost scheme (the actual objective function was reduced to fixed cost minimization), it would be interesting to see the cost estimation when variable transportation costs, holding costs and handling costs are taken into consideration. Further, the pegging procedure could be adapted and improved to resolve the geographical inconsistencies. It would be also interesting to see other applications of the shortest path algorithm for the pegging procedure, in order to account for distances between the nodes. Finally, it might be beneficial to solve the problem with inventory capacities to obtain a LP solution that would be more applicable to the real-world internal supply chain.

References

Avella, P., Mattia, S., & Sassano, A. (2007). Metric inequalities and the Network Loading Problem. *Discrete Optimization, 4(1)*, 103-114.

Babonneau, F., & Vial, J.-P. (2010). A partitioning algorithm for the network loading problem. *European Journal of Operational Research, 204(1)*, 173-179.

Balakrishnan, A., Magnanti, T. L., & Wong, R. (1989). A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design. *Operations Research, 37(5)*, 716-740.

Balas, E., Ceria, S., & Cornuejols, G. (1993). A Lift-and-Project Cutting Plane Algorithm for Mixed 0-1 Programs. *Mathematical Programming, 58(1)*, 295-324.

Barahona, F. (1996). Network Design Using Cut Inequalities. *SIAM Journal of Optimization, 6(3)*, 823-837.

Bienstock, D., Chopra, S., Guenluek, O., & Tsai, C.-Y. (1998). Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming, 81*, 177-199.

Bitran, G. R., Haas, E. A., & Hax, A. C. (1982). Hierarchical Production Planning: A Two-Stage System. *Operations Research, 30(2)*, 232-251.

Boland, N., Ernst, A., Kalinowski, T., Rocha de Paula, M., Savelsbergh, M., & Singh, G. (2013). Time Aggregation for Network Design to Meet Time-Constrained Demand. *20th International Congress on Modelling and Simulation*, (pp. 3281-3287). Adelaide, Australia.

Burchett, D., & Richard, J.-P. (2015). Multi-commodity variable upper bound flow models. *Discrete Optimization, 17*, 89–122.

Chauhan, S. S., Frayret, J.-M., & LeBel, L. (2009). Multi-commodity supply network planning in the forest supply chain. *European Journal of Operational Research, 196*, 688-696.

Chauhan, S. S., Frayret, J.-M., & LeBel, L. (2011). Supply network planning in the forest supply chain with bucking decisions anticipation. *Ann Oper Res, 190*, 93-115.

Chouman, M., Crainic, T. G., & Gendron, B. (2017). Commodity Representations and Cut-Set-Based Inequalities for Multicommodity Capacitated Fixed-Charge Network Design. *Transportation Science, 51(2)*, 650-667.

Christopher, M. (2016). *Logistics and Supply Chain Management* . New York : Pearson .

Crainic, T. G., Frangioni, A., & Gendron, B. (2001). Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics, 112(1)*, 73–99.

Crainic, T. G., Gendreau, M., & Farvolden, J. (2000). A Simplex-Based Tabu Search Method for Capacitated Network Design. *INFORMS Journal on Computing, 12(3)*, 223-236.

Fleischmann, B., Meyr, H., & Wagner, M. (2008). Advanced Planning. In H. Stadtler, & C. Kilger, *Supply Chain Management and Advanced Planning* (pp. 81-106). Berlin: Springer-Verlag.

Fragkos, I., Cordeau, J.-F., & Jans, R. (2017). The Multi-Period-Commodity Network Design Problem. *CIRRELT* , 1-50.

Gavish, B. (1986). Augmented Lagrangean Based Algorithm for Centralized Network Design . *IEEE Transactions on Communications, 33(12)*, 1247-1257.

Gavish, B., & Neuman, I. (1989). A System for Routing and Capacity Assignment in Computer Communication Networks. *IEEE Transactions on Communications, 37(4)* , 360-366.

Gendron, B., Crainic, T. G., & Frangioni, A. (1999). Multicommodity Capacitated Network Design. In B. Sanso, & P. Soriano, *Telecommunications Network Planning* (pp. 1-27). Boston: Springer.

Goetschalcks, M., & Fleischmann, B. (2008). Strategic Network Design . In H. Stadtler, & C. Kilger, *Supply Chain Management and Advanced Planning* (pp. 117-132). Berlin: Springer.

Grötschel, M., Monma, C., & Stoer, M. (1995). Design of Survivable Networks. *Handbooks in Operations Research and Management Science, 7(1)*, 617-672.

Kilger, C., & Wagner, M. (2008). Demand Planning. In H. Stadtler, & C. Kilger, *Supply Chain Management and Advanced Planning* (pp. 133-159). Berlin : Springer-Verlag.

Lamar, B. W., Sheffi, Y., & Powell, W. B. (1990). A capacity improvement lower bound for fixed charge network design problems. *Operations Research, 38(4)*, 704-710.

Lee, D. N., Medhi, K. T., & Strand, J. L. (1989). Solving large telecommunications network loading problems. *AT&T Technical Journal, 68(3)*, 48-56.

Ljubic, I., Putz, P., & Gonzalez, J.-J. S. (2012). Exact Approaches to the Single Source Network Loading Problem. *Networks, 59(1)*, 89-106 .

Magnanti, T. L., & Wong, R. T. (1984). Network Design and Transportation Planning: Models and Algorithms. *Transportation Science, 18(1)*, 1-55.

Magnanti, T. L., Mirchandani, P., & Vachani, R. (1995). Modeling and Solving the Two-Facility Capacitated Network Loading Problem. *Operations Research, 43(1)*, 142-157.

Mangan, J., Lalwani, C., & Butcher, T. (2008). *Global Logistics and Supply Chain Management.* Chichester: John Wiley & Sons, Ltd.

Mejri, I., Layeb, S., & Zeghal, M. F. (2019). Enhanced Exact Approach for the Network Loading Problem. *6th International Conference on Control, Decision and Information Technologies*, (pp. 970-975). Paris.

Mirchandani, P. (1989). Polyhedral Structure of a Capacitated Network Design Problem with an Application to the Telecommunication Industry.

Newman, A. M., & Kuchta, M. (2007). Using aggregation to optimize long-term production planning at an underground mine. *European Journal of Operational Research, 176(2)*, 1205-1218.

Park, H., & Thurman, W. N. (2013). A Bayesian approach to aggregation in demand systems: smoothing with perfect substitution priors. *Applied Economics, 45(31)*, 4452-4462.

Rohde, J., & Wagner, M. (2008). Master Planning . In H. Stadtler, & C. Kilger, *Supply Chain Management and Advanced Planning* (pp. 161-179). Heidelberg: Springer-Verlag.

Shapiro, J. F. (1998). Bottom-up vs. Top-down Approaches To Supply Chain Management and Modeling. *Advances in Quantitative Modeling for Supply Chain Management*, 737-760.

Stadtler, H. (1988). *Hierarchische Produktionsplanung bei losweiser Fertigung.* Heidelberg: Physica.

Stadtler, H. (2008). Supply Chain Management – An Overview . In H. Stadtler, & C. Kilger, *Supply Chain Management and Advanced Planning* (pp. 9-36). Berlin : Springer-Verlag.

Wang, L.-C., Cheng, C.-Y., & Wang, W.-K. (2016). Flexible supply network planning for hybrid shipment: a case study of memory module industry. *International Journal of Production Research, 54(2)*, 444-458.

Appendix

A. Tables

Table 9: Commodity flows (Non-aggregated)

Edge \ Period	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Total
0, 6	0,03		0,28	1,87	0,53	7,43	1,21	1,68	4,35	1,98	1,21	1,17	0,79	1,79	0,75	1,06	1,11	1,25	28,47
0, 9						2,51	6,21	2,93	2,60	5,87	5,86	2,11	3,05	2,17	0,78	1,66	3,36	1,29	40,39
2, 4				2,24	0,37	3,71	9,24	1,08	3,60	0,56	0,48	1,48	1,94	1,71	0,51	1,74	1,35	0,83	30,83
2, 5		13,14	19,04	11,39	6,93	8,32	2,79	7,68	4,63	2,65	2,22		1,23	0,90	0,53	1,25	0,92	0,18	83,78
4, 2	22,88	25,45	22,49	7,51	5,46														83,78
4, 9	16,33	16,37	19,58	3,95	6,79												0,13		63,14
5, 2	0,10	2,56	8,28	2,28	4,59	1,82	3,98		0,08	1,49	0,44	0,62	0,77	0,87	0,91	0,77	0,43	0,83	30,83
6, 0	2,42	19,77	14,99	0,07	3,14														40,39
6, 3	18,75	48,84	12,71	18,78	18,32	13,20	11,78	23,12	20,40	10,04	5,39	7,26	3,72	3,18	1,49	6,73	7,33	3,73	234,77
6, 4	69,28	78,65	42,06	23,48	19,92	4,35	5,85	2,14	5,23	1,75	2,38		4,30	1,11	0,18	4,69	2,57	1,69	269,63
7, 5	0,64	10,71	12,76	12,75	7,71	4,16	10,47	8,16	6,60	11,21	4,74	0,59	6,86	2,43	3,54	3,56	2,77	0,87	110,52
7, 8	4,53	15,22	22,06	43,28	3,03	2,82	0,30		5,11	1,34	1,35	1,16	1,02	0,39	3,09	1,90	2,04	0,53	109,16
8, 9	0,03	1,53	5,09	13,36	8,12	11,36	10,14	7,24	12,19	12,27	3,46	5,54	4,40	4,61	2,97	1,01	4,23	1,61	109,16
9, 0	0,03	1,39	4,79	7,41	3,30		1,71	1,10	0,22	1,68	0,50	1,94	0,02	0,57	0,94	0,58	1,28	1,01	28,47
Total	**135,02**	**233,63**	**184,13**	**148,37**	**88,20**	**59,67**	**63,68**	**55,12**	**65,01**	**50,83**	**28,01**	**21,87**	**28,11**	**19,73**	**15,67**	**24,95**	**27,52**	**13,82**	**1263,33**

Table 10: Commodity flows (Single-node aggregation)

Edge \ Period	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Total
0, 6				1,75	0,96	6,57	14,21	0,61	8,12	1,31	2,01	0,96	3,38	1,17	0,37	4,92	1,39		**47,74**
0, 9				0,37		1,24	1,17	0,11	1,82	1,15	8,57	3,18	6,30	0,22	0,77	0,07	0,07		**25,05**
2, 4				2,42	1,45	4,56	1,36	0,52	7,88	1,75	0,96	2,88	2,57	0,89	1,16	1,65	1,07	0,98	**32,08**
2, 5		11,71	14,18	10,96	9,46	3,54	2,66	7,82	6,81	7,74		1,29	2,86	0,72	3,22	0,24	0,54	0,03	**83,78**
4, 2		38,43	23,22	8,54	10,14	0,67			0,21			0,12	0,26	0,72	0,66	0,78		0,03	**83,78**
4, 9	0,12	41,43	11,17	2,97	6,85			0,72		3,84	0,33	4,29			3,30	2,52		0,94	**78,48**
5, 2	0,54		12,69	2,84	3,76	0,50	1,10		1,22	2,62	0,60	2,14	1,19	0,08	0,44	0,84	0,95	0,59	**32,08**
6, 0	2,29	15,91	27,38																**45,58**
6, 3	24,54	48,05	7,71	18,78	18,32	8,40	18,58	7,35	13,28	33,99	12,02	3,83	2,30	1,60	3,22	8,59	3,68	0,54	**234,77**
6, 4	67,02	79,86	34,39	23,43	23,49	8,29	9,29	0,67	9,20	8,91	1,20	3,74	3,83	2,41	5,61		1,77	0,60	**283,72**
7, 5	1,63	4,96	22,81	17,02	4,35	4,22	11,10	8,02	6,01	6,79	6,86	2,59	4,51	4,16	2,00	1,52	2,26	0,97	**111,78**
7, 8		24,51	24,12	30,95	10,94	0,53	4,79	1,84		2,28		0,54			3,04	2,13	1,88	0,36	**107,90**
8, 9		11,17	5,06	6,56	7,40	9,65	17,51	2,66	17,35	13,91	1,47	2,58	1,93	1,49	4,22	1,99	2,06	0,88	**107,90**
9, 0			16,24		2,63		1,07		1,47		0,02		0,18	2,57	0,13	1,62	1,29		**27,22**
Total	**96,14**	**276,04**	**198,98**	**126,58**	**99,74**	**48,17**	**82,84**	**30,33**	**73,36**	**84,28**	**34,04**	**28,14**	**29,31**	**16,02**	**28,12**	**26,90**	**16,97**	**5,93**	**1301,88**

Table 11: Inventory levels (Single-node aggregation)

Edge \ Period	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Total	Inventory Capacity
0	2,29	18,20	61,82	59,69	61,36	53,56	39,25	38,53	30,05	27,59	17,03	12,89	3,38	4,56	3,55	0,17		**433,91**	**106,88**
2	0,54	27,27	48,99	47,00	49,99	43,06	40,13	31,79	18,54	11,67	11,31	9,41	5,43	4,62	1,33	1,06	0,39	**352,52**	**76,16**
3	24,54	72,59	80,30	80,30	80,30	72,90	73,15	66,52	59,39	80,30	79,20	66,17	59,01	46,17	34,16	24,91	15,38	**1015,27**	**80,30**
4	66,91	66,91	66,91	66,91	66,91	65,24	66,58	56,52	66,91	64,43	52,82	44,99	38,96	30,33	24,34	15,49	9,17	**870,32**	**66,91**
5	1,09	17,76	42,06	57,70	57,70	54,32	57,70	57,70	57,10	57,70	49,97	41,87	38,83	31,75	26,37	16,22	7,93	**673,79**	**57,70**
6	97,82	101,25	101,25	101,25	101,25	101,25	95,63	93,93	80,50	38,91	27,70	22,91	20,16	17,33	8,86	5,19	1,14	**1016,35**	**101,25**
7	111,30	170,91	132,63	88,78	78,39	73,64	57,75	47,90	41,89	32,83	25,96	22,83	18,33	14,17	9,13	5,47	1,33	**933,25**	**170,91**
8		13,33	32,39	56,78	60,32	51,19	38,47	37,65	20,30	8,67	7,20	5,16	3,23	1,74	0,57	0,70	0,52	**338,21**	**72,10**
9	0,12	52,72	52,72	52,72	52,72	49,74	52,72	47,15	50,28	52,72	52,72	52,72	50,73	37,97	33,15	23,67	8,25	**722,82**	**52,72**
Total	**304,60**	**540,94**	**619,07**	**611,13**	**608,95**	**564,91**	**521,40**	**477,69**	**424,96**	**374,81**	**323,91**	**278,96**	**238,06**	**188,62**	**141,45**	**92,90**	**44,12**	**6356,46**	

Table 12: Flows (Non-aggregated)

Edge \ Period	0	1	2	3	4	5	6	7	8	9	Total
0, 6			0,18	1,81		0,41				0,08	2,48
2, 4				2,95	8,50	11,73	1,69	0,08	0,06	0,52	25,52
2, 5		8,36	17,15	22,10	11,62	4,59	5,49	0,17	1,56	3,06	74,11
4, 2	10,08	25,91	25,05	1,58	5,28	3,95	0,00	0,17	0,04	2,05	74,11
4, 9	3,96	24,58	23,19	14,58	13,00	24,14	9,18	13,07	0,09	2,90	128,69
5, 2		1,11	3,03	16,08	2,26	0,72	1,69	0,08	0,06	0,50	25,52
6, 3	31,16	33,95	54,47	4,18	2,71	6,12	1,66	1,13	1,53	7,49	144,41
6, 4	29,52	92,89	64,11	46,37	32,69	44,67	12,35	6,03	0,44	14,28	343,36
7, 5		1,78	17,60	50,53	17,59	0,25			0,66	0,54	88,95
7, 8		1,25	11,86	57,58	13,97	11,33			1,65	0,67	98,30
8, 9		1,25	11,86	17,31	16,58	19,31	26,00	3,00	0,61	2,38	98,30
9, 0			0,26	1,81		0,33				0,08	2,48
Total	74,72	191,07	228,76	236,87	124,21	127,56	58,07	23,73	6,70	34,54	**1106,23**

Table 13: Flows (Unrestricted aggregation)

Edge \ Period	0	1	2	3	4	5	6	7	8	9	Total
0, 6			19,77	16,03	2,17	3,08		0,40	0,19		41,63
2, 4				10,88	5,28	12,36	20,66	0,15	0,14		49,47
2, 5		28,70	10,65	28,59	16,58	18,32		1,70	0,28		104,82
4, 2	28,70	24,94	28,59	2,29	1,68	16,64		1,70	0,28		104,82
4, 9			65,73	18,16	6,38	20,12	36,02	1,23	13,21	0,23	161,09
5, 2			2,09	26,24	6,33	6,35	8,16	0,15	0,14		49,47
6, 3		59,45	44,14	24,77	9,56	3,08	1,27	1,96	0,19		144,41
6, 4	52,98	75,09	93,83	42,54	25,69	53,85	24,93	7,85	5,51	0,23	382,51
7, 5			20,61	38,14	4,41	14,83	3,27	0,30	0,65		82,20
7, 8			31,44	47,56	19,14	4,00	2,92				105,06
8, 9			19,77	26,80	25,36	26,08	3,00	0,69	3,35		105,06
9, 0			19,77	16,03	2,17	3,08		0,40	0,19		41,63
Total	81,68	188,18	356,40	298,03	124,75	181,79	100,23	16,53	24,11	0,47	**1372,17**

B. Figures

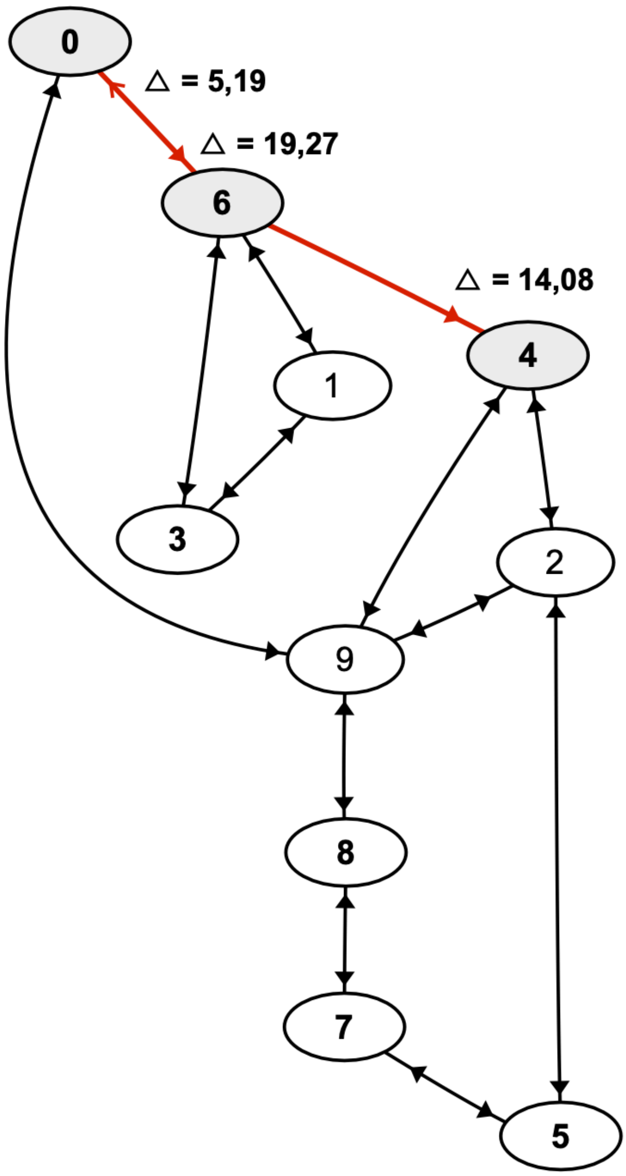

Figure 38: Flow difference in the single-node aggregation

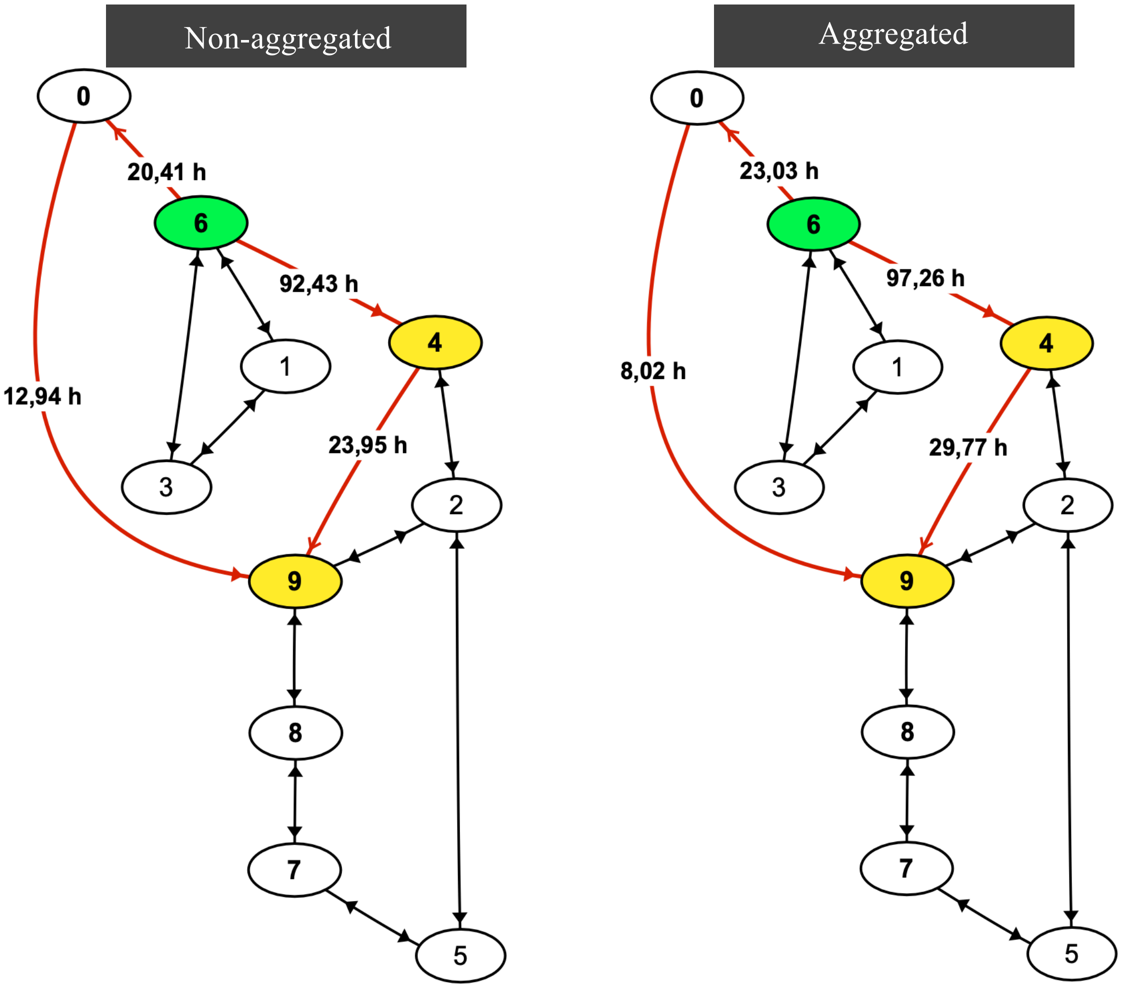

Figure 39: Transport time [Path: 6-9] (Non-aggregated vs. Aggregated)

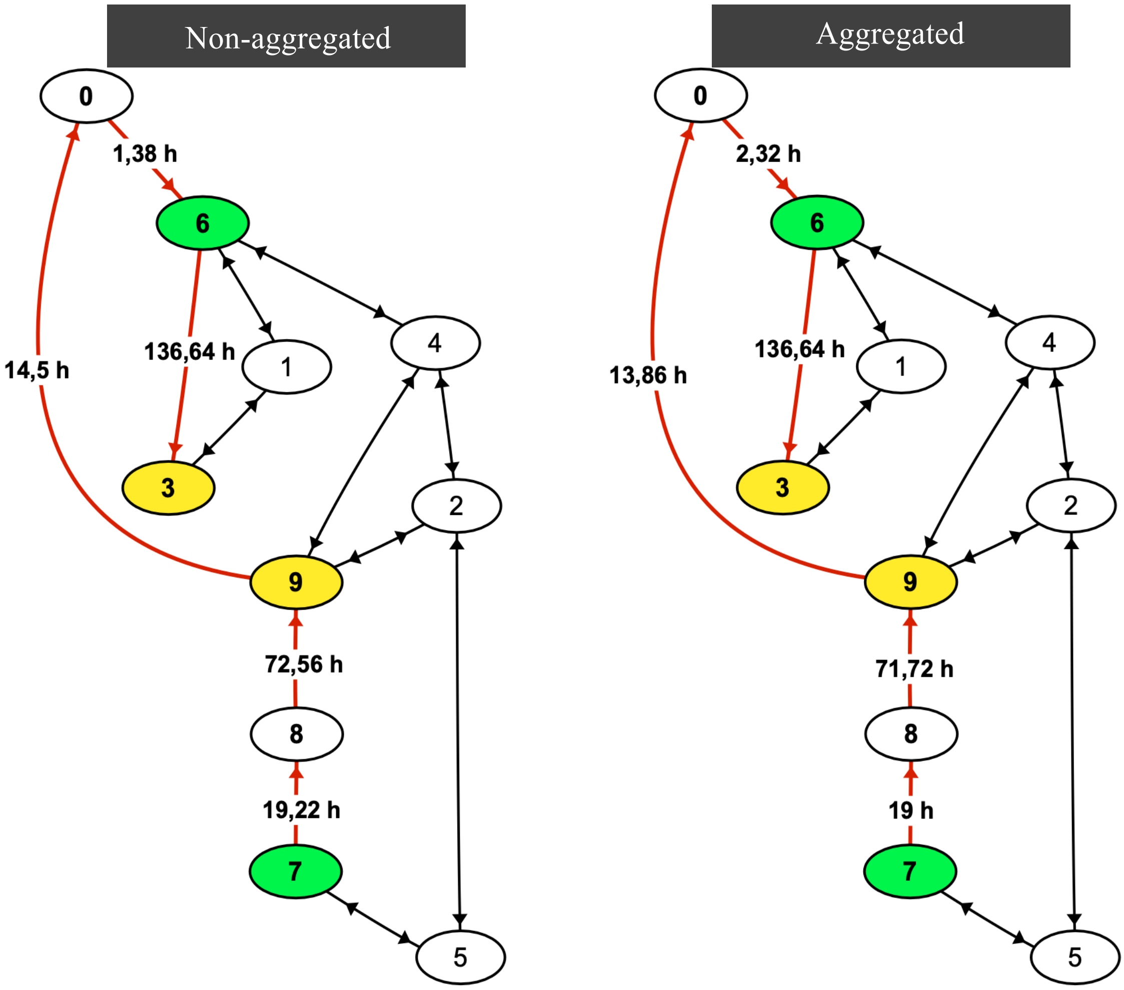

Figure 40: Transport time [Path: 7-3] (Non-aggregated vs. Aggregated)

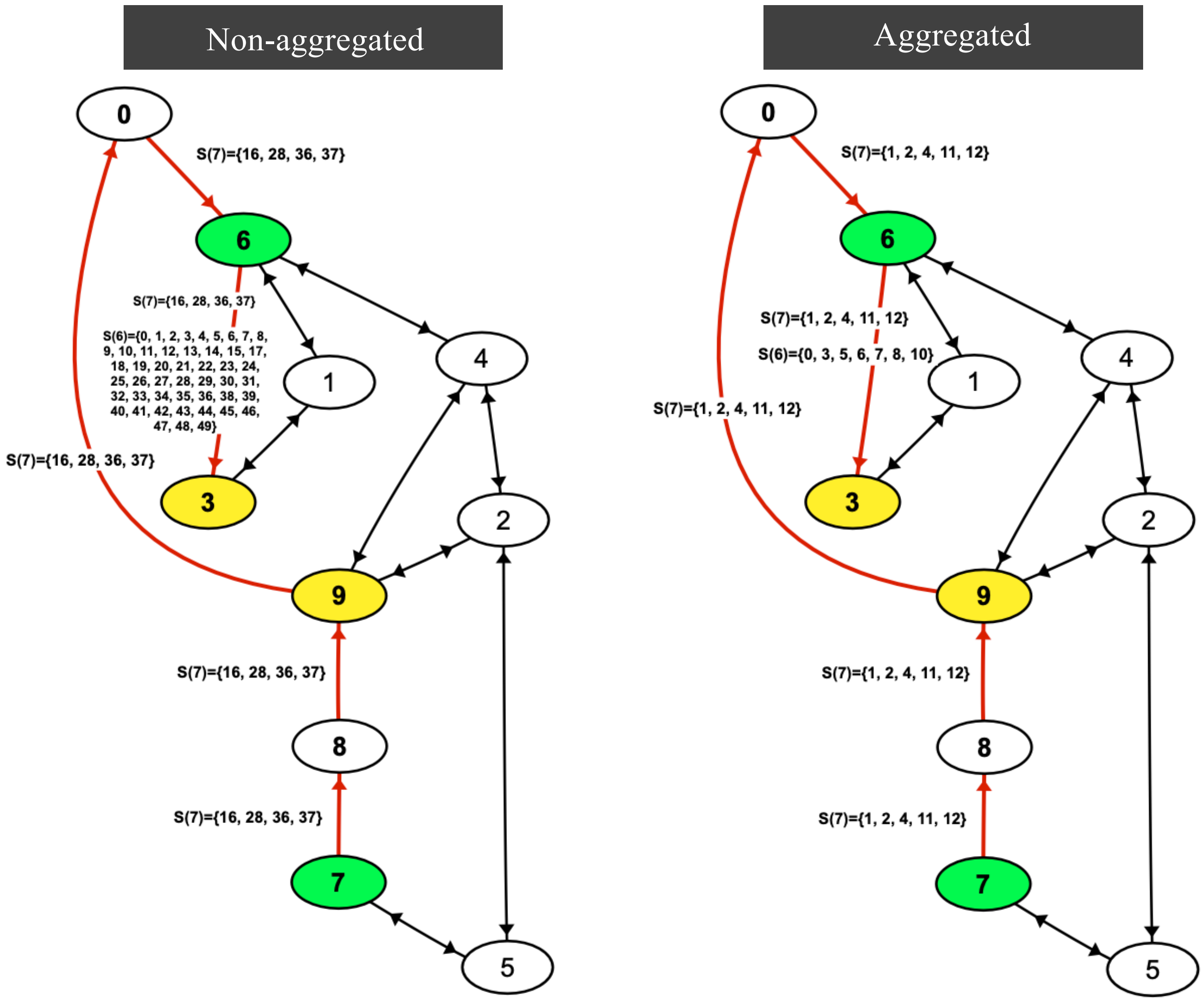

Figure 41: Commodity analysis [Path: 7-3] (Non-aggregated vs. Aggregated)

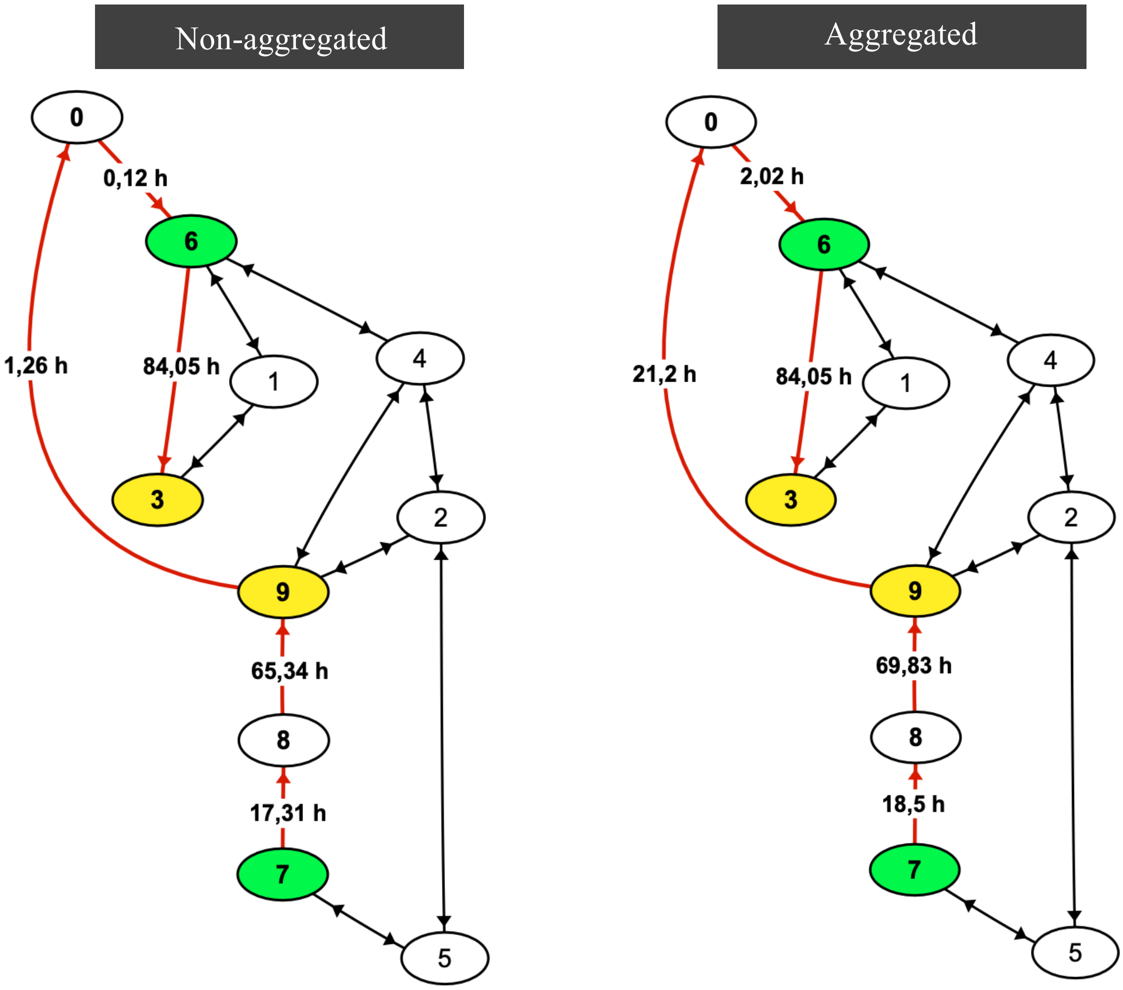

Figure 42: Transport time [Path: 7-3] (Non-aggregated vs. Aggregated)

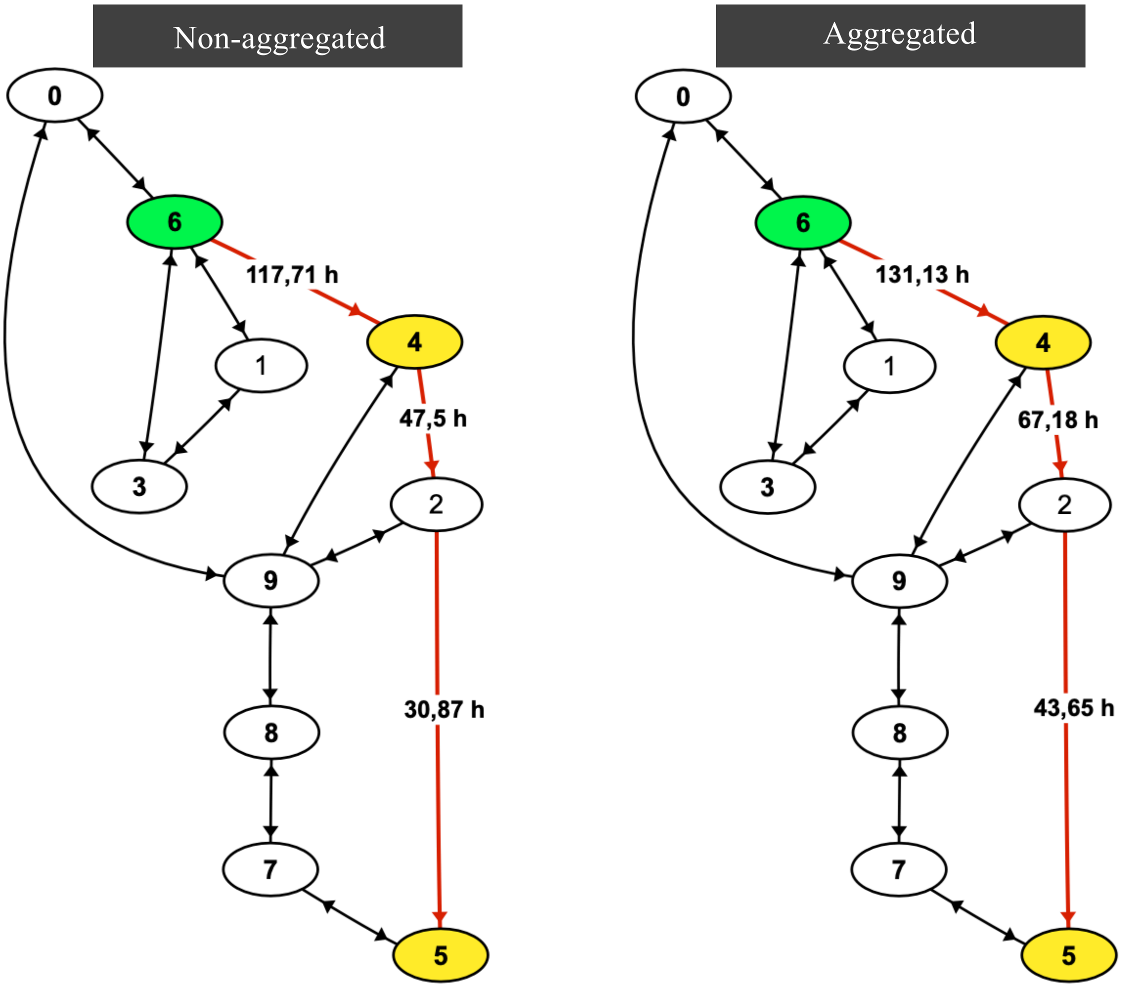

Figure 43: Transport time [Path: 6-5] (Non-aggregated vs. Aggregated)

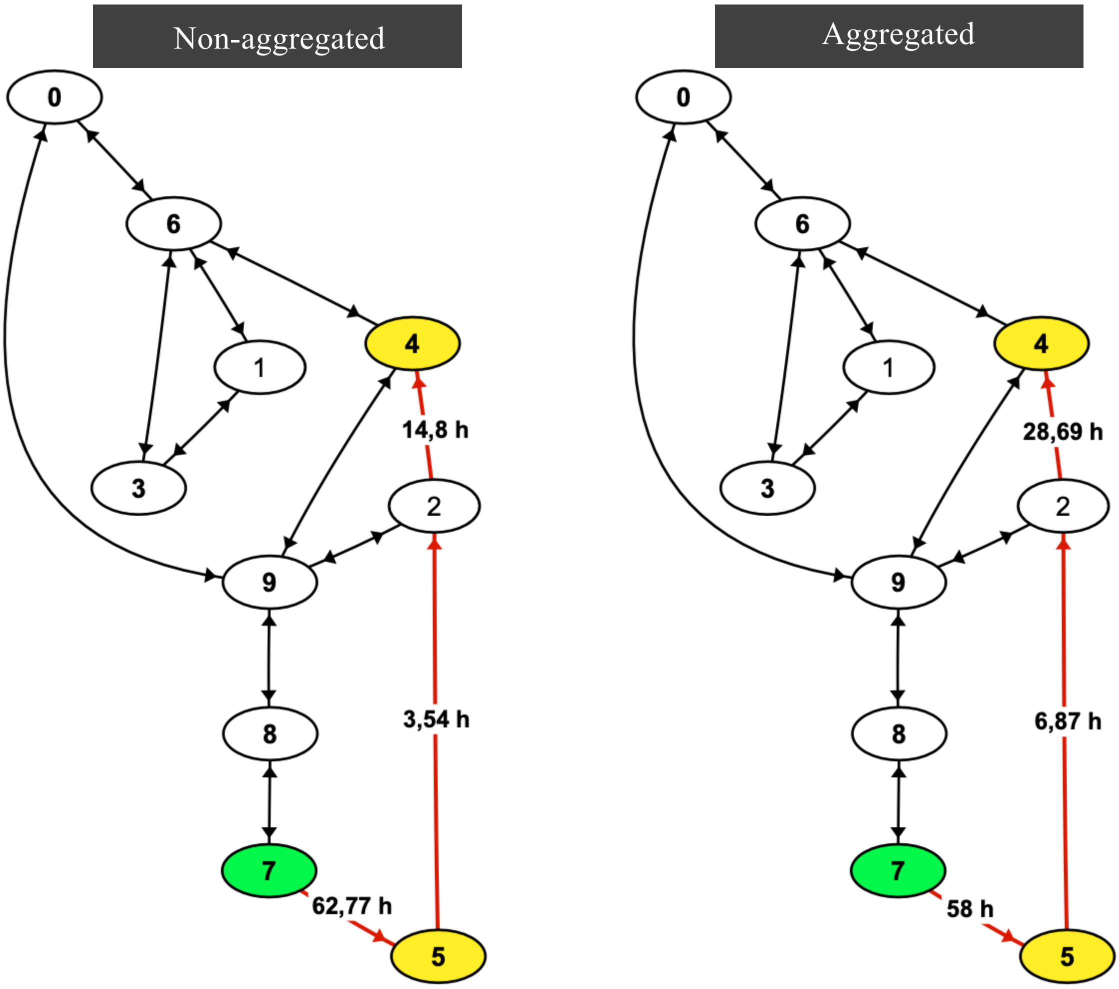

Figure 44: Transport time [Path: 7-4] (Non-aggregated vs. Aggregated)

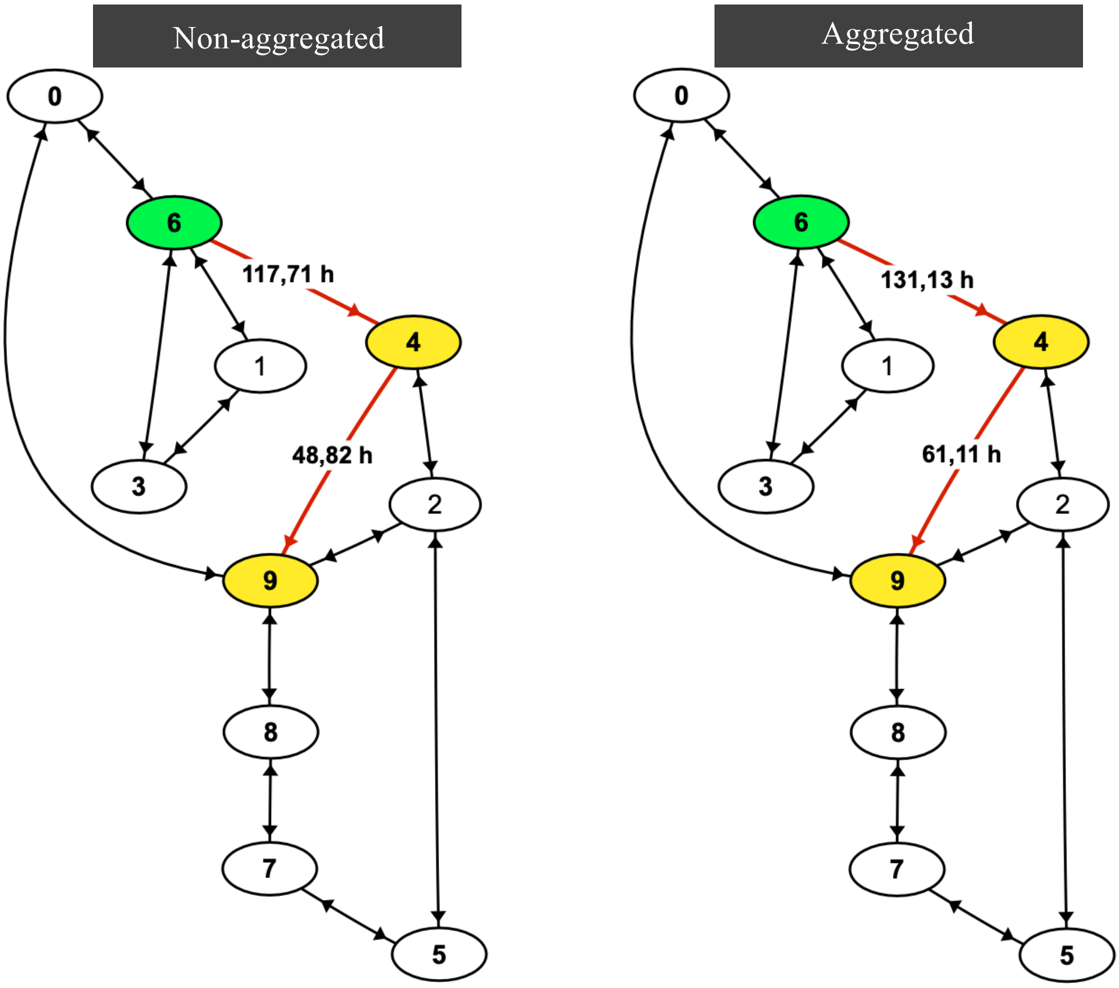

Figure 45: Transport time [Path: 6-9] (Non-aggregated vs. Aggregated)