Delft University of Technology
Faculty Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Container-to-mode assignment on a synchromodal transportation network: a multi-objective approach

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulllment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

Max Roberto Ortega Del Vecchyo

Delft, The Netherlands
July 2017

MSc THESIS APPLIED MATHEMATICS

Container-to-mode assignment on a synchromodal transportation network: a multi-objective approach

MAX ROBERTO ORTEGA DEL VECCHYO

Delft University of Technology

Daily Supervisor and responsible professor

Dr. D.C. Gijswijt

Supervisors at TNO

Dr. F. Phillipson

MSc. A. Sangers

Other thesis committee members

Prof.dr.ir. K.I. Aardal

Dr. D. Kurowicka

July, 2017 Delft

Abstract

In this report we present an interactive multi-objective optimization tool that was developed as part of this graduation project. This appliance is meant to be used as a decision support tool for transportation planners working on a synchromodal transportation network on the container-to-mode assignment, where different attributes are considered important. The tool offers the planner a range of solutions according to her/his preferences, and offers the opportunity to seek for new ones if the planner is not satisfied with the solutions found so far.

Before presenting the tool, a framework for synchromodal transportation problems is introduced (which was developed as part of a collaborative work with two other students and two supervisors). Then an analysis is done on mathematical modeling approaches of container-to-mode assignment, with special emphasis on computational time, given the time-sensitivity of this problem on synchromodal transport networks. From this analysis a model is chosen on which to built upon the multi-objective optimization tool, for which a thorough analysis on the attributes to be considered is carried out.

Preface

Looking back at the last two years of my life, I'm sure I will always remember them as very prosperous times. I had the fortune of living them among great people, I've learnt a lot of mathematics, and much more.

I am forever grateful for the Justus en Louise van Effen Foundation. Thanks for giving me the opportunity of a lifetime.

Thanks to Dion, Frank and Alex, for the guidance and patience. This work would not have been possible without their dedication and interest.

I am in debt with the professors and staff members from TU Delft I have encountered in my path. Their passion and will to do well their job made my experience incredible. For me it is clear that TU Delft is interested in the development and well-being of their students.

To all my family and friends, from here and there, and my girlfriend. For all the good times behind us, and those to come.

Contents

1. Introduction

In recent years, freight transportation planning is going through a transformation in practical and theoretical terms. This revolution is referred to as synchromodal transportation, the paradigm that is sought for in many logistics companies. The goal of this transition is, simply put, to maximize the efficiency of the transportation network. In theoretical terms, synchromodality implies taking into account uncertainty, and being able to process new information as it becomes available. This new information may imply new planning decisions that must be done quickly, since a newly chosen planning decisions may alter the distribution of freight at any time. This poses one of the challenges on synchromodality transportation that is addressed in this work: How to model container-to-mode assignment problems on a synchromodal transportation network in a time-efficient way?

The goal of synchromodality is to maximize efficiency of a transportation network, but the efficiency of a network is not a clear-cut concept, nor should it mean the same for different stakeholders: for the perspective of a customer, the timely delivery might be the decisive factor, whereas for a logistics service provider, both timely delivery and transportation costs are important. In practice, a transportation planner takes other attributes into account when making his decisions, based on his knowledge about the uncertainties of the network. The literature provides ways in which to optimize the flow of network with regards to cost, and some effort has been done in considering other attributes by adding them to the same objective function. However, this simplification forces the attributes to be comparable with each other in a specific predetermined way. Also, the single-objective optimization result will give a single optimal solution, which excludes the possibility of using the expertise of the decision maker to compare different possible solutions, and to explore the problem. In the central part of this work, we study, analyze and compare the different performance indicators that are in play on a synchromodal transportation chain, and we develop an interactive tool based on multi-objective optimization, which is meant to be used as a decision support tool for a transportation planner. This tool gives a range of solutions that considers the attributes found in the study, and finds new solutions attempting to fit the criteria and conditions that the decision maker desires.

Additionally, the literature study revealed that the obstacles on synchromodal transportation are overcomed via different approaches and with different levels of complexity; this is a consequence of the different branches of expertise involved in synchromodality, from where the need to categorize synchromodal transportation problems was felt.

Thus this study addresses the following questions:

- *How to consider different important performance indicators on a synchromodal transportation network?*
- *What to optimize on a synchromodal transportation network on a container-to mode assignment?*
- *How to model container-to-mode assignment problems on a synchromodal transportation network in a time-efficient way?*
- *How to categorize synchromodal transportation problems?*

On this first chapter we present the basic definitions concerning synchromodal transportation, the context and the motivations for our research. On Chapter 2 we present the relevant literature study for our problem, including an overview of synchromodal transportation problems, and the mathematical part of multi-objective optimization. Chapter 3 was done in collaboration with Myrte de Juncker, Dylan Huizing, Frank Phillipson and Alex Sangers, it proposes a framework to categorize synchromodal transportation problems on a tactical-operational level. Chapter 4 presents the minimum cost multi-commodity flow problem, an optimization problem used to model network flows, and Chapter 5 studies this optimization problem in our context, as well as comparing different approaches of this model, from this chapter a model is chosen to work with for the next chapters. Chapter 6 studies and analyses different attributes considered important in synchromodal transportation networks, as well as constructing a mathematical basis from the concepts, and adapts them to the best-suited model from the previous chapter. Chapter 7 presents the multi-objective optimization tool and gives an example. This last chapter relies on the mathematical formulations derived on the previous chapter.

1.1. **Context of the project and motivations.** This graduation project is part of a 5 year NWO project about predictive synchromodality referred to as COMET-PS. In this project, research institutes and logistic service providers are gathered to find solutions to the current obstacles of synchromodality. I had the opportunity to be involved on this COMET-PS via TNO for a period of 10 months (as an intern and as part of my graduation project). As a consequence, I had the fortune to be in contact with parties working in the development of synchromodal transportation networks, both in theoretical and practical terms. This diverse view has shown me some of the obstacles existing in both sides, as well as the connections that need to be constructed between these expertises. The contact with all these parties provided me with remarks that served as motivation, such as:

- Transportation planners are "stochastic calculators" that make decisions based on their experience. The problem of assigning a route is far more complex than just assigning the cheapest route for each incoming order. Mainly because of the uncertainties of the network. So far, models in the theory are effective at giving the least costly solution for a given problem, but this not enough. How to quantify for these uncertainties? What are the other attributes considered when making a decision?
- Planners want to make a choice of a group of possible planning solutions for them to choose from, (this is different from the current approach in theory, where for each problem, only one solution is given).
- The needs of a customer change per customer, for example, containers do not necessarily have to arrive on time, this depends per customer.
- The costs incurred for transportation are done per day or per trip. This is very different from what is usually done in theory, where costs are incurred per container (as it is done in models of cost of flow).

The last two points also being mentioned on literature [23].

1.2. **Definitions .** Synchromodality is quite an ambiguous term, and such is the case of other logistics transport paradigms: more often than not, there is no consensus on the definitions, or on the features that characterize each of the paradigms. Therefore, it is necessary to clarify what we mean by each term. These definitions are based mostly on those given in [25] and capture what we believe to be the essence of each of them:

- A transportation network is called a **multimodal transport network** if the transportation of goods can be made via different modes, where a **mode** is understood as a mean of transportation, such as a barge.
- An **intermodal transportation network** is a multimodal network where the goods are transported through a standardized unit of transportation, which we call **freight**, and in practice is usually a **container**.
- A **synchromodal transportation network** is an intermodal transportation network where the real-time information is used to update the flow of freights to optimize efficiency.
- A **hub** is a terminal in the transportation network from which transportation decisions can be made

About the definitions above, we remark that synchromodality requires that freights have flexibility in terms of the modes they can use to reach their destination. Also, by *information* on the definition of a synchromodal transport network we mean awareness of anything that can affect the behavior of the network: traffic conditions, weather (it can change travel times of certain modes), arrival of new freights, decisions of other operators on the network, etc.

Other definitions of interest are:

- A **resource** is considered as one of the objects that can provide transportation, that is, a specific barge, train or truck (not the same as mode).

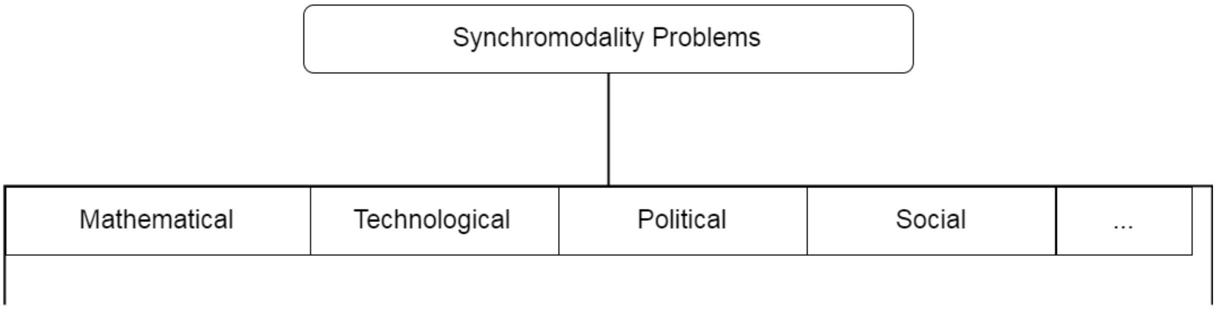

Figure 1.1. Synchromodality Problems

- A **trip** is defined as the moving of a resource from one location (origin) to another (destination) with no intermediate stops, where freight can be allocated or dismantled at the origin or destination.
- An **order** is a number of containers that have a specific origin and destination, with a specified time of release on the origin, and deadline.

The goal of synchromodality is to improve the overall performance of the network. Although it still remains to specify what we mean by 'improve' and 'performance', we hint to some potential benefits of synchromodality which are considered improvements of the performance of the network: The constant feed of new information can be used to ensure that the plan being carried over at a given instant is still a good plan, or even a feasible one. For example, weather conditions might disable a certain trip, or arrival of a new order can be combined economically. In this respect, synchromodality seems like a reasonable concept, in fact, real-time changes have been done by planners of logistic service providers way before the concept synchromodality was introduced.

The impact of synchromodality can be assessed by the success of some pilot projects such as the Rotterdam-Tilburg case [35] whose positive outcome has made it often referenced on the literature about synchromodal transportation ever since. Some claim that the motivation of synchromodality was that after the crisis of 2008, logistic service providers were pushed to decrease costs while maintaining or improving service quality. However, several problems with a synchromodality essence have been studied since at least 1988 [7], although by then the word "synchromodal" had not been established yet.

1.3. **Challenges in synchromodality.** Synchromodality needs to overcome several obstacles from a wide range of disciplines. In [22] seven critical success factors of synchromodality are discussed:

(1) Network, collaboration and trust
(2) Awareness and mental shift
(3) Legal and political framework
(4) Pricing/cost/service
(5) ICT/ITS technologies
(6) Sophisticated planning
(7) Physical infrastructure

The first crucial factor refers to the cooperative behavior network operators must assume in a practical implementation of synchromodality: on a large-scale distribution network, information must be given and received from competing logistic operators. The second factor is related to showing logistic service providers about the benefits of synchromodality. These two factors are usually presented as two of the major challenges of synchromodality, and efforts have been done in order to foster the cooperative mindset needed [28]. For a more thorough explanation on these factors see [22].

From the list above, we can identify several different branches of knowledge involved in each of the critical factors (see Figure 1.1). Roughly, we could say that the first and second factor

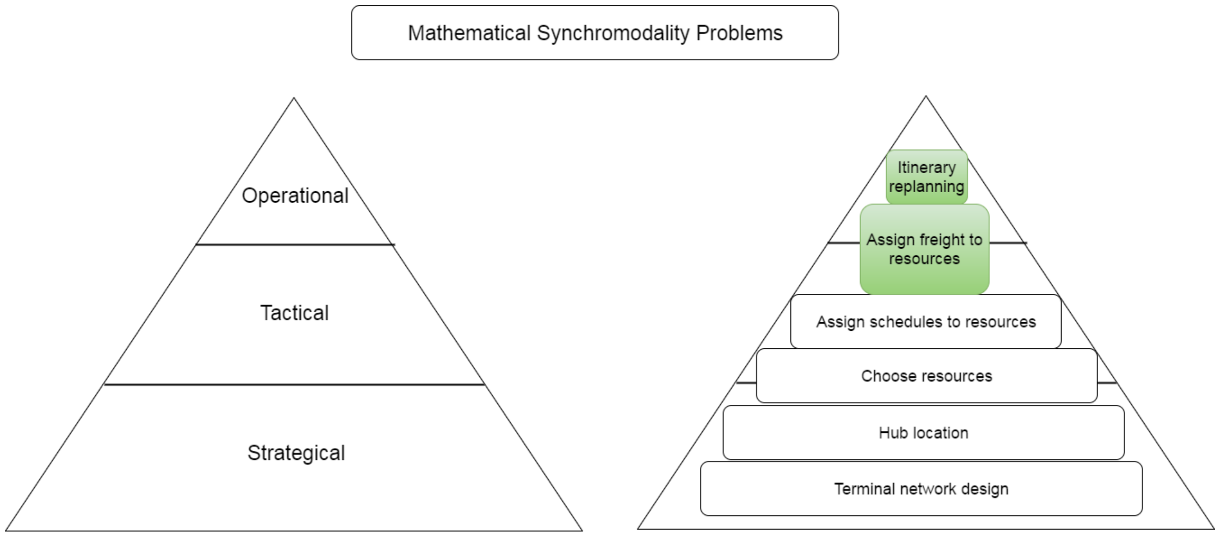

FIGURE 1.2. Mathematical Synchromodal Problems. The problems in green are those within the focus of this work

are mainly social problems, the third is a political problem, the fourth is mathematical, social and political problem, the fifth is a technological problem, the sixth is a mathematical problem, and the seventh is a technological problem. In this work we are interested in those challenges within the realm of mathematics. These challenges are therefore directly related to the 6^{th} and 4^{th} success factor, but can also impact on the first three in an indirect way, by promoting the effectiveness of synchromodality.

The basic mathematical branches that deal with synchromodal problems are graph theory and optimization (these are very natural tools of choice given the logistic nature of the problem), but knowledge and methods of other branches such as stochastic processes, systems and control or statistics are used frequently. Broadly speaking, mathematical synchromodal problems (and other non-mathematical problems) can be divided into three categories depending on the time horizon to which they correspond:

Strategical. Strategical problems refer to those that represent an investment on the transport chain, such as the construction and location of hubs, or acquisition of new resources. These problems will impact the transport chain on the long run, which is why they are considered as long time horizon problems.

Tactical. Tactical problems on a transport chain refer to those on a median time horizon (days to weeks): the planning of the schedule of the resources and the assignment of containers flows in the network fall within this category.

Operational. Operational problems are the ones related to the real-time aspect of transport planning (minutes to hours), for example, itinerary replanning and assignment of individual containers to resources.

We can see that these problems are related in a pyramidal-like structure (see Figure 1.2) in the following sense: tactical problems are usually engaged where a specific strategical instance is given, and operational problems are frequently solved where a strategical and tactical structure is fixed, although sometimes problems in two consecutive levels are solved simultaneously, for instance, in [2] the frequency of a resource is determined along with the flow of freight (that is, part of the schedules to resource and the freight to resources are solved at once). For further reading, the paper [25] is considered a good place to start for a broad view of multimodal transport problems.

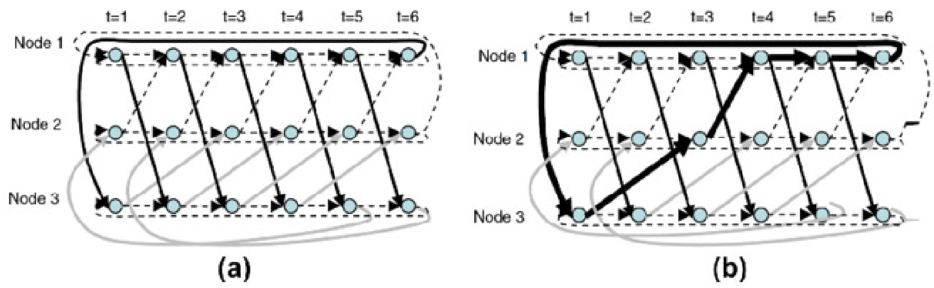

FIGURE 2.1. Space-time network taken from [25]

2. Literature Review

2.1. **Synchromodality.**

"Models should be formulated to address specific problems of specific stakeholders at specific levels" [2]

In this section we discuss the mathematical approach to the challenges exposed in the previous section. Then we observe where our problem lies within this general overview.

Strategical. Hub location problems have been quite extensively studied in the literature for many different transportation paradigms. In graph theory terms, these problems are concerned with the location and position of arcs and nodes, where nodes represent hubs (or different mode-specific parts of a hub) and arcs represent the different connections between (sections of) hubs. For intermodal transportation, in [12] the added complexity of the different modes is also addressed, and the problem is formulated as a Mixed Integer Linear Program (MILP) which, given its complexity, is solved via a tabu search meta-heuristic. On [15] it is mentioned that "there are currently no published works that explicitly analyze strategic planning problems in synchromodal freight transport". This might not be so surprising, given that the difference between synchromodality and intermodality lies specifically with the use of real-time data, and the strategical problems are concerned on long-time run, therefore there might be little difference between analyzing a synchromodal strategic problem and an intermodal strategic problem. However, the strategical problems lie outside the scope of this work.

Tactical. In the literature, Service Network Design (SND) is the most common concept on which models are built upon in order to tackle tactical problems. In the broadest sense, a Service Network Design is represented via a directed graph where arcs represent a change of state and nodes represent different locations. SNDs are further subdivided into Static SND (SSND) and Dynamic SND (DSND). As the name may suggest, SSND and DSND vary in the fact that on DSND some features vary over time, for example, the travel times. The DSNDs are those that can be represented via a space-time network, that is, we consider a directed graph (G, A) where each node represents a station of interest (terminal, hub, mode terminal) and each arc ab represents a trip from a to b with time travel tt_{ab}. For a fixed time horizon T, certain time-stamps are marked that subdivide the interval $[0, T]$, and for each time-stamp t a copy of the nodes from G is made, so that the set of nodes is $\{a_t | a \in G, t \text{ is a timestamp}\}$ and a trip from a to b is modeled via an arc from a_{t_0} to b_{t_1} where $t_0 < t_1 < T$ and the travel time of this trip is $t_1 - t_0$. Figure 2.1 shows an example of a space-time network. This kind of representation is useful for analysis of flows of freights in the network, and to show the characteristics of the trips available. We will be working with a DSND in the rest of this work, but we refer to it simply as space-time network.

Problems formulated via DSND can be quite difficult due to the size of the set of variables (which increases rapidly as instances become bigger, and time-stamps more abundant). Additionally, one should not be deceived by the seemingly simple definition of a DSND, the features mentioned above are only the elementary building blocks. As DSNDs become increasingly exhaustive, the set of constraints and variables grows rapidly (see [15] pp. 24-34) and thus the complexity and computational demand grows. As mentioned in [25] "Due to the complexity of these problems, heuristic and meta-heuristic solution methods are the prime choice". Some examples of solution methods are: approximate dynamic programming [20] and two-stage stochastic programming [1].

Apart from the DSND structure, models in literature have little in common: most of the literature is case-specific on real-world problems, which means algorithms are customized and models are done emphasizing what is of importance for the problem at hand. For example, in [25] pp. 8 transshipment operations are considered within the model, which is fairly uncommon for models about this problem. In some papers, uncertainty is incorporated to a factor of the DSND, usually on the arrival of demand items [25, 20]. However, other authors use the term Stochastic SND to indicate the presence of a stochastic element. Another approach that has been used to model tactical synchromodal transportation problems is model predictive control [15]. This branch is used to represent the behavior of complex dynamical systems, and is commonly applied in the modeling of process industries.

Operational. As can be seen form Figure 1.2, the problem of assigning freight to schedules is considered somewhere in between the operational and the tactical part. Usually, in a tactical perspective this problem is referred to as flow of containers, whereas in an operational perspective, the assignment of resource to container is done per container (Although many authors have this view [15], other authors regard the problem of flow of containers the same as the container to resource assignment problem [20]). This hints that, apart from the difference in the time-horizon, it is also a matter of the number of containers being dealt with. When addressing operational transport planning problems, on [15] pp.5 it is noted that:

> The operational freight transport planning problem faced by intermodal freight transport operators is in general a mixed integer optimization problem in which individual containers are directly modeled and scheduled in the planning. This problem is NP-hard and requires huge computational efforts to solve as the number of the shipments increase,

Indeed, most engagements into the operational problem in the literature share the same computational conflicts addressed in the tactical part. The addition of stochastic elements can be tackled via simulation methods which makes the problem deterministic, but comes with a computational trade-off [14]. This suggests that by improving the computational efficiency of the deterministic case, the stochastic case is also indirectly being addressed.

There is acknowledgment that attention must be given to the stochastic elements of a transportation network [23] and many recent papers exist on this matter, for example, [14, 20]. However stochastic arrival of elements have been studied since 1988 by Crainic [8].

In this work, we deal with the assignment of containers to resources, so our problem definition has more of a tactical/operational nature. The Figure 1.2 shows where our problem lies with respect to the synchromodal problems.

2.2. **Cost Function.** In the literature on multimodal transport, the main performance indicators are cost and service time, the latter meaning the arrival of freights on time. Many other attributes are often mentioned but barely studied [25].

As it is stated in [30], many transportation planning problems are solved via a deterministic optimization-based tool where the lowest-cost solution is chosen. This solution approach is very efficient, but has two major flaws:

(1) The plan relies on all input being deterministic, which in turn relies on forecasts or estimates that can be very inaccurate. This is a simplification in order to deal with the future uncertainties of the transportation chain
(2) If these initial assumed values turn out to be realized differently, then the value of the originally optimal solution may change drastically, or might be unfeasible!

In the literature these problems are sometimes addressed in papers via different terms: reliability, flexibility, robustness and resilience are all used and although most readers will have an intuitive feeling for their meaning, and in the literature of transportation networks these concepts are often mentioned, there is no consensus in what they mean, which inevitably makes them lean towards

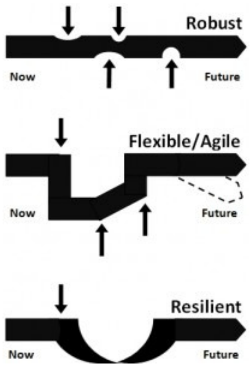

FIGURE 2.2. Concepts on transportation planning processes according to [11]

being buzzwords instead of the key attributes they were intended to capture in the first place. For most authors, however, they still are somewhat related, and are a crucial part of the whole transportation process. From these concepts, perhaps the ones with the most stable meaning throughout the literature (and also the most often mentioned) are flexibility and reliability. Reliability is usually understood as the ability of a plan to arrive on time on its orders, whereas flexibility is defined as the ability of a plan to adapt to uncertain events in the future.

In [11] definitions are proposed as an attempt to encompass consistently the meaning intended in other papers for each concept (See Figure 2.2):

- Robustness is the ability to endure foreseen and unforeseen changes in the environment without adapting.
- Flexibility is the ability to react to foreseen and unforeseen changes in the environment in a pre-planned manner.
- Agility is the ability to react to unforeseen changes in the environment in an unforeseen and unplanned manner.
- Resilience is the ability to survive foreseen and unforeseen changes in the environment that have a severe and enduring impact.

All of the above definitions are, of course, in the context of transportation planning processes. In this work, we will use the following definitions based meanings explained both implicitly and explicitly on several sources such as [11, 30, 19, 25, 3]:

- Robustness is the capacity of a plan to overcome uncertain events or disturbances in the future and still be carried over as planned.
- Flexibility is the capacity of a plan to adapt to uncertain events or disturbances, when these force the plan not to be able to be carried on anymore.
- Reliability is the capacity of a plan to arrive on time to its destinations on time.

The research question "what to optimize?" has been scarcely addressed in the literature, despite it being a recognized problem [12, 23]. In [12] it is proposed that cost, service, frequency, service time, delivery reliability, flexibility and safety are all performance indicators. In most papers, cost of the operation is the defined objective, and other attributes are neglected. However, in practice, planners do take into account other factors to make their planning decisions, which makes the planning problem quite complex. If a specific important and urgent order has two possible routes, where one is much more robust but slightly expensive than the other, it is not trivial which one the planner should choose. Also, in certain cases, an order is allowed to have a delay with no penalty whatsoever. This of course, is dependent on the order. In fact, logistic planners make many decisions based on the experience they have on the uncertainty of the network they interact with. In [3] different attributes for transportation networks were proposed, and an analysis was done in the importance of these different performance indicators for different instances. They define the following attributes:

- Time: door-to-door transport time
- Reliability: percentage of on-time deliveries

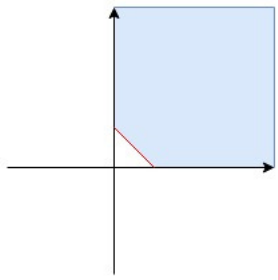

FIGURE 2.3. The feasible set (blue) and all Pareto solutions (red) for the example above

- Flexibility: percentage of non-programmed shipments that are executed without undue delay
- Cost: door-to-door transport cost
- Loss: percentage of commercial value lost from damages, stealing and accidents
- Frequency: service per week

The study concluded that in the majority of papers, cost is the most important attribute, followed by time. Frequency and loss are usually negligible. But perhaps more importantly, the study shows how the relative importance of each attribute highly depends on several different factors on the transport chain: Their study analyzed the case of 113 different firms with their respective transport network, and grouped them depending on their characteristics. It concludes that several factors such as the value of the goods transported, the average transport distance, the mode and the willingness to switch modes all play a part in the relative importance of each of the attributes. Quite likely, there are many other conditions that affect the choice of an optimal solution, this result motivates the usage of a multi-criteria analysis to our problem that provides a collection of possible optimal solutions, where the importance of the attributes is instance-dependent.

2.3. **Multi-objective optimization.** A multi-objective optimization problem is an optimization problem where there is more than one optimization objective, that is, several expression are desired to be optimized. In mathematical terms, if F is the feasible set, and $c_1, c_2, ..., c_p$ are objective functions from F to $\mathbb{R}$ that we seek to minimize, then the problem can be expressed as

$$\begin{aligned} &\min\ (c_1(x), c_2(x), ..., c_p(x)) \\ &x \in F \end{aligned} \tag{2.1}$$

One key difference between these optimization problems and the single-objective optimization problems, is that in the general case, there is no optimal solution that minimizes all objectives simultaneously. For a simple example, consider the set $F = \{(x, y) \in \mathbb{R}^2 | x+y \geq 1, x \geq 0, y \geq 0\}$, and the objective functions $c_1(x, y) = x$ and $c_2(x, y) = y$. Then the program

$$\begin{aligned} &\min\ (x, y) \\ &(x, y) \in F \end{aligned}$$

has not a single "optimal" solution: There is no way to compare $(0, 1)$, $(1, 0)$ and all elements $(\lambda, (1-\lambda))$ for $\lambda \in (0, 1)$. The following definitions state which solutions are regarded as optimal solutions for multi-objective optimization

Definition. Let P be a problem as (2.1). We say that an element $x \in F$ is dominated by $y \in F$ if we have $c_i(y) \leq c_i(x)$ for all $i \in \{1, 2, ..., p\}$, $c_i(y) < c_i(x)$ for some $i \in \{1, 2, ..., p\}$ and $x \neq y$. We say that x is a Pareto optimal solution (or simply Pareto solution) if it is not dominated by any $y \in F$. The set of all Pareto solutions of P is referred to as the Pareto frontier.

Any solution regarded as an optimal must be a Pareto solution, since if it weren't, the element that dominates it is a better choice. In the context of multi-objective optimization, the Pareto frontier is the set of all possible optimal solutions. Therefore, the solution to a multi-objective optimization is usually defined as the Pareto frontier. In the general case, however, this is not easy to obtain, and often not necessary, and so a solution can be interpreted as a representative set of the Pareto frontier. Of course, which set of solutions is representative varies per method. After a set of Pareto optimal solutions is given, a decision maker (DM) must chose one from this set of possible solutions, which is a fundamental difference with single objective optimization problems, where only one optimal solution is needed. The area of research that involves decision problems with multiple objectives is referred to as multiple criteria decision problems (MCDP) and the goal of the mathematical optimization model is to aid the DM by providing a set of solutions, from which the DM will chose according to his preference.

Some approaches to multi-objective optimization are the so-called *a priori* solution methods. In these methods, a preference of the objective functions is specified in advance. One example of an a priori method is a scalarization of the original problem. A scalarization is a single-objective version of the original multi-objective problem. This can be achieved in many different ways, for example, by assigning weights w_i to each objective and considering the problem

$$\min\, w_1c_1(x) + w_2c_2(x) + ... + w_pc_p(x)$$
$$x \in F$$

Or one may fix some bound values for certain objectives and consider

$$\min\, w_1c_1(x) + w_2c_2(x) + w_pc_p(x)$$
$$c_i(x) \leq \lambda_i \quad \text{for some } i \in \{1, 2, ..., p\}$$
$$x \in F$$

The latter means to constraint some of the objective functions. Notice that weights have to be determined in advance in order to formulate the problem, this means that a decision has to be done in advance about the importance of each objective, which is why scalarization is an a priori solution method. Also, not every scalarization gives a Pareto optimal solution

Another common a priori method is the *lexicographic method*, where objectives are ranked with respect to their importance, and constraints are added to obtain a solution that does not compromise the value of the most important objectives. That is, let x_1 be the solution to

$$\min_{x \in F} c_1(x) = y_1$$

Then let $c_2(x_2) = y_2$ where x_2 is the solution to

$$\min\, c_2(x)$$
$$c_1(x) \leq y_1$$
$$x \in F$$

And iteratively we obtain the Pareto optimal solution x_p of

$$\min\, c_p(x)$$
$$c_1(x) \leq y_1$$
$$...$$
$$c_{p-1}(x) \leq y_{p-1}$$
$$x \in F$$

Which is the solution proposed by the lexicographic algorithm. In many problems this solution method can be quite restrictive, since it does not allow for any loss of the highest ranked objectives. In general, a priori methods are useful because of their simplicity, but their characteristic feature can be a very important drawback, since it may be impossible to determine the importance of the attributes in advance. Also, only a single or few Pareto optimal solutions are provided. However, they can serve as the building blocks for other methods, such as the *interactive methods.*

Another kind of solution methods for multi-objective optimization are the *a posteriori* methods. This methods aim to give a representative set of the Pareto optimal solutions. Often this is achieved via a number of scalarizations, and many different ways to choose the scalarizations have been proposed, most of them useful for specific cases. Another characteristic of these methods is that they tend to propose an array of spread-out Pareto optimal solutions. This is useful in the case where the objectives are expected to have similar importance. However, in our case, we expect one of the objectives to be quite dominant over the others, namely, cost.

Perhaps the most interesting methods of multi-objective optimization for our case are the interactive methods. On these methods, the user is expected to have input on the algorithm to explore the solutions that are of interest. On [18] the main steps of interactive methods are explained in the most general sense. Briefly, these are:

(1) Provide the DM with the range that the different objectives can take, when possible.
(2) Provide a starting Pareto optimal solution(s) to the problem.
(3) Ask the decision maker for preference information.
(4) Generate new Pareto optimal solution(s), show them and other possible relevant information to the DM.
(5) Stop, or o back to 3.

The purpose of the first two steps is to get the DM to be acquainted with the possibilities and limitations of the problem at hand. The last three steps will also provide further insight to the DM, but are mainly geared towards finding the best Pareto optimal solution with respect to the DM preferences. Some highlights of these kind of methods are:

- The expertise of the DM is used as input on the method, which should give more satisfactory results from the point of view of the DM. The expert stirs the solution with respect to her or his preferences, and the method provides a solution towards these desired goals. Thus in this method the DM plays a very important role.
- A variety of solutions will be provided, which is a desired feature for our case.
- The decision maker does not need to know in advance the limitations of the problem with respect to the objectives. Rather, she or he learns from the problem at each iteration.
- There is no need to have preference for objectives in advance!

Therefore, interactive methods overcome the weaknesses shown previously of the other methods that are relevant to us. For this reason, in this work we develop an interactive method for our case. For an overview about different approaches on interact methods, and in multi-objective optimization in general, see [18]. Many approaches for each of the methods discussed above are proposed on the literature, even some hybrid methods have been developed, where the DM is first offered a range of solutions via an a posteriori method to get a large overview of the problem, and afterwards an interactive method to scroll through Pareto optimal solutions. In general, there is no solution method that is better than all others, rather the best method depends on the case at hand.

3. Framework

In this co-authored chapter by De Juncker, M.A.M., Huizing, D. and Ortega del Vecchyo, M.R. we introduce a framework by which to classify mathematical models described in the literature on synchromodal transportation problems. The reason for such a framework will be elaborated on in the next section, but briefly put, this framework should help researchers and developers by pointing towards solution methodologies that are commonly used in their problem instance.

Section 1 will describe the context and motivation of the framework. Section 2 will introduce the classification framework and Section 3 will introduce a short-hand notation for it. In Section 4, some examples are provided. In Section 5, these examples are used to discuss strengths and weaknesses of the framework.

The original work was edited in order to have continuity with the rest of the report

3.1. **Motivation.** As was mentioned before on Subsection 1.3, there are seven critical success factors of synchromodality discussed on [22], and several different branches of knowledge are involved in each of the critical factors (see Figure 1.1) since there are social, political, mathematical and technological problems. In this work we are interested in those challenges within the realm of mathematics. We have also observed that mathematical synchromodal transportation problems on a tactical or operational level are usually represented via tools from graph theory and optimization [25]. However, more often than not, the similarities end there: most of the models used to analyze a synchromodal transportation network are targeted to a specific real problem of interest [25], and knowledge and methods of other branches such as statistics, stochastic processes, or systems and control are often used. The models emphasise on what is most important for the given circumstances. Consequently, mathematical synchromodal transportation problems on a tactical or operational level have been reviewed with approaches that may differ in many aspects:

- The exhaustiveness of the elements considered varies, e.g. traffic conditions are considered in some models (such as the one presented in [15]) but not all.
- The elements that can be manipulated and controlled may vary, e.g. the departure time of some transportation means may be altered if suitable (as it happens in the model of [2]) or it may be that all transportation schedules are fixed.
- The amount of information relevant to the behaviour of the network may vary, and if a lack of information is considered, the way to model this situation may also vary [20].
- Whether some other stakeholders with authority in the network are in the model, and if so, how their behaviour is modelled.

A model is not necessarily improved by making it increasingly exhaustive. As it happens with most model-making, accuracy comes with a trade-off, in this case, computational power. This computational burden is an intrinsic property of operational synchromodal problems [23] and one that is of the utmost importance given the real-time nature of operational problems: new information is constantly fed and it should be processed in time.

There is no rule of thumb for making the decisions above, also, each of the decisions mentioned above will shape the model, and likely stir its solution methods to a specific direction. Though literature reviews of synchromodal transportation exist [25, 23], it appears no generalised mathematical model for synchromodal transportation problems has been found yet, nor a way of categorising the existing literature by their modelling approaches. The framework for mathematical synchromodal transportation problems on a tactical or operational level we present in this chapter aims to capture the essential model-making decisions done in the model built to represent the problem. When no such model is specified, it shows the model-making decisions likely to be done in that case (which makes classification partly subjective). This is done in an attempt to grasp the characteristics of the model/case in a compact way, enabling easy classification and comparison between models and cases, as well as a way to see the complexity of a specific case at a glance. Also, it provides a perspective to better relate new problems with previous ones, thus identifying used methodologies for the problem at hand.

3.2. **Framework identifiers and elements.** Within the framework we will refer to *demand* and *resources*. In synchromodal transportation models, demand will likely be containers that need to shipped from a certain origin to a destination. Resources can for example be trucks, train and barges. However, the framework allows for a broader interpretation of these terms. In repositioning problems, empty containers can be regarded as resources, whereas the demand items are bulks of cargo that need to be put in a container.

The framework has two main parts. The first part consists of the *identifiers*; these are specific questions one can answer about the model that depict the general structure of the model. The other is a list of *elements*; these elements are used to depict in more detail what the nature is of the different entities of the synchromodal transportation problem.

3.2.1. *Identifiers* . In this subsection we will elaborate on the identifiers of the framework. These identifiers are questions about the model. They identify the number of authorities, i.e. how many agents are in control of elements within the model. They will also identify the nature of different elements within the model. The list of elements will be discussed in detail in subsection 3.2.2, but they are used to determine which components in the model are under control, which are fixed, which are dynamic and which are stochastic. For instance, the departure time of a barge may be a control element, but it could also be fixed upfront, or modelled as stochastic. Some of the questions address how the information is shared between different agents and if the optimisation objective is aimed at global optimisation or local optimisation. All the answers on these questions together present an overview of the model, which can then be easily interpreted by others or compared to models from the literature.

The identifiers that discuss the behaviour of the model in more detail are discussed below.

(1) *Are there other authorities (i.e. agents that make decisions)?*
Here it is identified if there is one global controller that steers all agents in the network or that there are multiple agents that make decisions on their own.
  - *If there other authorities, how is their behaviour modelled:* one turn only, equilibrium *or* isolated*?*
    If the previous question is answered with yes, i.e. there are multiple agents that make decisions, one needs to specify how these authorities react to each other. We distinguish three different ways for modelling the behaviour of multiple authorities in a synchromodal network:
    – *One turn only*: this means that each agent gets a turn to make a decision. After the decision is made, the agent will not switch again. For instance we have three agents A, B and C. Agent A will first make a decision, then agent B and then agent C. The modelling ends here, since agent A will not differ from its first decision.
    – *Equilibrium*: the difference between "one turn only" and "equilibrium" is that after each agent has decided, agents can alter their decision with this new knowledge. In the same example: agents A, B and C make a decision, but agent A then decides that due to the decision of agent B to alter its decision. If nobody wants to alter their decision an equilibrium between the agents is reached.
    – *Isolated*: if the behaviour of the various authorities is isolated, it means that from the perspective of one of the authorities we only have limited information about the decisions of the other agents. For instance: agent C needs to make a decision. It is not known what agents A and B have chosen or will choose, but agent C knows historic data on the decisions of agents A and B. Agent

C can then use this information to make an educated guess on the behaviour of agents A and B.

(2) *Is information within the network global or local?*
This identifies if the information within the network is available globally or locally. If the information is locally available, it means that only the agents themselves know for example where they are or what their status is at a certain time. If the information is global, the network operator and/or all other agents know all this information as well.

(3) *Is the optimisation objective global or local?*
The same can hold for the optimisation objective. If all agents need to be individually optimised, the optimisation objective is local. If the optimisation objective is global, we want the best option for the entire network.

(4) *Which elements do you control?*
Since we want to model a decision problem, at least one element of the system must be in control. For example: if one wants to model which containers will be transported by a certain mode in a synchromodal network, we have control over the *demand-to-resource allocation*. If we want to model which trains will depart on which time at certain locations, we have control of the *resource departure time*. An extensive list of elements in mentioned in Subsection 3.2.2.
Of course the controllable element can have constraints: for instance we can influence the departure times of trains, but they cannot depart before a certain time in the morning. This is still a controllable element. We thus consider an element a controllable element if a certain part of it can be controlled.

(5) *What is the nature of the other elements* (fixed, dynamic, stochastic *or* irrelevant)*?*
The other elements within the network can also have different behaviours. We distinguish four:
- *Fixed*: a fixed element does not change within the scope of the problem.
- *Dynamic*: a dynamic element might change over time or due to a change in the state of the system (e.g. the amount of containers changes the travel time), but this change is known or computable beforehand.
- *Stochastic*: a stochastic element is not necessarily known beforehand. For instance it is not known when orders will arrive, but it is a Poisson process. It might also occur that the time the order is placed is known, but the amount of containers for a certain order follows a normal distribution.
- *Irrelevant*: the list we propose in Subsection 3.2.2 is quite extensive. It might occur that for certain problems not all elements are taken into consideration to model the system. Then these elements are irrelevant.

(6) *What is the optimisation objective?*
This identifier is for the optimisation objective. One can look at the exact same system but still want to minimise a different function. One could think of travel times and CO_2 emissions. It is also possible to provide a much more specific optimisation objective. Examples of optimisation objectives are in Subsection 3.4.

3.2.2. *Elements* . In this subsection we will give a list of elements we believe exist in most synchromodal transportation problems. They are divided in two parts: *resource elements* and *demand elements*. The resource elements are all elements related to the resources, which are mostly barges, trains and trucks. However, for compactness we also view a terminal as a resource. The demand elements are all elements related to the demand, which are most of the time freight or empty containers. Most elements mentioned in this list are straight-forward, we mention small clarifications if necessary.

- Resource elements:
  - *Resource Type* (*RT*): Different modalities can be modelled as different resource types. Another way to use this element is for owned and subcontracted resources.
  - *Resource Features* (*RF*): These features can be appointed to the different resource types or can have the same nature for the different types. For instance, it may be

that there are barges and trains in the problem, but their schedules are both fixed, thus making the nature of the resource features *fixed* for both resource types.

    * *Resource Origin* (RO)
    * *Resource Destination* (RD)
    * *Resource Capacity* (RC): Indication of how much demand the different resources can handle.
    * *Resource Departure Time* (RDT)
    * *Resource Travel Time* (RTT): Time it takes to travel from the origin to the destination (in the case of a moving resource).
    * *Resource Price* (RP): This can be per barge/train/truck/... or per container.
  - *Terminal Handling time* (TH): Time it takes to handle the different types of modes at the terminal. This can again be per barge/train/truck/... or per container.
- Demand elements:
  - *Demand Type* (DT): One can also think of different types of demand. For instance, larger and smaller containers or bulk.
  - *Demand-to-Resource allocation* ($D2R$): The assignment of the demand to the resources.
  - *Demand Features* (DF)
    * *Demand Origin* (DO)
    * *Demand Destination* (DD)
    * *Demand Volume* (DV): It might be that different customers have a different amount of containers that is being transported. (Note that the demand element in this case will always be 1 container, since each container can have its own assignment.)
    * *Demand Release Date* (DRD): The release date is the date at which the container is available for transportation.
    * *Demand Due Date* (DDD): Latest date that the container should be at its destination.
    * *Demand Penalty* (DP): Costs that are incurred when the due date is not met or when the container is transported before the release date (this is sometimes possible with coordination with the customers).

3.3. **Notation.** In this subsection, we will introduce some notation which will make it easier to quickly compare different models. Obviously, it is hard to keep a compact notation and still incorporate all aspects of a synchromodal system. Therefore, we made the notation as compact as possible and left out some of the details. When comparing models in detail it is easier to look at all answers to the identifiers mentioned in Subsection 3.2.1. Our notation has similarities to (among others), the framework of Kooiman for Time stamp Stochastic Assignment Problems [14], Kendall's notation for classification of queue types [13] and the notation of theoretic scheduling problems proposed by Graham, Lawler, Lenstra and Rinnooy Kan [10].

A synchromodal transportation model is described by the notation:

$$R|D|S \quad \text{or} \quad R|D|S|B,$$

depending on whether or not there are other authorities in the system. The letters denote the following things:

- R: resource elements,
- D: demand elements,
- S: system characteristics,
- B: behaviour of other authorities (if applicable).

Resource and demand elements The first two entries in the notation can be filled with all elements mentioned in the list in Subsection 3.2.2. As mentioned before an element can be one of five different things: controlled, fixed, dynamic, stochastic or irrelevant. Let us elaborate on the notation of these differences. For the sake of example, we will use the element *Demand-to-Resource allocation* ($D2R$). We propose to use the following notation for the different natures of this element:

- controlled element: $[D2R]$,
- fixed element: $\overline{D2R}$,
- dynamic element: $\widetilde{D2R}$,
- stochastic element: $\widehat{D2R}$,
- element irrelevant: $\not{D2R}$.

Writing down all elements will still result in a large string of text. Therefore, we suggest to use R and D for the most common aspect and thus noting only the elements that are different. For example, for the resource elements everything is fixed, except for the departure time, which can be controlled. We would recommend the reader to write down: $\overline{R}, [RDT]$.

If different resource types or demand types have a difference in some of the elements one can also write this down. We propose to write down in between brackets the different resource type. For example: for barges everything is fixed, except for the dynamic capacity, but for the train the capacity is stochastic and the other elements are fixed. This can be noted in the following way: $\{\overline{R}, \widetilde{RC}\}, \{\overline{R}, \widehat{RC}\}$. Note that this is the only way in which we incorporate the types into the notation. To know which types are used in the different models one has to look at the expanded notation. This choice is also made for the sake of compactness.

System characteristics For the system characteristics we have developed a notation in which you have an answer to questions $1, 2$ and 3 of the identifiers. Thus: are there other authorities, is the information global or local and is optimisation global or local.

The notation is based on Figure 3.1. In a similar way to this figure, the four options for the field *System characteristics* in the notation are:

- *selfish*: information global and optimisation local,
- *social*: information global and optimisation global,
- *cooperative*: information local and optimisation global,
- *limited*: information local and optimisation local.

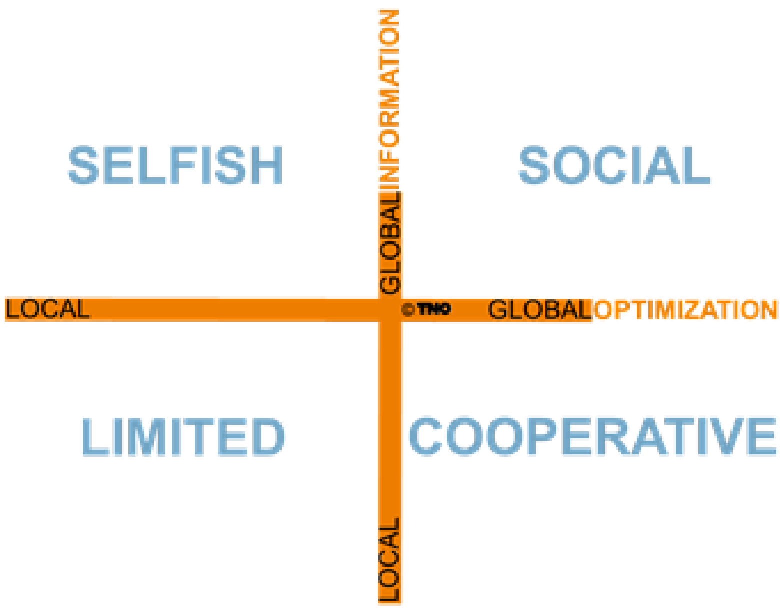

FIGURE 3.1. Different models of a synchromodal network.

In order to see if there are other authorities within the system we write down either an (1) or (1+) behind the option chosen. If it is known how many authorities there are it also possible to denote that number between brackets. One could for example write down: social(1) or cooperative(1+).

Behaviour of other authorities If there are other authorities within the system, their behaviour should be known (see question 1a in Subsection 3.2.1). The options are the same as discussed before:

- *one turn only*,
- *equilibrium*,
- *isolated.*

This field can be left blank if there are no other authorities in the system.

Remarks The notation we developed does not include the optimisation objective. This is done on purpose. We would like to show all the model characteristics within the notation. Within a specific model there is of course an option to look at different optimisation objectives. Since these might be quite elaborate, we did not want to shorten these objectives to a few words. We think this will only result in notation that needs more clarification. If one is interested, he/she can look at the entire list of identifiers.

This framework is developed in collaboration with multiple parties that study synchromodal systems. Therefore, we think we identified the resource and demand elements that are most common in synchromodal problems. However, for certain specific problems one might to extend the framework. We think this is easily done in the same way as we set up the framework. For example, one can add some elements within the list of elements or a different nature of one of the elements. However, one must keep in mind that the scope of the framework mainly covers mathematical problems on the operational and tactical levels.

3.4. **Examples .** As discussed earlier, one of the ideas of the framework is that, when starting work on a new problem, one can first classify the assumptions this model would need, then investigate papers that have similar classification. Therefore, we present a number of classification examples for both existing models as well as new problems. First, we answer the framework questions for the Kooiman pick-up case [14] in Table 3.2, and show how this can be written in our compressed notation. Afterwards, Table 3.3 shows compressed notation of some other problems described in papers, so that the interested reader can study more examples of our framework classification. Then, using Table 3.4, we examine some real-life cases and classify how we would choose to model these problems. These real-life problems do not yet have an explicitly described model, so this classification is based on how we would approach and model these practical problems, but other modellers may make other modelling decisions. Finally, the given examples will be used as input for discussion.

In the Kooiman pick-up case, a barge makes a round trip along terminals in a fixed schedule to pick up containers to bring back to the main terminal; however, the arrival times of the containers at the terminals are stochastic. At each terminal, a decision has to be made of how many containers to load onto the barge, and an estimate has to be made of how much capacity will be needed for later terminals, all while minimising the amount of late containers. The actual time of residing at the terminal is disregarded. We refer to table 3.2 for the answering of the framework questions. We refer to table 3.1 for a reminder of the framework element abbreviations.

RO: resource origin	DO: demand origin
RD: resource destination	DD: demand destination
RC: resource capacity	DV: demand volume
RDT: resource departure time	DRD: demand release date
RTT: resource travel time	DDD: demand due date
RP: resource price	DP: demand penalty
TH: terminal handling time	$D2R$: demand-to-resource assignment

TABLE 3.1. Abbreviations of the framework elements used in the compressed notation.

Other authorities	No
Information global/local	Global
Optimisation global/local	Global
Resource elements	RT: barges Controlled resource elements: none RF: fixed, except $\cancel{TH}$
Demand elements	DT: freight containers Controlled demand elements: $D2R$ DF: fixed, except $\widehat{DRD}$
Optimisation objective	Maximise percentage of containers that travel by barge instead of truck

TABLE 3.2. The framework questions applied to the Kooiman pick-up case.

Note that we have only taken barges into consideration as resources, not trucks. It would also have been possible to describe trucks as resources as well, but we have chosen to classify these as part of the lateness penalty, because there is no decision-making in how the trucks are used. Also, it may seem strange to speak of global or local information and optimisation when there are no other decision-making authorities. The information is considered global, because the only decision-making authority knows ‘everything’ that happens in the network; the optimisation is considered global, because the decision-maker wants to optimise the performance over all demand in the network put together, not over some individual piece or pieces of freight.

Using the framework notation, most of Table 3.2 can be summarised as follows:

$$\bar{R}, \cancel{TH}|\bar{D}, [D2R], \widehat{DRD}|social(1).$$

Only the optimisation objective and type specifications are lost in this process. In Table 3.3, we apply the framework to more problems from academic papers. In this table, we include the optimisation objective to illustrate the wide range of optimisation possibilities. It is not actually necessary to describe the optimisation objective when using the compressed problem notation. In some cases, especially practical problem descriptions, optimisation objectives may not yet be explicitly known. Therefore, Table 3.4 leaves them out. In that table we review some practical problem descriptions and apply the framework to these descriptions.

Behdani [2] $\bar{R}, [RDT]\|\bar{D}, [D2R]\|social(1)$ Objective: minimise transportation costs and waiting penalties
Kooiman [14] $\bar{R}, \cancel{TH}\|\bar{D}, [D2R], \widehat{DRD}\|social(1)$ Objective: maximise percentage of containers by barge instead of truck
Le Li [15] $\bar{R}, \cancel{RDT}\|\bar{D}, [D2R], \widetilde{DV}, \cancel{DRD}, \cancel{DDD}\|cooperative(1+)\|equilibrium$ Objective: with self-optimising subnetworks, minimise total cost in union
Lin [16] $\bar{R}, \widetilde{RC}, \cancel{RP}\|\bar{D}, [D2R]\|social(1)$ Objective: minimise total quality loss of perishable goods
Mes [17] $\bar{R}, \widetilde{RP}, \cancel{RC}\|\bar{D}\|social(1)$ Objective: best modality paths against different balances of objectives
van Riessen [24] $\{\widetilde{R}, \bar{RO}, \bar{RD}, [RDT]\}\{\widetilde{R}, \bar{RO}, \bar{RD}\}, \bar{TH}\|\bar{D}, [D2R], \widetilde{DP}\|social(1)$ Objective: minimise transport and transfer cost, penalty for late delivery and cost of use of owned transportation
Rivera [20] $\widetilde{R}\|\widehat{D}, [D2R]\|social(1)$ Objective: minimise expected transportation costs
Theys [26] $\bar{R}, [RP], \cancel{RDT}\|\bar{D}, [D2R], [DP], \cancel{DRD}, \cancel{DDD}\|selfish(1+)\|equilibrium$ Objective: fairest allocation of individual costs
Zhang [27] $\bar{R}\|\widetilde{D}, [D2R]\|social(1)$ Objective: maximise balance of governmental goals

TABLE 3.3. For selected papers, a classification of where their problem falls in the synchromodal framework.

Lean and Green Synchromodal [31] $\bar{R}\|\bar{D}, [D2R]\|selfish(1)$
Rotterdam – Moerdijk – Tilburg [35] $\bar{R}, \widehat{RTT}, \widehat{TH}\|\bar{D}, [D2R]\|social(1)$
Synchromodaily [32] $\bar{R}, [RDT]\|\widehat{D}, [D2R]\|social(1)$
Synchromodal Control Tower [33] $\bar{R}, [RC], \widehat{RP}, \widehat{RTT}, \widehat{TH}\|\bar{D}, [D2R], [DV]\|social(1)$
Synchromodal Cool Port control [34] $\bar{R}, [RDT], \widehat{RTT}\|\bar{D}, [D2R], \widetilde{DDD}, \widetilde{DP}\|social(1)$

TABLE 3.4. For selected use cases, a classification of where a possible model for this problem would fall in the synchromodal framework.

Another example we reviewed is the modelling of an agent-centric synchromodal network. Here all agents want to be at their destination as fast as possible, but everyone does share the information about where they are and where they are going with everybody else in the network. Table 3.5 shows the answer to the questions of the framework. In the short notation this problem is:

$$\overline{R} \mid \widehat{D}, [D2R], \cancel{DP} \mid \text{selfish}(1+) \mid \text{equilibrium}$$

Other authorities	Yes
Information global/local	Global
Optimisation global/local	Local
Resource elements	RT: barges, trains and trucks Controlled resource elements: none RF: fixed
Demand elements	DT: containers Controlled demand elements: $D2R$ DF: stochastic, except $\cancel{DP}$
Optimisation objective	Minimise travel times

TABLE 3.5. The framework questions answered for the agent-centric synchromodal network.

3.5. **Discussion.** These examples show some strengths and limitations of the classification framework, which are discussed in this subssection.

One of the goals of this framework was to offer guidance when tackling a new problem: as an example, if the problem from the Synchromodaily [32] case is modelled in a non-stochastic way, we can now see that it may be worthwhile to study the solution method presented by Behdani [2], because they then have the same compressed framework classification. If such a record is kept of papers and models, this could greatly improve the efficiency of developments in synchromodal transport. This would fulfil the second goal of the framework: to collect literature on synchromodal transportation within a meaningful order.

The final goal of this framework was to expose and compare relationships between seemingly different problems: for example, we can now see that the problems described by Le Li [15] and Theys [26] have similarities, in that they investigate negotiation between parties and do not focus on timeliness of deliveries. Similarly, we can see that the model assumption Mes [17] makes in disregarding resource capacity, is an unusual decision.

In the Synchromodaily case [32], our interpretation of the problem implies that the demand features are stochastic. However, the problem could also be approached in a deterministic way, depending on choices that the modeller and contractor make based on the scope of the problem, the requirements on the solution and the available information. This shows the most important limitation of the classification framework: what classification to assign to a problem or model remains dependent on modelling choices, as well as interpretation of problem descriptions. Even without framework, however, modelling choices will always introduce subjective elements into how a real-world problem is solved. This framework can be used to consistently communicate these underlying model assumptions.

A second limitation of the framework is that, because of the large amount of elements described in it, two similar problems are relatively unlikely to fall in the exact same space in the framework because of their minor differences. Therefore, one should not only look for problems with the exact same classification, but also problems with a classification that is only slightly different. In a more general sense, solution methods may apply to far more than one of these very specific framework classes. If two problems have the exact same controlled elements, it is imaginable that their models and solution methodologies may largely apply to the other. As a point of future research, it could be interesting to investigate which classification similarities are likely to imply solution similarities, which may also be a stepping stone towards a general solution methodology.

As a final limitation, the compressed notation does not reveal that the paper by Lin [16] and the 'Synchromodal Cool Port control' [34] case both focus on perishable goods. This shared focus is not only cosmetic: mathematically, it may imply objective functions and constraints not focused on in other cases. To mitigate this limitation, we advise anyone using the framework to offer both a compressed and an extended description of their problem or model.

4. Minimum cost multi-commodity flow

The minimum cost multi-commodity flow problem is a generalization of the minimum cost flow problem to the case where there are multiple flows with multiple origin-destination (OD) pairs. It is widely used in modeling all sorts of networks: communication, computer and distribution networks to name a few. As was mentioned before, flow of goods on a tactical-operational level in a synchromodal, intermodal or multimodal network can be modeled via a multi-commodity flow problem on a special kind of graph called space-time network. Modeling in these kind of graphs comes inherently with some challenges and advantages. Also, in most cases, a basic multi-commodity flow problem formulation is too simple to accurately model a transportation network, and features needed to make the model more realistic are often mentioned in the literature and practice. On top of that, computational efficiency is of the utmost importance for a synchromodal transport chain. The goal of this section and the following is to encourage the research on this subject by showing the potential progress in a solvable instance, while making a contribution in this topic, and to analyze the obstacles and assumptions done in this model: Comparisons are done between different modeling approaches of the minimum cost multi-commodity flow problem applied to this specific case. Additionally, our model incorporates the possibility of delay in the due date of orders, and two different kinds of transport are considered, namely, subcontracted and owned. These features are mentioned both in the literature and in practice to be relevant for the planning process. The characteristics of the problem as well as the modeling decisions make these models quite different from a basic minimum cost multi-commodity flow problem, which in turn make effectiveness of solution methods differ. Finally, future areas of research are pointed out.

As stated previously, in our case it is crucial to reduce the computational time as much as possible, not only because a speed up is always an upgrade, but because the review on the literature corresponding to our problem gave some striking insights about the needs and characteristics of our problem:

(1) Huge computational demand is an intrinsic characteristic of synchromodality, and fast computation is essential for the real-time nature of synchromodal transportation. For instance, simulation methods are often performed to overcome uncertain or stochastic environments. This implies to calculate a lot of instances in a short time. Breakthroughs on synchromodality use clever solution methods to overcome this computational load.
(2) Exhaustive models exist, however the computational demand increases rapidly as the problem instance increase. This is also the case of MCMCF's on space-time networks.

4.1. Multi-commodity flow.

On a graph (G, A) with n nodes and m arcs, where each arc (i, j) has capacity $u_{ij} > 0$, the multi-commodity flow problem is a network flow problem with k commodities of d_k demand of flow between different source nodes s_k and sink nodes t_k. This problem is concerned with finding a feasible flow. The minimum cost multi-commodity flow problem (MCMCF) is the problem of finding a minimum cost feasible flow. Following a representation similar to the minimum cost flow problem, MCMCF problem can represented via the following linear program (LP).

$$
\begin{aligned}
&\min \sum_{(i,j)\in A} \sum_{k} c_{ij} x^k_{ij} \\
&\sum_{j} x^k_{ij} - \sum_{l} x^k_{li} = \delta^k_i \quad \text{for all } i \in N(G), k \\
&\sum_{k} x^k_{ij} < u_{ij} \quad \text{for all } (i,j) \in A(G) \\
&x^k_{ij} > 0 \quad \text{for all } (i,j) \in A(G), k
\end{aligned}
\tag{4.1}
$$

Where

$$\delta_i^k = \begin{cases} d_k & \text{if } i = s_k, \\ -d_k & \text{if } i = t_k, \\ 0 & \text{otherwise.} \end{cases}$$

Each decision variable x_{ij}^k represents the flow of a specific commodity on a given arc. We will refer to the previous LP program as the *arc-based formulation* of MCMCF.

Another formulation for the multi-commodity flow problem is the following. Let $P(k)$ be the set of all directed simple paths on G from s_k to t_k, $C(P)$ the cost of the path $P \in \cup_k P(k)$, that is, the sum of all the costs of arcs $(i,j) \in P$. Then the MCMCF problem can be formulated as

$$\begin{aligned} &\min \sum_k \sum_{P \in P(k)} C(P)X(P) \\ &\sum_k \sum_{P \in P(k)} X(P)\delta_{ij}(P) \leq u_{ij} \quad \text{for all } (i,j) \in A \\ &\sum_{P \in P(k)} X(P) = d_k \quad \text{for all } k \\ &X(P) \geq 0 \quad \text{for } P \in \cup_k P(k) \end{aligned} \tag{4.2}$$

Where

$$\delta_{ij}(P) = \begin{cases} 1 & \text{if } (i,j) \in P \\ 0 & \text{if } (i,j) \notin P \end{cases}$$

In the previous formulation, there is one decision variable $X(P)$ for each path between an OD pair, for each OD pair. Therefore the previous formulation will be referred to as the *path-based formulation* of MCMCF.

Since MCMCF problem can be formulated via a LP, it can be solved in polynomial time (that is, the time to find a solution is bounded by a polynomial in the size of the input) which is always a desirable feature to have. Given a specific instance, both the arc-based and the path-based formulation would of course yield the same optimal solution. However, the LP formulations are quite different: the arc-based has $n * k + m$ constraints (not counting the positivity constraint of the variables) and $m * k$ variables, whereas the path-based formulation has $m + k$ constraints and can have as many as $O(n^{n-2}) * k$ variables[1]. Therefore these ways of expressing the problem usually differ in computational time.

Given the potentially astronomical number of variables in the path-based formulation, it might result unsurprising that the arc-based formulation is more popular. However, in some cases, path-based formulation is more convenient, and in many applications they rely on column generation to deal with the number of variables [29].

4.2. **Integral multi-commodity flow.** In an intermodal transportation network, flow of goods is done via containers. In mathematical terms, this translates that only integer feasible flows are of interest. This brings the problem of freight transport into the realm of Mixed Integer Linear Programming (MILP), which unfortunately is NP-hard in general [5]

A first reasonable question in any MILP is whether the integral condition on the decision variables are needed to obtain an integral optimal solution, that is, whenever the polyhedron defined by the program is integral. If the answer is no (lucky), then it means that the MILP problem can actually be solved via a faster LP. Unfortunately, this is not the case for MCMCF. In the graph of Figure 4.1 we see that the optimal flow is the one described.

In fact, even the problem of obtaining an integral feasible flow can be quite hard, as stated in [6] "integer multi-commodity flows are as difficult as the TSP even with two commodities and

[1] This would be in the worst case, which corresponds to a complete graph K_n of degree n, on which careful thought will reveal that the number of paths between two nodes on K_n is a_n with the sequence defined as $a_n = 1 + (n-2)(a_{n-1})$ and $a_1 = 0$)

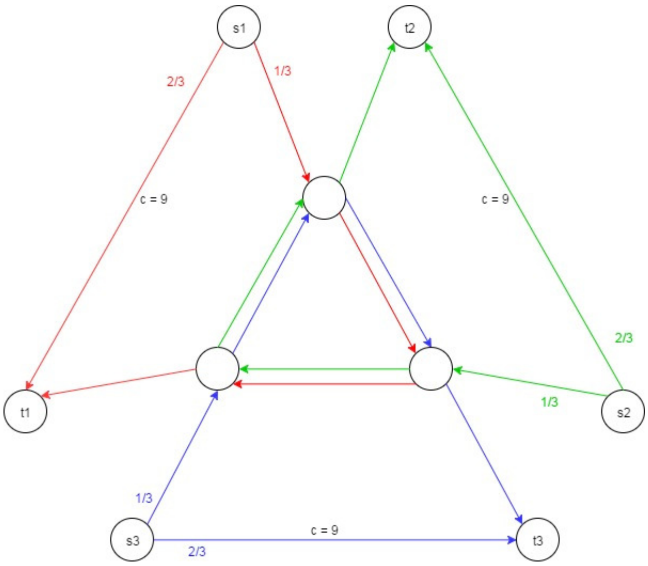

FIGURE 4.1. We assume the capacity of each arc to be $u = 1$ and the cost c also equal to 1 on all except the arcs with cost 9 as it is displayed. With demand for each commodity equal to 1, we can see that the optimal flow is the one given by the color arcs, with flow 1/3 on the cycle and 2/3 outside of it (The color arcs represent the flow, not the actual arcs on the graph, so repeated arcs represent flow of two different commodities).

each capacity set to 1". That is, the integral multi-commodity flow problem is NP-complete, which means the integral MCMCF is NP-hard.

4.3. **Integral multi-commodity flow on space-time network.** Although there might be complications given by the non-integrality of multi-commodity flows in general graphs, we would like to check if there are some advantages or disadvantages in our case. That is, whether there are any benefits or drawbacks of dealing with multi-commodity flows on a space-time network. After all, space-time graphs have a very characteristic topology. In this subsection we discuss the integral MCMCF on space-time graphs.

The idea behind a space-time graph, as its name suggests, is that every node represents a location at a specific time, and arcs represent a change of state. They are meant to show the characteristics of an underlying graph G' with node set S as time changes discretely from 1 to T where each of these discrete times is referred to as a time-stamp. Although it has been referred to previously on the literature review, we start this section with the following definition.

Definition 4.1. We say that a graph G is a space-time network (or space-time graph) if its node set is of the form $S \times \{1, 2, ..., T\}$ for some $T \in \mathbb{Z}^+$ and some set S and every arc $((a, p), (b, q)) \in A(G)$ satisfies $p < q$. We refer to the node (a, p) as location a at time p, and to T as the time horizon of G. We say that G' is a static network of a space-time network G if the node set of G' is S and if $(a, b) \in A(G') \iff \exists p, q \in \{1, 2, ..., T\}$ such that $((a, p), (b, q)) \in A(G)$

From the definition it is clear that there is a unique static network for each space-time network, however, many different space-time networks can give rise to the same static network. Therefore, although a static model is easier to interpret in many contexts (for instance, nodes being hubs and arcs being transportation links) a static model cannot represent the logistic transport chain with the required exhaustiveness; space-time networks enable, for instance, to model a transportation mode with OD pair $(a, b) \in S \times S$ that departs and arrives at a specific time-stamp.

Although it might be intuitively clear, a mathematical proof that flows on a space-time network correspond to flows on a given network throughout time is done in [9]. There, it is done for a special case of space-time graph called time-expanded graph, but the arguments are exactly the same for the general case. Therefore, we are interested in finding a flow in a space-time network.

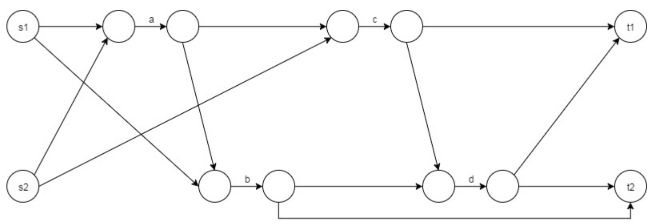

FIGURE 4.2. Assuming every edge has capacity 1, and that the demand for each OD pair (s_i, t_i) is 1, this graph has no feasible integral flow. This is because every 1-path from s_1 to t_1 will full the capacity of either edges a and c, or edges b and d

Proposition 4.2. *Space-time networks are acyclic, that is, there is no directed cycle in them. Conversely, every acyclic graph is isomorphic to a subgraph of a time-space graph.*

Proof. The fact that every space-time network is acyclic is trivial since every arc's endpoint has a higher value in its second entry than the arc's origin. The converse follows from the fact that the node set of every acyclic graph can be partitioned into sets that can be regarded as "generations", in mathematical terms, let

$$G_1 = \{i \in N(G) | \forall j \in N(G) \nexists (j,i) \in A(G)\} \neq \emptyset$$

Then we define iteratively

$$G_i = \{i \in N(G) \setminus \cup_{k=1}^{i-1} G_k \ | \forall j \in N(G) \setminus \cup_{k=1}^{i-1} G_k \nexists (j,i) \in A(G)\}$$

By definition the sets G_i are a partition and every arc on G goes from a node in G_i to a node in G_j for some $j > i$. We make the needed space-time graph SG by defining the time horizon as $T = \min\{i | G_i = \emptyset\}$, $N(SG) = N(G) \times \{1, 2, ..., T\}$ and $A(SG) = \{((a,t),(b,s)) | a \in G_t, b \in G_s, (a,b) \in A(G)\}$. Then by considering the function $\Pi : N(G) \to N(SG)$ with $\Pi(a) = (a,i)$ if $a \in G_i$ we see that $(i,j) \in A(G) \iff (\Pi(i), \Pi(j)) \in A(SG))$ that is, G is isomorphic to a subgraph of SG ■

As was shown in the previous section, the polyhedron defined by the MCMCF problem is not integral. Restricting ourselves to a space-time graph is no exception: The graph in Figure 4.2 is an acyclic graph with no feasible integer flow, and therefore, by the previous proposition, it is isomorphic to a subgraph of a space-time network. Then the only feasible flow (and therefore, the optimal one) will be fractional. Notice that even if there were another feasible flow that is integral on the space-time network that has the graph in Figure 4.2 as a subgraph, the costs of the arcs in the space-time network can be manipulated so that the optimal flow is still fractional.

Thus, we have just seen that the integral MCMCF problem may not be solved via LP in general, since we have found a space-time graph with optimal fractional flow. However, the idea of investigating the complexity of this case came to interest after conducting a series of computational experiments: more than 200 simulations of instances with varying size[2] were carried over, and using the graph's attributes as input on a LP solver, each one of them returned an optimal integral flow. This is quite surprising and it suggests that despite the fact that the polyhedron of problem at hand is not integral, it is worthwhile to first check whenever the non-integral LP formulation returns an integral solution (many MILP solvers usually work by first computing a LP solution and then checking if it is integral or not). Thus, we propose to investigate further this unusual behavior as an interesting route for future research.

[2]Our simulations considered instances ranging from 4 to 30 locations, 20 to 300 time-stamps, 5 to 20 commodities and with varying density. All parameters as presented in the next section

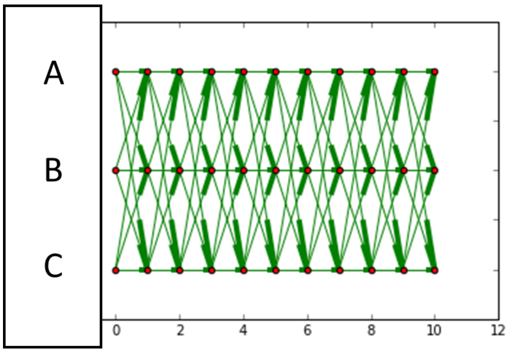

FIGURE 5.1. Space-time network with three locations and ten time-stamps with all truck-arcs. The thick end represents the end of the arc

5. MCMCF: Assumptions and implementations

We seek making a model that has a good balance between having an applicable real instance and a solvable instance. Therefore the assumptions done in the model are of the utmost importance: assume too much, and the problem is irrelevant. Assume too little, and the problem is too hard. In this section we discuss what assumptions are sensible to do, our modeling approach to the space-time network, and the implementations done for MCMCF on the graph built. In the previous chapter we presented two different approaches to model a MCMCF problem, namely, arc-based and path-based. Here both approaches were considered and built upon to compare their performance on a space-time network. Also, we discuss some features that were added to the model since they are considered relevant in the literature study.

Therefore, the two main goals we aimed for while building the models are:

- Solve the integral MCMCF on a space-time network in an efficient way with an instance that has a good balance between being realistic and solvable
- Add features that are considered relevant both in the literature as well as in practical cases of synchromodal transport planning.

These are the main drivers behind the assumptions done in the models here presented.

5.1. **Assumptions.** We used the library *networkx* for python in order to build the required space-time network. In the literature of intermodal transportation planning on a tactical-operational level, one modeling assumption is encountered frequently:

> Assumption 1: At every time-stamp, there are an unlimited number of trucks going from any location to any other location. These trucks are more expensive and quicker than any other means of transportation.

We additionally consider the following assumption

> Assumption 2: Every number of containers has the possibility of remaining idle in a given location with no additional cost.

This is not a very unrealistic assumption, as it is mentioned in [15] pp. 39 "Based on the NEA report, and after consulting experts in the freight transport... the storage cost at terminals for a relatively short period is very small or even zero".

To illustrate the effect of these two fundamental assumptions on our space-time graph, in the Figure 5.1 we see a graph with three different locations, in our terms, $S = \{A, B, C\}$ and $T = 10$ with only truck arcs and idle arcs.

The previous assumptions guarantee there is always a feasible flow. In a sense, solving the MCMCF problem in a space-time graph with these assumptions is minimizing the use of trucks in a cost-effective way. A naive MCMCF implementation on a space-time network with no further assumptions will result inefficient because a lot of the truck arcs will not be used, and can be unnecessary computational burden. For this reason, the following assumption is useful in implementation because it allows for a single truck arc for each order.

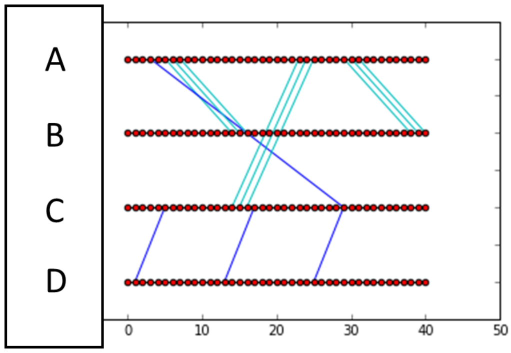

FIGURE 5.2. Space-time network with four different kinds of arcs: owned periodic arc (from A to B), owned simple arc (from C to A), subcontracted periodic arc (from D to C) and subcontracted simple arc (from A to C)

Assumption 3: Truck price is fixed and is the same for every OD pair

In essence, the above assumption may be weakened by saying that truck price must obey a triangle inequality in the following sense: It is always cheaper to take a truck directly than to take a mode of transport and then a truck. In mathematical terms, if c_{t_1} and c_{t_2} are the costs of truck from A to B and from C to B, respectively, then $c_{t_1} \leq c_{t_2} + c$ where c is the cost of any mode from A to C. This weakened version might be useful in some applications, but for simplicity, we assume that the truck price is fixed. The impact on the combination of Assumption 1 and Assumption 3 will be gauged in the beginning of the next subsection

In mathematical terms, the following assumption guarantees that we are dealing with a simple digraph instead of a multi-graph.

Assumption 4: Only one arc from (A,t) to (B,s) is allowed

This is not a very restrictive assumption, since on a transportation chain it is unlikely to have two options of transport that share the exact same OD pair, departure time and arrival time. It is not uncommon to make this assumption [20].

The space-time graph generated by our program constructs a given amount of randomized extra arcs, as well as establishes the characteristics of a number of orders as requested by the user (which will be further explained in the next subsection). Two distinctive features have been added on the space-time graph, given that they were considered relevant in [23]. These features are:

(1) Distinction between owned and subcontracted transport is considered.
(2) Delay in orders is allowed

Feature 1 implies a modeling difference that converts our "most likely" LP program into MILP (remember that, as was discussed in the section "integral multi-commodity flow on space-time graph", solving via LP usually suffices even if we restrict to only integer solutions) because pricing schemes for owned transport differ from those of subcontracted transport. Also, owned transport has some flexibility in terms of departure times. Delay was implemented via inverse arcs (with the needed constraints). An example of a resulting space-time graph is in Figure 5.2, where the cyan arcs are repeated consecutively a predefined amount of times (in this case, 3) to represent possible departure times. Notice that in this new version of the graph, the truck arcs are not drawn anymore since, as can be guessed from Figure 5.1, the draw would be messy and not representative. The precise way in which these features were added is explained in the following two subsections.

5.2. **Implementations.**

5.2.1. *Problem instances.* In this subsection we introduce the problem instances that are used to make comparisons between the methods.

We use the library *pulp* from python to input and solve our (MI)LP. In order to compare the different implementations, we consider three different space-time graphs with different sizes:

Problem 1: G has a time horizon $T = 30$, number of locations is 6, 40 trips and 10 orders.
Problem 2: G has a time horizon $T = 80$, number of locations is 10, 100 trips and 20 orders.
Problem 3: G has a time horizon $T = 200$, number of locations is 20, 250 trips and 40 orders.
Problem 4: G has a time horizon $T = 200$, number of locations is 20, 400 trips and 200 orders.

We will document the difference on performance by taking into account the following numbers:

- Computational times are measured in seconds, which if relevant is presented in:
  - Computational time on building and pre-processing (if available) the space-time graph.
  - Computational time on building the (MI)LP program
  - Computational time on solving the (MI)LP program
- Problem size: number of variable inputs, and constraints

The estimates of data provided in [15] propose the costs per container per kilometer as 0.2758 for a truck, 0.0635 for a train and 0.0213 for a barge. Therefore a barge is around 13 times cheaper than a truck, whereas a train is 4.5 times cheaper. We model the cost of a truck as 40, and the cost of a barge as $3 + \text{unif}\{-1, 0, 1\}$ where unif takes the values $-1, 0$ or 1 with equal probability. The cost of a train is $9+$ unif$\{-2, -2, 0, 1, 2\}$, however, in the following comparisons only barges are considered, and arcs representing trips were evenly randomized among the possible OD pairs. Thus all the analysis done in the following section will assume only barges as resources with such OD distribution. This will not limit the results, since it will barely have impact on the efficiency of the computation of solutions.

Experiments are repeated ten times and average values as well as standard deviation are documented. The capacity of each arc is a random integer between 5 and 19, whereas the size of the order is a random integer between 1 and 29. The influence of delay is also pointed out. After this a conclusion on the benefits and drawbacks of the approaches is done, and some routes for future improvement are pointed out. The average value per container is mentioned in each method, however, this should not be relevant as all methods will yield the optimal solution for each instance. The main difference in each method lies in the average computational time (ACT).

The computational results are summarized in tables at the end of this section.

5.2.2. *MCMCF methods and heuristics.* In this subsection we introduce the methods and heuristics that will be used.

Method 1.1. A naive arc-based MCMCF implementation just as formulated in (4.1) with the exception that we don not add the constraints of capacity for the truck arcs since we assume they're infinitely capacitated.

Method 1.2. An arc-based implementation were some redundant arcs were omitted: instead of drawing truck arcs from every location to every other location at each time-stamp, we use Assumption 1 and Assumption 3, allowing us to consider only one arc for each order; that is, for each order with origin (x, t_1) and destination (y, t_2), we draw the truck arc $((x, t_2 - 1), (y, t_2))$. Also, two important features that were added on this model operates are:

(1) The possibility to incorporate possible delay on the arrival date of an order
    This is done via inverse arcs towards each destination node (d, t) for each OD pair, that is, a specific allowed delay of time-stamps s is determined and arcs $((d, t+h-1), (d, t+h))$ are added to the space-time graph with a cost of 1, and constraints are added to ensure this arcs are used only for delayed deliveries.
(2) Two different kinds of transports: owned and subcontracted. The difference between these kinds of transports are two:
    - The transportation cost done on a subcontracted transport is done per container, whereas the cost on a owned transport is per trip
    - The departure date on a owned transport can be delayed to better fit our planning

    The Method 1.1 based on a direct implementation of arc-based MCMCF problem allowed only for costs to be done as subcontracted transport. This method provides the opportunity to consider costs in this new way. However, this comes with a trade-off:

owned transport must be modeled via binary variables, to determine departure times, and whenever certain possible trips are happening or not. In Figure 5.2 there are three consecutive trips from C to A. In the modeling of such a graph, there is a binary variable representing each of these trips, with the constraint that at most one of them is equal to one (so that three possibilities of departure times are available). The cost of these arcs is 0, which implies there is no cost for each container that flows on this arc. However, the objective function has a new term added, namely, $\sum_{o \in \text{owned}} 40x_o$. This means that for every trip of an owned transport, there is a cost of 40. Also, the possibility of having periodic trips for both owned and subcontracted transport is done in this model to make the schedules more realistic. Arc capacity for owned transport can be expected to be higher, thus the capacity for this transport is settled as a random integer between 12 and 19

Other minor structural technical changes are done, such as reformulating the way to input the flow conservation constraints so that $n * (k + 2m)$ operations are performed instead of $2n^2 * k$.
Method 2.1. A path-based implementation based on the path-based formulation shown in equation (4.2) with the two features of Method 1.2. In order to deal with the potentially huge amount of paths, a pre-processing stage is done in which the relevant paths for each OD pair are specified: For each OD pair, we use Dijkstra's algorithm to obtain all paths from the origin to the destination on the space-time graph. For each path on the space-time network, consider the corresponding path on the static network, that is, consider a path with the locations only as nodes, disregarding time. Because of the Assumption 2, we know that if the corresponding path on the static graph is not a simple path (where a simple path is a path whose vertices are not repeated) then the path on the space-time network is not cost-effective. Therefore we can discard all such paths. This turns out to be a very aggressive reduction, resulting in few paths to consider. However, the process of obtaining all paths for each OD pair can be quite computationally expensive, which makes this method more computationally heavy on the pre-processing than the previous ones. Nevertheless, it might be the case that there is not a lot of different paths for a given OD pair.
Method 2.2. This method is a further improvement of Method 2.1, where in order to maximize computational power, certain functions such as printing the space-time network (which is not useful for large cases anyway) are ignored. Also, a rearrangement of the code is done to avoid bottlenecks.
Heuristic 1. In the same practical spirit as found in some of the literature, a simple and quick heuristic was developed which is a successive shortest path algorithm:

Find the shortest path (in terms of cost) for the order (start with the first), and increase the flow variable of that path (decreasing the capacity of the arcs of the path) until either the demand of that order is reached or the capacity of any of the arcs on that path is full. If the former, reinitialize the algorithm for the next order, if the latter, reinitialize the algorithm for the same order.

In principle, this may not be a good approximation for a general MCMCF problem, and the amount of error has no upper bound. However, this depends on the paths available for each order, and on the characteristics of the space-time graph: for example, if the cheapest paths of each order is disjoint with the cheapest paths of every other order, then the solution of using the shortest path algorithm and the MCMCF problem are the same. Also, if the capacities are too large compared to the orders size, so that fully capacitated arcs are not possible, then shortest path and MCMCF yield the same result. However, it should be noted that in practice, fully capacitated modes are frequent, and cheapest paths are often entwined.

This heuristic was adapted to the case with owned transport via an auxiliary graph. This auxiliary graph modifies the cost of the owned arcs from 0 to 40c/cap where 40 is the cost of using the owned transport (coefficient on the binary variable in the objective function), cap is the capacity of the barge and c is a constant chosen to be 1 on this example, but that can (and should) be modified with respect to the expected use of an order in that owned trip (c meaning around $1/c$ of the trip is expected to be used). Also, proper care is done to ensure that at most

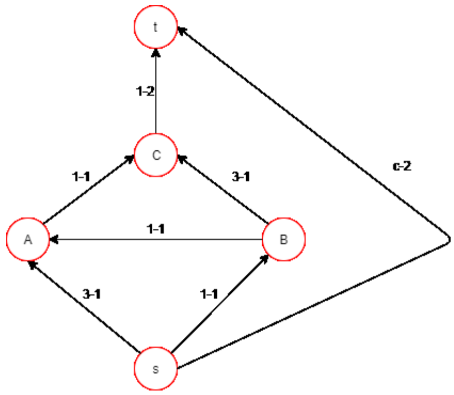

FIGURE 5.3. The numbers in the arcs represent cost-capacity. 2 units of flow must be sent from s to t.

one of the arcs representing the same trip at different times is used (for a more detailed of how this is done, see from Definition 6.5 to Definition 6.7).

Heuristic 2. Heuristics 2 is a slightly different approach to Heuristics 1. The algorithm goes as follows:

Find the minimum cost flow for the order (start with the first) with source the origin node, sink the destination node, and flow equal to the demand of the order. Decrease the capacity of the graph accordingly to the flow and reiterate for the next order.

Notice that for a problem of a single commodity, the solution of Heuristic 2 is precisely the MCMCF, whereas the Heuristic 1 can be arbitrarily worse (in terms of the cost function) than Heuristic 2: In Figure 5.3 using Heuristic 1, the cost of the transportation will be $4 + c$ whereas with Heuristic 2 the cost is 10. Since c can be arbitrarily large, Heuristic 1 can be arbitrarily large. Also, notice that the graph is acyclic and thus can be embedded in a space-time graph by Proposition 4.2.

The fact that for a simple commodity Heuristic 2 performs better than Heuristic 1 does not imply that for the case of multiple commodities. In fact, often Heuristic 1 gives a more cost-effective solution than Heuristic 2 for instances with many orders (more than 100).

At first, this might seem counter-intuitive, But it should be noted that the problem solved by each heuristic is different, since each iteration (order) in the heuristic determines the following sub-problem. If the number of orders is large, then there is larger chance that the sub-problems solved at further stages are different, which restricts the possibility of determining whenever one algorithm will perform better (in terms of cost) than the other. If the number of orders are small however, Heuristic 2 usually gives better results than Heuristic 1.

5.2.3. *Computational results.*

Method 1.1.

Problem 1. To assess the need of improvement in modeling approaches, we analyze the performance of Method 1.1 on Problem 1. A typical example of the kind of space-time graph for this problem is in Figure 5.4

As observed in the section of multi-commodity flow, the number of variables given in this problem is $m * k$, that is

$$\begin{aligned}(\ \#\text{ of truck arcs at each node } * \ \#\text{ of nodes at each timestamp}*\\ \#\text{ of timestamps } + \ \#\text{ of barge arcs }) * \ (\#\text{ of orders})\\ = (6 * 6 * 30 + 40) * 10 = 11200\end{aligned}$$

and the number of constraints is $n * k + m' = (31 * 6) * 10 + 40 = 1900$ where m' is the number of barge arcs. Thus every implementation has 11200 variables and 1900 constraints (except if an arc is overwritten, in which case the number of variables is reduced by 10 and the number of constraints is reduced by 1. However, given the number of arcs, this happens with an approximate frequency of 1 out of 10 times).The average value per container is 32.1375.

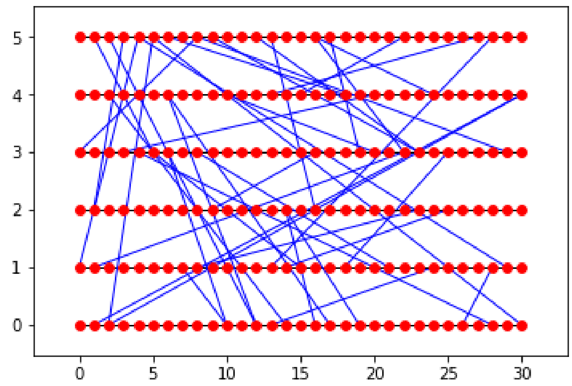

FIGURE 5.4. STN of Problem 1

The average computational time (ACT) is 60.6225 seconds with a standard deviation of 6.0782 seconds. It should be noted, however that the average time to solve is 6.5555 seconds, and most of the computational time is done on the input of constraints on the problem (on average, 50.8746), particularly in the flow conservation constraints, since they involve three nested "for" loops, with a total amount of $2n^2 * k = 345,960$ operations. We can therefore deduce that this implementation will scale badly as instances increase, since in Problem 2, we would have $2 * 1010^2 * 20 = 40,804,000$ operations. That is, the program must take at least 100 times more to solve than it did to solve Problem 1. Indeed, the program took longer than 1 hour for a single instance.

Method 1.2.

Problem 1. For Problem 1 considering only subcontracted arcs with no delays (as it was solved with Method 1.1), the reduced amount of arcs is $m = m' + m'' + k = 40 + (6 * 30) + 10 = 230$, where m'' is the number of arcs of the form $((x, t), (x, t + 1))$. This number is 5 times smaller than the number of arcs we had in the previous model. We get then a total of 2300 variables and 1900 constraints. The ACT is 2.4090 seconds with a standard deviation of 0.16 where the computational time is mostly spent on solving the problem, with an average of 1.5730 to solve. The average value per container is 31.2731 which, as expected, should not change so much from the previous method.

Problem 2. First we examine the case with all subcontracted transport and no delay. Method 1.2 can perform satisfactory with instances of the size of Problem 2. We simulated and solved 10 instances of this problem and obtained the following: The implementations have 18400 variables and 16300 constraints, with an ACT of 16.0368 and standard deviation of .9656. The average value per container is 32.0713

If we allow delay of at most 5 time-stamps, we obtain that the ACT is 17.8946 with a standard deviation of 1.1444 and an average value per container of 29.6361 which is a bit lower than the non-delay version as expected. Unlike the previous cases, the amount of variables and constraints now vary per instance, since the amount of inverse arcs depend on the OD pairs. In fact, you can add as many as $k * 5 * k = 2000$ variables (thus giving a total of 20400 variables) but this depends on the overlap and lateness of the inverse arcs. The average amount of variables is 20058 with a standard deviation of 103.3247. Since variables vary, so does the constraints, with an average value of 18018.4 and a standard deviation of 91.2548. We can conclude that delay makes the problem a bit more computationally heavy, but not significantly (compare with the values obtained above).

If 100 owned barges are used instead of subcontracted barges, with three consecutive possible departure times per barge (see Figure 5.5), we obtain an ACT of 21.6002 with a standard deviation of .57992 and an average value per container of 31.3409.[3] The average amount of

[3]The reason to have a slightly lower average container cost lies in the choice of parameters: the average capacity of a owned barge is 15, and if an owned barge has more than 13 containers, then it is cheaper per container

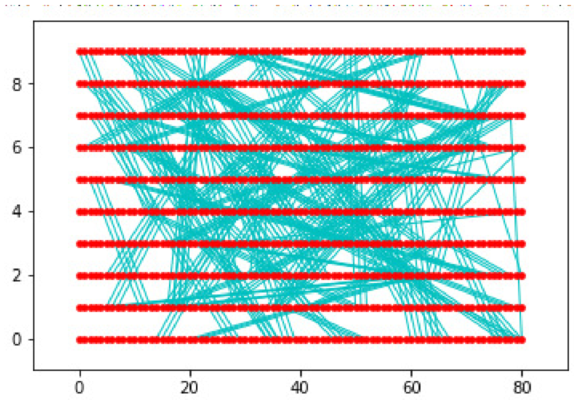

FIGURE 5.5. STN of problem 2 with owned transport

variables is 22552 with a standard deviation of 73.3212 The average amount of constraints is 16885 with a standard deviation of 7.3239. Therefore, for this problem, the use of owned transport (and therefore the use of ILP instead of LP) increases the computational burden by a factor of around 1.35

If both owned and subcontracted transport are included with 50 arcs each, and a delay of up to 5 is allowed as well, we obtained an ACT of 19.9685

Unfortunately, Method 1.2 cannot process Problem 3 via pulp in our computer, since the solve stage demands space requirements that exceed the capacity of the computer. In this method, the problem has 259, 490 variables and 222, 622 constraints, therefore finding a different solver that allows this size of problem as input will likely still be inefficient.

Method 2.1.

Problem 1. By proceeding as in the previous sections to solve the problem 1, we obtain an ACT of .9130 with a standard deviation of .1635, just 16 variables on average with a standard deviation of 2.0493 and 15.6 constraints on average with a standard deviation of 2.4576.

Problem 2. For problem 2 the results are as follows: ACT of 2.6691 with a standard deviation of .0820 where most of the time is spent on the pre-processing part, having on average a computational time of 2.0668. Average amount of variables is 31 with a standard deviation of 4.1472 and average amount of constraints equal to 33.7 with a standard deviation of 4.2673.

Problem 3. In problem 3, we obtain that the ACT is 66.4086 with a standard deviation of 2.1258 and again most of the computational time is concentrated on the pre-processing stage, with an average of 63.3958 and a standard deviation of 2.3038. The average amount of variables is 93.5 with a standard deviation of 17.8955 and an average amount of constraints of 106.3 with standard deviation 17.2049. Surprisingly, by increasing the amount of arcs from 250 to 400, we obtain a ACT of 70.2576 with a standard deviation of 1.75 and overall similar values to those obtained from the previous problem.

If transport is owned instead of subcontracted, with three consecutive possible departure times per barge, we get an ACT of 74.8002 with a standard deviation of 2.6577 with an average computational time on the pre-processing stage of 67.3614. The amount of variables is much more than before, since now there is an extra three variables per owned transport. Thus we obtain an average amount of variables of 933.5 with a standard deviation of 65.1678 and an average amount of constraints of 1103.2 with a standard deviation of 22.8245. If the amount of owned arcs increases to 400, then the ACT rises to 828.5723 with an amount of variables of 1819 and 1770 constraints.

If a delay of at most 5 is allowed, we obtain results quite similar to the non-delay case, with a ACT of 70.8767. If both subcontracted and owned transports are considered, as well as delay allowed, we get an ACT of 72.2120.

---

than a subcontracted barge because the cost of an owned trip is 40, and the average cost of a container in a subcontracted barge is 3.

Average computational time	Problem 1	Problem 2	Problem 3	Problem 4
Method 1.1	60.6225	> 3600	N/A	N/A
Method 1.2	2.4090	16.0368	N/A	N/A
Method 2.1	.9130	2.6691	66.4086	101.4469
Method 2.2	.7470	1.6355	6.6938	74.3075

Standard deviation computational time	Problem 1	Problem 2	Problem 3
Method 1.1	6.0782	N/A	N/A
Method 1.2	0.16	.9656	N/A
Method 2.1	.1635	.0820	2.1258

Amount of variables (average)	Problem 1	Problem 2	Problem 3
Method 1.1	11,200	N/A	N/A
Method 1.2	2300	18,400	N/A
Method 2.1	16	31	93.5

Amount of constraints (average)	Problem 1	Problem 2	Problem 3
Method 1.1	1900	N/A	N/A
Method 1.2	1900	16,300	N/A
Method 2.1	16	34	106.3

Average computational time	Delay	No Delay
Method 1.2 on Problem 2	17.8946	16.0368
Method 2.1 on Problem 3	70.8706	66.4086

Average computational time	Subcontracted	Owned	Mixed w/ delay
Method 1.2 on Problem 2	16.0368	21.6002	19.9685
Method 2.1 on Problem 3	66.4086	74.8002	72.2120

TABLE 5.1. Comparison of Methods for different problems

Method 2.2.

Since the changes from this method to the previous one imply changes only on the ACT, we show this on the table only.

We conclude that Method 2.2 gives satisfying results for an instance of considerable size, therefore for the rest of the project we work with Method 2.2. The results on the tables agree with intuition, with delay increasing computational time, as well as owned transport.

Heuristics.

We analyze the performance of the heuristics only for Problem 4. In ACT for this problem, the first heuristic takes 49.4613 on preprocessing, and only 2.0996 on the running the heuristic. It provides a solution which is, on average 4.89% above the optimal solution, with all values ranging between 3% and 8%. However, it should be noted that this is very instance dependent, since with different parameters for the space-time graph and for the problem, the values have registered as high as 28.68% above optimal solution. The Heuristic 2 performs quite poorly in terms of ACT: 98.5392 plus the 49.4613 on preprocessing make it even slower than the optimal solution Method 2.2. Also, the solution on average is 4.74% above the optimal solution, which is barely an improvement over Heuristic 1.

This implies that, given the computational burden of Heuristic 2, and the little gain to obtain from it, it might be better to use Heuristic 1 in most cases, and otherwise stick with Method 2.2. This suggests that the shortest path approach might not be a bad approximation, especially since the amount of paths for each order is not so large. It is also extremely fast when compared to all other methods.

Further improvements to the Method 2.2. As instances grow larger, with more orders, the format in which to present the optimal solution given by the method becomes non trivial. In order to

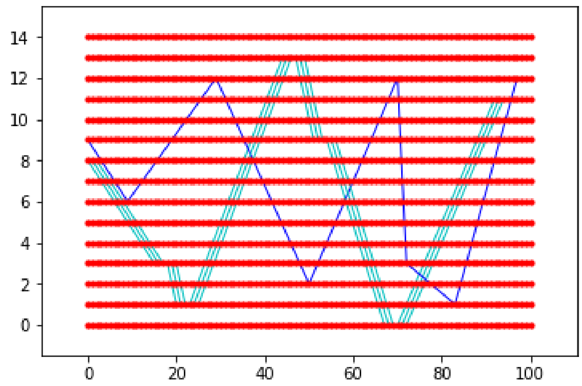

FIGURE 5.6. Journeys of an owned resource and a subcontracted resource

make the solutions readable, we adapted it to output a text file in which the flow of each order is specified. We refer to this file as the full text file. An example can be seen in the Appendix

Working with the model also revealed two important lacks that needed to be addressed:

- More realistic journeys of resources: a resource goes throughout several locations sequentially as in the Figure 5.6
- We are not considering handling times nor handling costs

The first lack was incorporated into the last model, whereas the second is partially dealt with through our multi-criteria analysis.

In the rest of the thesis, we refer to this last path-based model as our model.

5.3. **Future work.** Here we point out some possible improvements that can be done in the current model, and some routes for future research.

Firstly, the unusual integrality of solutions as observed at the end of subsection 4.3 is an interesting route of future research.

Also, the way in which solutions are visualized is via the full text file (see Appendix) since it was considered the most practical way to have all information of the optimal solutions obtained. However, it may be of interest to investigate whenever there is a better way to visualize the optimal solutions given by the methods.

Additionally, as the number of orders becomes increasingly larger, so does the number of variables. If the number of variables becomes too large it may be useful to do a column generation approach on the LP to cope up with this (as it is often done on the path-based formulation of MCMCF) or heuristic approaches.

Other improvements that could be done are:

- More realistic parameters: for example, make a distance matrix for the set of locations, from which to obtain costs, making sure this does not conflict with the triangle inequality observed after Assumption 3. According to [15] average speed of trucks, trains and barges are 55 km/h 35 km/h and 15 km/h respectively, which can serve to build the matrix.
- Increase exhaustiveness of the method: for example, handling costs including loading and unloading costs are not considered. Also, make a distinction between different locations (a network operator would likely have preference of leaving idle containers at some locations).
- The tree-like structure of the static graph (a case seen in waterways such as the port of Rotterdam) may imply certain properties of the space-time graph extension. This may translate in improvements on computational time of the calculation of the solution.

6. Multi-objective optimization: What to optimize?

When booking a flight, most of the people look for the cheapest flight available. If this flight is over a long distance to a terminal that is not so busy, quite likely it will not be a direct flight, with one or two connecting flights on the journey. These connections have sometimes a very short time-window between arrival and departure of flights, and although the booking is done in a way that the connection should be possible, the actual arrival time of the flight has always some uncertainty; the handling time for arriving passengers to connecting flights on each terminal is also uncertain. These two factors make journeys with short time between flights not that robust to some extent. Thus, if two different choices for journeys are offered with very similar costs, but one of them is not robust, most people would choose the most robust opportunity for peace of mind. Of course, depending on the customer, some other aspects might be of importance, for instance, it might be good to avoid flights with very inconvenient departure times, and also, depending on the needs of your trip, your departure and arrival dates might be soft dates, which opens new possibilities of journeys. The same phenomenon happens with container transport: although cost is the dominant factor, it is desirable to have transportation that fulfills other attributes to some extent. This brings us to the realm of multi-criteria or multi-objective optimization.

Research has been carried out on which factors are considered relevant, and on whether there is an appropriate weight for each attribute [3]. However, no rigorous attempt to quantify these attributes has been made. This might come as no surprise, since the meaning of concepts such as "robustness" and "flexibility" often varies, and even if the meaning is stated strictly, finding a mathematical expression that encompasses the whole concept is not trivial, and subject to interpretation. Nevertheless, a mathematical expression that partially expresses the concepts can be constructed, and although there can be debate over the accuracy of the expression given, for our purposes, the usefulness of the attributes lies much more on its capacity to provide new possible optimal solutions, rather than on the conceptual accuracy of the attributes themselves: the research done here is meant to be used as a decision support tool for planners, by providing them with a set of possible solutions to choose from.

In order to carry over a multi-criteria analysis, attributes must be constructed. In this section we develop the concepts on which to build the objective functions we seek. Furthermore, an analysis of these optimization objectives will be carried out, based on different performance indicators found in the literature, and on multi-criteria analysis.

6.1. **Attributes from concept to implementation.** As was stated in the literature review about the cost function, a study on the theory and practice of the different attributes that are of importance in a multimodal transportation network led to the following definitions

- *Robustness* is the capacity of a plan to overcome uncertain events in the future and still be carried out as planned.
- *Flexibility* is the capacity of a plan to adapt to uncertain events, when these force the plan not to be able to be carried out anymore.
- *Reliability* is the capacity of a plan to arrive on time to its destinations

In practice, an "uncertain event" on a transportation network can come in many different forms: disturbances in expected handling times upon arrival on a terminal, arrival of new orders, assignment of timeslots for arrival of certain modes, and so on (the relevance of these uncertainties usually varies depending on the different time-scales); planners often deal with these uncertainties via strategic behavior, that is, by acknowledging these uncertainties and taking them into consideration when making their decisions. However, the uncertain events that can affect the operation of a transportation network are vast, and once again, some compromise must be made for the sake of keeping a model manageable and meaningful at the expense of exhaustiveness.

In our model (that is, the model based on Method 2.2 shown in the previous section), we consider the uncertain events of travel times and handling times on terminals. Therefore, for our case, the definitions of the concepts can be read as follows:

- Robustness is the capacity of a plan to overcome delays in travel times and handling times on terminals and still be carried on as planned.
- Flexibility is the capacity of a plan to adapt to delays in travel times and handling times on terminals when these force the plan not to be able to be carried on anymore.

In our model we disregard reliability, since all routes are assumed to arrive on time within a certain margin of tolerance (via the inverse arcs). We do, however, take into account the lateness of a whole plan via the attribute customer satisfaction.

6.1.1. *Robustness.* To illustrate the meaning of robustness in our model, we first show how we would like to quantify robustness for a simple case. In a space-time network with a number of orders, we want to give a numerical value to each solution of the problem, that is, a value per transportation plan. In the case of a single order, we would assign a value to a path; consider a path P such as the one in Figure 6.1 with an OD pair $((A,0),(C,5))$, that is, with a source on location A at time 0 and sink on location C at time 5. The robustness value is meant to represent how likely this plan can endure despite delays in travel times and handling times, that is, we need to see how likely the transportation mode from location B at time 3 to location C at time 4 (arc from $(B,3)$ to $(C,4)$) will be able to take place for path P despite delay in travel time from A to B and handling times at B. If the resource doing the trip $((A,0),(B,1))$ is also the one on the trip $((B,3),(C,4))$ then there will be no handling at location B, thus the flow of containers through this path will certainly make this connection. Otherwise, the handling at location B will depend on these factors:

- The number of containers going through arc $((A,0),(B,1))$, since all of these will be handled at location B, which is, in this case, the flow going through path P.
- The number of timestamps available from the estimated time of arrival to arc B, which is 1, and the time of departure of the trip to C, which is 3.

This kind of link is what we refer to as events, and we define the robustness of a plan with respect to the robustness of these events.

Definition 6.1. For a path P on a space-time graph, we say that $e = ((A,t_0),(B,t_1),(B,t_2))$ is an **event** of the path P if the path $((A,t_0),(B,t_1),(B,t_1+1),...,(B,t_2),(C,t_3))$ for some $C \neq B$ is a subpath of P, and the resource of the trip $((A,t_0),(B,t_1))$ is a different resource than the one of trip $((B,t_2),(C,t_3))$. Also, $e = ((A,t_0),(B,t_1),(B,t_2))$ is an event of P if the path $((A,t_0),(B,t_1),(B,t_1+1),...,(B,t_2))$ is a subpath of P and (B,t_2) is the last node on P. If the event is of the latter form we refer to it as the **last event** of P. We use the short notation $e \in P$ to denote that the event e is an event of the problem P. For Pr a path-based multi-commodity flow problem on a space-time graph, we say that e is an event of the problem Pr if it is an event of a path P of an OD pair. We use the short notation $e \in Pr$ to denote that the event e is an event of the problem Pr. If x_P is the flow variable of a path P, and F is a solution to Pr, the flow on an event is defined as $F_e = \sum_{P \in P(e)} x_P$ where $P(e) = \{P \in \cup_k P(k) | ((A,t_0),(B,t_1)) \in P\}$ (see equation 4.2 for notation).

In the path of Figure 6.1 if the resource of edge $((A,0),(B,1))$ is a different resource from the one in edge $((B,3)(C,4))$ the path has the event: $e_1 = ((A,0),(B,1),(B,3))$. In any case, the last event $e_2 = ((B,3),(C,4),(C,5))$ in on the path. Our main assumption when determining robustness of an event is that the information in the three elements that constitute the event and the flow of the event are necessary and sufficient to determine the robustness of the event. More specifically, we determine the robustness of an event via a measure of robustness, which depends on the amount of flow f and the number of timestamps t.

Definition 6.2. We say that a function r' is a **measure of robustness** if $r' : \mathbb{R}^+ \times \mathbb{Z}^+ \to [0,1]$ and the following hold:

- $r'(0,t) = 1$ for all t, $\lim_{f\to\infty} r'(f,t) = 0$ for any fixed t, $r'(f,t)$ is a decreasing function of f for any fixed t.
- $r'(f,0) = \epsilon$ for $\epsilon > 0$ a number close to zero, $\lim_{t\to\infty} r'(f,t) = 1$ for any fixed f, $r'(f,t)$ is an increasing function of t for any fixed f.

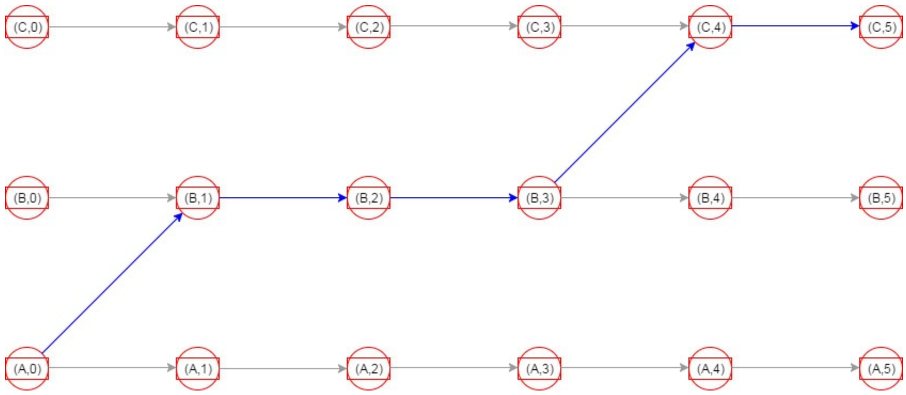

FIGURE 6.1. Robustness of a path (in blue).

We define the robustness $r(e, f)$ of an event $e = ((A^e, t_0^e), (B^e, t_1^e), (B^e, t_2^e))$ with flow f as $r(e, f) = r'(f, t_2^e - t_1^e)$. If the second argument f is omitted then $r(e) = r'(F_e, t_2^e - t_1^e)$.

Thus, the two variables of the function $r'(f, t)$ are thought of as denoting the amount of flow of the first arc of the event and the timestamps available. The properties of the measure of robustness attempt to be the minimum requirements we would expect for a way to measure the robustness of such an event: if the amount of flow is small with respect to the number of timestamps, then quite likely (close to 1) the event will be a success in terms of arrival before departure time of the next transportation mode, whereas if the amount of flow is large with respect to the number of timestamps, then it might be difficult to make this connection. We should note that whenever a specific flow is considered large or small depends of course on the units considered for each timestamp, but in any case, the (approximate) equalities and the limits must hold. The reason for the small value $r'(f, 0) = \epsilon$ is that just-in-time connections might not be very robust, but they can still be made.

Definition 6.3. Let F be a solution flow for a path-based multi-commodity flow problem Pr on a space-time graph. We define the **robustness of the solution** $R(F)$ as the product of the robustness of the events of the plan. that is

$$R(F) = \prod_{e \in Pr} r(e) = \prod_{e \in Pr} r'(F_e, t_2^e - t_1^e)$$

Thus, in our solution, in order to quantify the robustness, a robustness measure r' must be specified. We propose the function

$$r'(f, t) = \begin{cases} e^{-\lambda \frac{f}{t}} & \text{if } t > 0 \\ e^{-\lambda \frac{f}{.5}} & \text{if } t = 0 \end{cases}$$

with $\lambda > 0$ a parameter to be specified depending on the units that represent each timestamp. For simplicity, if an event e is such that $t_2^e - t_1^e = 0$ we write $\frac{F_e}{t_2^e - t_1^e}$ when we actually mean $\frac{F_e}{.5}$. Then robustness is defined as

$$R(F) = \prod_{e \in Pr} r'(F_e, t_2^e - t_1^e) = \prod_{e \in Pr} e^{-\lambda \frac{F_e}{t_2^e - t_1^e}}$$

Maximizing the robustness function is the same as maximizing the logarithm of the robustness function, so that

$$\log R(F) = \log \prod_{e \in Pr} e^{-\lambda \frac{F_e}{t_2^e - t_1^e}} = \sum_{e \in Pr} \log e^{-\lambda \frac{F_e}{t_2^e - t_1^e}} = \sum_{e \in Pr} -\lambda \frac{F_e}{t_2^e - t_1^e} = -\lambda \sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e}$$

Since $\lambda > 0$, maximizing the robustness function is equivalent to minimizing

$$\sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e} \tag{6.1}$$

Notice that this expression is linear with respect to the flow path-based variables x_P, and the sum depends only on the events of the problem, which are independent of the solution proposed. Therefore this expression can be constructed as a linear objective on a LP solver. We refer to the expression 6.1 as the **robustness expression.**

Notice that $R(F)$ tends to decrease if the number of events in the problem $|\{e \in Pr\}|$ increases. For this reason, in order to treat instances of different sizes in a similar way, it is practical to introduce the **geometric mean robustness of the solution** $MR(F)$ defined as

$$MR(F) = (\prod_{e \in Pr} r(e))^{\frac{1}{|\{e \in Pr\}|}}$$

Then we obtain, using the same robustness measure as before

$$\log MR(F) = \frac{\log \prod_{e \in Pr} e^{-\lambda \frac{F_e}{t_2^e - t_1^e}}}{|\{e \in Pr\}|} = \frac{-\lambda}{|\{e \in Pr\}|} \sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e}$$

To wrap up, the objective $\sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e}$ is the linear expression that represents robustness when the measure of robustness is chosen to be $r'(f,t) = e^{-\lambda \frac{f}{t}}$. Robustness and mean robustness is maximized when this expression is minimized, regardless of the $\lambda > 0$. The equalities $\sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e} = -\frac{\log R(F)}{\lambda}$ and $\sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e} = -\frac{\log MR(F)|\{e \in Pr\}|}{\lambda}$ allow us to calculate the robustness and the mean robustness of a solution, and give a value we should be choosing for the robustness expression when we are aiming for a given robustness or mean robustness value. Also, a good estimate of λ should be chosen when the unit of the timestamps is determined, from the interpretation given to $r'(f,t) = e^{-\lambda \frac{f}{t}}$.

Example. Effect of the robustness expression as constraint

In order to gauge the influence of robustness, here we present some preliminary results about the impact of the robustness expression as a constraint on the MCMCF problem on a space-time graph with all subcontracted transport and no delay. Suppose we want to solve an instance of Problem 2 (see previous section) with 200 orders instead of 20, and that each timestamp represents one hour. With the statistical data obtained form our research (classified data) we can estimate that quite certainly (with probability .90) 10 containers can be handled in one hour[4], that is to say

$$r'(f,t) = e^{-\lambda(10)} > .90 \implies -10\lambda > \log(.90) \implies \lambda < .0105$$

We fix $\lambda = .01$. By solving the MCMCF problem without involving robustness, we obtain an optimal solution F_1 with a mean robustness of

$$MR(F_1) = \exp(\frac{-\lambda}{|\{e \in Pr\}|} \sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e}) = .7761$$

and a cost $C(F_1) = 204,399$. If we wish to improve the mean robustness slightly to .78, we input the constraint $\sum_{e \in Pr} \frac{F_e}{t_2^e - t_1^e} = -\frac{\log MR(F)|\{e \in Pr\}|}{\lambda} \leq -\frac{\log(.78)|\{e \in Pr\}|}{\lambda} = 7602.9175$ and rerun the program. This constraint will guarantee the solution will have a mean robustness of at least .78

This change, as little as it may seem, already brings some significant differences on the new solution F_2. F_1 and F_2 differ in the transportation planning of 21 out of the 200 orders, despite the fact that $C(F_2) = 204,400.96 \approx C(F_1) + 1$ and both have same number of trucks used, that is, the plan is altered without compromising cost.

By changing the target mean robustness to .79, we get a solution F_3 whose plan is different from the plan in F_2 on 10 orders. The cost $C(F_3) = 204,608.21 \approx C(F_2) + 208$. It might seem odd that the difference in terms of different planning of orders between F_3 and F_2 is not as much

[4]Since the average handling time per container is 3.5 seconds. The claim remains quite arbitrary, but if the reader still feels skeptical, she or he might find relief in the fact that the estimate of lambda has no impact on the results: This value determines the robustness and mean robustness of each solution, but all solutions maintain the same order with respect to each other (in terms of robustness or mean robustness) regardless of the lambda chosen.

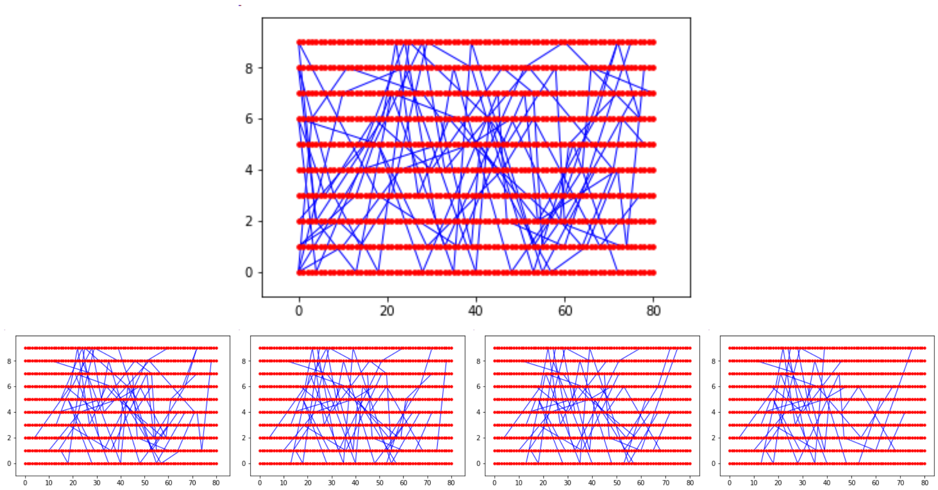

FIGURE 6.2. On top, a space-time network with all available transportation arcs (no trucks). On bottom, the arcs that are used by the solutions as the mean robustness decreases (from left to right)

as the difference of F_2 and F_1 given the changes in mean robustness. This can be explained by the following fact: if one were to create a solution F' from the problem with the robustness expression constrained by $-\frac{\log(.77)|\{e \in Pr\}|}{\lambda}$ then one obtains a solution F' quite different from F_1 despite the fact that F_1 is on the feasible set. There can be many optimal solutions, and which one is picked depends on the solution method of the algorithm used by the solver.

To graphically see the effect of mean robustness, In Figure 6.2 we see the transport arcs used in each plan as the mean robustness of the solution is increased (excluding truck arcs), for an instance with the same characteristics as the example. Comparison between the full text files (as defined at the end of Section 5.2) of the solutions, reveal the following tendencies about the more "average robust" solutions:

- They tend to prefer paths that have less connections between different resources.
- They tend to prefer earlier arrival.
- They tend to prefer paths connected by the same resource.

The first characteristic may be helpful to prevent from possible handling costs incurred in terminals. The second characteristic can be beneficial so that future resources are allocated for future uncertain happenings, but it can affect if there are costs for long idle times at destination terminals.

Since in this model the only paths with a robustness of 1 are the paths corresponding to trucks, if the mean robustness is close to 1, then the solution obtained is close to making all transportation via truck. Thus mean robustness and cost are competing attributes.

Owned transport and delay allowed. In the case where delay is allowed on the arrival of orders, the robustness expression on (expression of robustness) may have some negative terms, since $t_2^e - t_1^e < 0$ for the last event of delayed paths. This negative numbers are indeed senseless because no robustness is added by arriving late. In terms of robustness, it is more convenient to interpret the last events of delayed paths as events that arrive just in time, Therefore, if the event e is the last event of a delayed path, we add the the term $\frac{F_e}{.5}$ to the robustness expression. This expression is thus transformed to

$$\sum_{e \in Pr} \left(\frac{F_e}{t(e)}\right)^2$$

Where $t(e) = t_2^e - t_1^e$ if $t_2^e - t_1^e > 0$ and $t(e) = .5$ otherwise.

With respect to owned transport, robustness has no relationship with the cost of the arcs used, nor with the objective function of cost (which are the characteristic features of owned transport).

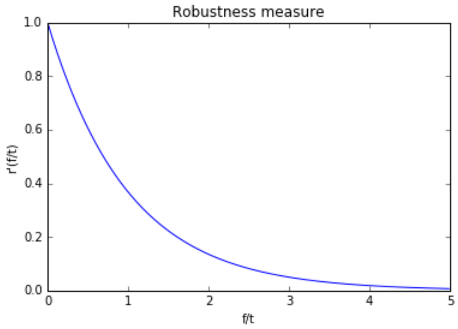

FIGURE 6.3. Robustness measure

Thus no change has to be done in order to enable the use of owned transport when measuring robustness.

About the choice of the measure of robustness (optional read). The model proposed depends on the function of robustness measure $r'(f,t) = e^{-\lambda \frac{f}{t}}$. In principle, any function with at least the characteristics of a measure of robustness can be a candidate. Given the interpretation that we give to these measures of robustness (how likely will a connection with t timestamps be done if there is f flow arriving at the handling terminal), some other properties may be desired for the robustness measure. For instance, the approximate equality $r'(cf, ct) \approx r'(f,t)$ for $c \in \mathbb{Z}^+$ seems suitable if containers are handled at an approximate equal speed regardless of the amount of containers. Although this is not always the case, especially if the number of containers or timestamps is quite low (as was shown in practice, there is some setting up time for a resource to be able to handle or be handled containers), these values should be fairly close.

The exponential robustness measure we propose fulfills $r'(cf, ct) = r'(f,t)$. In the figure on 6.3 we see a sketch of the graph with respect to f/t with $\lambda = 1$. This exponential function is handy for the computations and realistic enough, but the derivative of the function is

$$\frac{\partial r'}{\partial \frac{f}{t}} = -\lambda e^{-\lambda \frac{f}{t}}$$

which is an increasing function of f/t. Perhaps a more realistic proposal would be $r'(f,t) = e^{-\lambda(\frac{f}{t})^2}$ since its derivative has a more close behavior to the actual phenomena: the derivative of the function is

$$\frac{\partial r'}{\partial \frac{f}{t}} = -2\lambda \frac{f}{t} e^{-\lambda(\frac{f}{t})^2}$$

whose shape is as in Figure 6.4 with $\lambda = 1$. The function r' decreases at a slow rate close to $f/t = 0$, with the greatest rate of decrease on $f/t = \frac{1}{\sqrt{2}\sqrt{\lambda}}$, and a lower rate for values greater than $\frac{1}{\sqrt{2}\sqrt{\lambda}}$ (this behavior is even more prominent with bigger powers, but perhaps second degree suffices).

However, with such robustness function, the robustness expression would be

$$\sum_{e \in Pr} (\frac{F_e}{t_2^e - t_1^e})^2$$

which is a quadratic expression of the flow path-based variables x_P, and thus, unable to put as a linear expression.

For the scope of this research, the proposed $r'(f,t) = e^{-\lambda \frac{f}{t}}$ serves well, but a study of more involved robustness measures is worthwhile, as well as the challenges they imply.

6.1.2. *Flexibility.* We have defined flexibility as the capacity of a plan to adapt to delays in travel times and handling times on terminals when these delays force the plan not to be able to be carried on anymore. To calculate the flexibility of a single path (simple case with a single

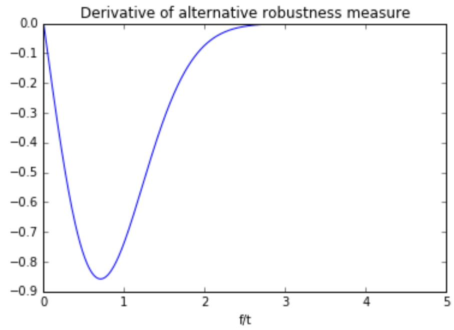

FIGURE 6.4. Derivative of alternative robustness measure $\frac{\partial r'}{\partial \frac{f}{t}} = -2\lambda \frac{f}{t} e^{-\lambda(\frac{f}{t})^2}$

commodity, and one path carries all the flow) such as the one described in Figure 6.5 we first identify those links in the path that could be problematic in terms of flexibility as we have defined it. As in the case of robustness, we refer to these problematic links as events. In this path, with OD pair $((A,0),(C,5))$, if there was a delay on the transportation arc from $(A,1)$ to $(B,2)$ or on the handling time at B such that the connection with the arc $(B,3)$ to $(C,4)$ is lost (in this case, the delay made the trip arrive at time 4), there is still the possibility to take the arc from $(B,4)$ to $(C,5)$. The flexibility of this path is defined in terms of the cost of this alternative route with respect to the cost of the original route. In the case where there is more than one event on the path, the flexibility of the path is done with respect to the cost of the alternative routes corresponding to each of these events.

In order to state unambiguously the flexibility of a flow, a series of definitions are necessary. The definition of event is as it was explained on Definition 6.1, except that in this context, last events are not considered events.
Definitions.

- For a path P on a space-time network and an event $e = ((A,t_1),(B,t_2),(B,t_3))$ on the path, we define the subpath P_e with respect to e as the subpath of P that contains all the nodes from (B,t_3) onward. In the case of the example in Figure 6.5, for the event $e = ((A,1),(B,2),(B,3))$ the subpath defined is $P_e = ((B,3),(C,4),(C,5))$
- For a solution F of a multi-commodity flow problem on a space-time network G we denote by $G\backslash F$ the space-time network G whose arc's capacity has been lowered according to the flow of F, that is, the capacity of an arc in $G\backslash F$ is the capacity of the arc on G minus the flow passing through that arc on F
- For a pair of nodes (A,t_1) and (B,t_2) on a space-time graph G and a positive real number r we denote by $\text{mincost}((A,t_1),(B,t_2),r)_G$ the cost of the optimal solution of the minimum cost flow problem with source node (A,t_1), sink node (B,t_2) and flow r
- For a path P with flow x_P of a solution F of a multi-commodity flow problem on a space-time network G and an event $e = ((A,t_1),(B,t_2),(B,t_3))$ on the path, we define the anti-flexibility $\varphi_{G\backslash F}(e,x_P)$ of the event as the least cost that would be incurred if the trip scheduled from A at time t_1 to B at time t_2 would arrive one timestamp after time t_3 to B. That is,

$$\varphi_{G\backslash F}(e,x_P) = \text{mincost}((B,t_3+1),(S_P,t_P),x_P)_{G\backslash F} - C(P_e)x_P$$

Where $C(P_e)$ is the cost of the subpath P_e and (S_P,t_P) is the last node on P. Notice the dependency of the min-cost algorithm on the solution flow F as well as on the G, that is, the capacity of the arcs on G are lowered corresponding to the flow F

We call the above anti-flexibility because $\varphi_{G\backslash F}(e,x_P)$ decreases as the flexibility of the event increases, according to our definition of flexibility.

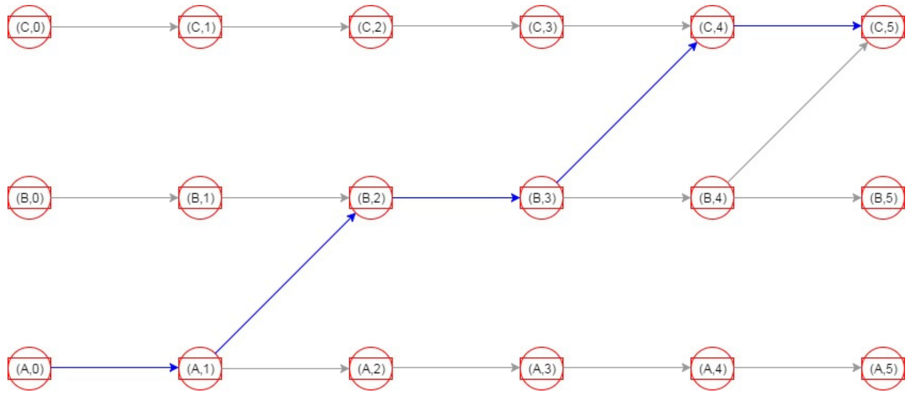

FIGURE 6.5. Flexibility of a path (in blue)

- For a solution flow F of a path-based multi-commodity flow problem on a space-time graph G and a robustness function r, we define its anti-flexibility $\phi_G(F)$ as

$$\phi_G(F) = \sum_{P \in F, x_P > 0} \sum_{e \in P} \varphi_{G \setminus F}(e, x_P)(1 - r(e))$$

In our case, the robustness function r used is the one implied by the exponential robustness measure $r'(f,t) = e^{-\lambda \frac{f}{t}}$ shown in the previous section. The last expression is a sum of all the incurred costs that could happen on the plan from delays, this is the expression we seek to minimize, however, it is far from linear in terms of the flow variables of the paths. Notice also that in order to calculate the anti-flexibility of an event of a path P, the value of the flow variable x_P must be known in advance, which is of course not the case. In addition, a constraint whose coefficients involve solving several min-cost problems can be very computationally heavy. Thus a linear expression(s) that overcomes these challenges is sought for in order to include it in a LP formulation. For this purpose, the following linearization is constructed.

Definition 6.4. For a path P on a space-time network G and an event $e = ((A, t_1), (B, t_2), (B, t_3))$ on the path, we define the *linear anti-flexibility* of the event $\iota_G(e)$ as

$$\iota_G(e) = C(P_e^d) - C(P_e)$$

Where P_e^d is the shortest (least costly) path on G from $(B, t_3 + 1)$ to (S_P, t_P). For a path P on a space-time network G, and a robustness function r, we define the *linear anti-flexibility* of the path $\iota_G(P)$ as

$$\iota_G(P) = \sum_{e \in P} \iota_G(e)(1 - r(e, c))$$

Where c is an arbitrary fixed number, preferably close to the average flow of a path. We fix c to be the lowest possible order size.

Thus, the linear anti-flexibility of a path is a coefficient relatively easy to calculate. With it, we define the *first linear anti-flexibility expression*

$$\sum_P \iota_G(P) x_P \tag{6.2}$$

This is a linear expression that is not so time consuming to calculate. Apart from the simplification of considering a shortest path instead of a min-cost flow, and the choice of an arbitrary c, a main drawback of this expression is that the anti-flexibility of elements is done only with respect to the graph G because a solution flow F is unavailable in advance. Thus an event that is marked as having low linear anti-flexibility (high flexibility) might actually not be so after the solution is derived, since the corresponding arcs might be saturated. If a solution flow F is given, we are able to give a *second linear anti-flexibility expression*

$$\sum_P \iota_{G \setminus F}(P) x_P \tag{6.3}$$

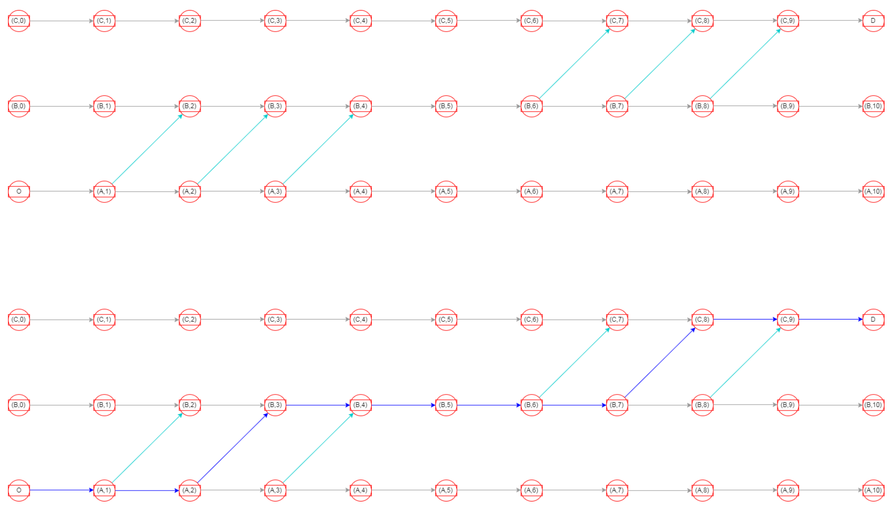

FIGURE 6.6. On top, a space-time graph with owned transport. On bottom, a path (in blue) in that space-time graph

Notice that $\iota_{G\setminus F}(P) > \iota_G(P)$. Computational results suggest that the second linear anti-flexibility expression, or linear combinations of expressions of this form, might be more representative of anti-flexibility.
Flexibility on owned transport and delayed delivery. So far we have discussed how to quantify anti-flexibility for a space-time graph with a cost function on its arcs. In principle this could be applied to any space-time network, but the anti-flexibility value obtained from it might not make sense anymore if the space-time graph is modeling a different phenomena, as it's the case with owned transport. In Figure 5.2 we see that one trip of an owned transport is represented via a series of arcs. If one were to naively quantify the (linear) anti-flexibility of a path (single commodity case) as in Figure 6.6 the value obtained would be 0, since for each event, the backup arc that would be considered is one corresponding to the same voyage with a later departure date; this value is not the penalty that would be incurred if the transport would have been missed. Also, all owned transport arcs are modeled to have a cost of 0, so in the case of the path on Figure 6.6 , as well as in the case of most paths on a space-time graph with only owned transport, the linear anti-flexibility of the path would be zero! For this reason, the implementation of anti-flexibility requires some careful tuning to be properly generalized to the owned transport case while keeping the meaning of the concept. For this reason, we build auxiliary graphs that help to quantify anti-flexibility and linear anti-flexibility as we intended. We start with linear anti-flexibility.

Definition 6.5. For a space-time graph G, we define its first auxiliary graph G' as the graph G with costs on owned transport (that is, with cost 0 and with a choice of departure time) arcs (a, b) changed to $c'(a, b) = \kappa \text{cost}(a, b)/\text{cap}(a, b)$ where $\text{cap}(a, b)$ is the capacity of the arc (a, b), $\text{cost}(a, b)$ is the coefficient in the objective function on the binary variable corresponding to the arc (a, b) (see Method 1.2 on Section 5.2) and κ is a constant.

For a path P on a space-time network G, we define the auxiliary graph G'_P with respect to P as the auxiliary graph G' such that for every arc in the path P that represents an owned transport, all other arcs that represent the same trip of the transport at different departure times are removed.

In the case of Figure 6.6, the graph G'_P would be just as the graph on bottom excluding all cyan arcs. G' attempts to be more accurate than G to quantify the costs incurred if there is a disturbance.

This auxiliary graphs allows us to calculate the linear anti-flexibility of paths:

Definition 6.6. For a path P on a space-time network G and a robustness function r, we define the *linear anti-flexibility* $\iota_G(P)$ of the path P as $\iota_G(P) = \sum_{e \in P} \iota_{G'_P}(e)(1 - r(e, c))$

	linear anti-flexibility	cost	anti-flexibility
F_1	5075	153,655	6079
F_2	3000	153,843	3502
F_3	2000	154,285	2665
F_4	1000	155,724	1567
F_5	650	156,471	1341

TABLE 6.1. Trade-off between anti-flexibility and cost

Notice that for a space-time network with no owned transport, $G = G' = G'_P$ for every path, thus this definition coincides with the previous one on the case of subcontracted transport. To calculate the anti-flexibility of a given solution, a different graph needs to be constructed

Definition 6.7. For a solution F of a multi-commodity flow problem, We define its auxiliary graph G'_F with respect to the solution F just as the auxiliary graph G' of G with the following modifications: For every owned transport arc (a, b) on G such that the flow of F on (a, b) is different from 0 all other arcs that represent the same trip of the transport at different departure times are removed, and the cost of the arc $c(a, b)$ is changed from $\kappa \text{cost}(a, b)/\text{cap}(a, b)$ to 0

The reasoning behind this definition is the following: if a solution is already chosen, then the departure time of the owned arcs that are going to be used are already fixed. Since the cost of an owned transport arc is done per trip, then no further cost is incurred if more containers are to be assigned to such arc. Therefore G'_F should be more representative than G' of the costs incurred if there is a disturbance.

Definition 6.8. For a solution flow F of a path-based multi-commodity flow problem on a space-time graph G and a robustness function r, we define its anti-flexibility $\phi_G(F)$ as

$$\phi_G(F) = \sum_{P \in F, x_P > 0} \sum_{e \in P} \varphi_{G'_F}(e, x_P)(1 - r(e))$$

Notice that in the case where there is no owned transport, the definitions of anti-flexibility coincide since $G'_F = G \backslash F$. Therefore linear anti-flexibility and anti-flexibility are generalizations of the previously defined concepts.

In order to incorporate delay, it was necessary and sufficient to ensure that when calculating the anti-flexibility (linear anti-flexibility) of an event, there was feasibility of the minimum-cost flow (shortest path) algorithm via truck arcs. This is always possible given our Assumption 3. No further modifications are needed since, based on our concepts, how delayed a path is has no relationship with its flexibility.

Example. Effect of the anti-flexibility expression as constraint

To explore the effect of the linear anti-flexibility expressions obtained, we consider a MCMCF problem as Problem 2 with 200 orders instead of 20, and 160 arcs, all subcontracted transport and no delays. The problem without any linear anti-flexibility constraint yields a solution F_1 with an anti-flexibility value $\phi_G(F_1) = 6078$, a linear anti-flexibility of $\sum_P \iota_G(P) x_P = 5075.08$ and a cost of $C(F_1) = 153,655$. By inputting the linear flexibility as a constraint, and playing with the value given, we obtain costs and anti-flexibility values as shown in the table 6.1. Note that the anti-flexibility is massively reduced by adding barely any cost with a linear flexibility value of 3000, and with a value of 2000, anti-flexibility is reduced by more than half!

We consider the problem with the added constraint of a linear anti-flexibility value of less or equal than 1000. We consider the solution $F_4 = F_a$ for this problem and we add the constraint

$$\sum_P \iota_{G \backslash F_a}(P) x_P + \sum_P \iota_G(P) x_P \leq 1000 * 2$$

	linear anti-flexibility	cost	anti-flexibility
F_a	1000	155,724	1567
F_b	855	156,078	1480
F_c	740	156,388	1165

TABLE 6.2. Use of constraints of the form of equation 6.3

The result of this is a solution F_b with values as shown in the table 6.2. We do yet another iteration and add the constraint

$$\sum_P \iota_{G\setminus F_b}(P)x_P + \sum_P \iota_{G\setminus F_a}(P)x_P \leq 1000 * 2$$

From which we obtain the solution F_c. This solution, as can be seen from the tables, has better values in terms of cost and anti-flexibility than the solution F_5

In terms of how different the solution are, F_2 has a different plan for 34 orders when compared to F_1, whereas F_4 has 39 orders with a different plan with respect to plan F_1. Plan F_a has 28 orders with different plan when compared to F_c. This suggests that the linear anti-flexibility constraint affects the plan considerably.

Comparison between the solutions reveals the following tendencies about the more "flexible" solutions:

- They tend to prefer trips with a cheap backup alternative in the future (notice that flexibility of a path can be negative, meaning that the backup route is cheaper!)
- They tend to prefer single link trips, or trips from the same voyage.

The phenomena of F_c besting the solution F_5 shown in the previous example is not always the case, but it is certainly common on a simulation such as Problem 2. Whenever it is more efficient to consider different constraints or to lower the value of the constraint is instance dependent. However, it is always the case that simply by adding the linear anti-flexibility constraint with relatively high values, the anti-flexibility can be reduced with little or no impact on the cost.

What does anti-flexibility numbers mean? Do we want to minimize anti-flexibility? As it was defined, anti-flexibility represents the expected extra costs that will be incurred on the plan, assuming there is a full refund for the arcs that were planned to be used but were not reached on time. Of course this refund does not necessarily take place, making the anti-flexibility more of a lower bound on the expected extra costs. If the expected costs are to be minimized, then it may be appropriate to minimize the lower bound on this expected cost, that is, the sum of costs and anti-flexibility. Unfortunately, anti-flexibility cannot be put on the LP, so one must rely on the use of the linear anti-flexibility to reach a low value for the sum of costs and anti-flexibility.

For the case of one commodity that can be served with a single path, given a collection of possible solution paths, observe that if the anti-flexibility of a path is minimized, then the path obtained might have negative values, meaning that the backup paths are cheaper than the solution path. If this paths are much cheaper, then the anti-flexibility is much lower! Of course this wants to be avoided, and shows that the anti-flexibility is an expression that comes with trade-offs. Anti-flexibility measures how cost-effective are the backup plans with respect to the chosen plan, and although it is good to have these alternatives cost-effective, it is probably not good to have them much cheaper than the chosen plan.

How are anti-flexibility and robustness related? From the way they are defined, if a given solution is very robust, then there is little chance the plan will have to be altered, which implies the expected extra costs for this disturbances will also be lower. This is indeed the case as can be corroborated in Example 7.2.1. On the other hand, if a plan is very flexible, then robustness might not be so important, since even if the plan is not carried over as scheduled, this will have little impact on the cost. This does not imply that robustness is unimportant, since in some situations it can be vital to stick to the plan exactly as it is.

6.1.3. *Customer satisfaction.* For the point of view of a customer, perhaps the most important thing of an order is its timely delivery. However, as our meetings with different stakeholders revealed, this attribute is dependent on the client. That is to say, some clients might have no problem if their order arrives past the agreed arrival time, whereas for others it might be crucial to be on time. In our model, as was shown in the previous section, we can allow delay in some orders with a penalty on the cost. Here we improve our model, deattaching lateness from cost by changing the cost of the inverse arcs to 0 and make it into an independent attribute. We fix a maximum amount of lateness that can be allowed for each order, and in order to measure customer satisfaction of a solution plan, we observe how late is the arrival of each order, taking into account the priority of each client. Since the order cannot be considered delivered until every container has arrived to the destination, we make the following definition

Definition. For a solution flow F of a multi-commodity flow problem P on a space-time graph, and an order $o \in P$ of the problem , we define the delivery time of the order $d(o)$ as the maximum of the arrival times of the containers on that order.

Definition. For each order o with and OD pair, and $t \in \{0, 1, 2, ..., r\}$ a number of timestamps, we refer to the satisfaction $s(o, t) \in [0, 1]$ as the number that reflects how satisfied will be the customer of order o if the order arrives t timestamps after the due time. For a fixed o, we assume $s(o, t)$ to be decreasing on t. The maximum number of lateness $r < T$ is fixed for computational ease.

Notice that this can be extended to a case where there is also penalty for early arrival, or even further, for arrival at any specific timestamp per order. However, this is not done in the current research.

Definition. For a solution flow F of a multi-commodity flow problem P on a space-time graph and a family of numbers $w(o) \in [0, 1]$ such that $\sum_{o\in P} w(o) = 1$, we define the customer satisfaction as

$$(\sum_{o\in P} s(o, t)w(o))^2 \tag{6.4}$$

If the weights are chosen to be $w(o) = \frac{1}{|\{o\in P\}|}$ for all o we refer to it as the average customer satisfaction.

Customer satisfaction was implemented via several indicator variables: one per order, per number of timestamps delayed. However, given the addition of several binary variables, considering customer satisfaction comes with a computational burden.

Example: Effect of customer satisfaction as constraint

As with the previous attributes, we use an instance as Problem 2 with 200 orders to see the effect of the customer satisfaction expression as a constraint, with randomly generated weights. The maximum number of delayed timestamps r is set to 10. Without any constraint related to customer satisfaction, we obtain a solution F_1 with a customer satisfaction value of .8190. By increasingly constraining the value of customer satisfaction, we obtain the costs summarized in the following table.

	Customer satisfaction	Cost	Average Value per Container
F_1	.8190	60082	18.8167
F_2	.8525	60094	18.8205
F_3	.9	60548	18.9627
F_4	.95	62474	19.5659
F_5	.98	64488	20.1966

The table shows that substantial cost reduction can be obtained at the expense of customer satisfaction, i.e. timely delivery. In this case, the average computational time for calculating the solutions F_i was 10.2551 seconds, whereas in the case with no customer satisfaction variables and constraints considered it took only 6.4451 seconds.

Overall, by adding the attribute of customer satisfaction to the model, and stretching the possibilities of delayed containers, we obtain a new range of solutions and a way to compare their effectiveness.

6.1.4. *Other important criteria.* In this subsection we present other criteria that are useful and important to consider for solutions.

(1) Number of trucks used: As was mentioned before on the assumptions subsection, this attribute is highly correlated with the cost of the solution given the parameters of the problem. However it is still worthwhile to maintain it as an independent attribute to ensure number of trucks in a solution is below a certain threshold.
(2) Percentage of usage in a given transportation owned mode, that is, the capacity of an owned mode can be set to have a minimum. This was added since on repeated occasions during our meetings with different stakeholders, we were informed that this is a constraint for transport owners.

Another criteria that is very important is the environmental impact of the plan. However, this criteria is also highly correlated with cost, since trucks are the less environmental friendly transport, followed by train, and then barge. Environmental impact is not a competing attribute with respect to cost. Also, it is usually implemented in the same way cost is implemented (one impact per trip, or per container) so it is not of mathematical importance. For this reasons, it was considered not worthwhile to include environmental impact in the scope of this research

6.2. **Future work.** Here we point out some possible improvements that could be done with respect to the material shown in this section, as well as some routes of work that can be of interest.

- The principle used in customer satisfaction could be generalized for other purposes: indicator variables that show when is an order arriving at a terminal, to signal whenever it is actually arriving at a time when it can be dispatched or not. This is an alternative to using a hard constraint that disables the barge to arrive on a specific time, and it is better to do it this way to have the benefits of multi-objective optimization.
- Calculating the anti-flexibility value is very computationally expensive, in many instances it takes as long to compute than the whole LP to build and solve! Also, it relies on a minimum-cost algorithm that must be also improved, since the one used in this project does not always work well in the presence of owned transport (apparently the "parallel" arcs of owned transport with the same cost would make the algorithm crash). This was taken care of in the final result by ignoring to calculate the anti-flexibility when these issue presented itself, and given the high correlation that anti-flexibility and linear anti-flexibility have, the latter can serve as a substitute for reference. Nevertheless it is of interest to find a better way to calculate this concept.
- In this section effort was made to express the attributes in a linear way, for the sake of computational efficiency. This does not necessarily mean that one should restrict to linear expressions only. Perhaps a study in formulations that are non-linear using tools of optimization outside the scope of linear programming may be more accurate for some attributes, but the trade-off that must be payed in terms of computational cost has to be analyzed.
- It may be useful to extend the attributes presented in this section for other problems. For example, in the routing problem which is studied in the master thesis of Dylan Huizing (co-writer of Section 3), robustness of a given schedule of barges may be measured (assuming some flow of containers through the transport modes) considering the same uncertainties and theory developed for robustness in this section.

7. Multi-criteria analysis

In this section, we propose a subset of solutions for the multi-criteria multi-commodity flow problem on a space-time network with different approaches of multi-criteria optimization. Then we illustrate the solutions obtained by applying the methods proposed on an example.

The objectives considered are then:

(1) Cost: $\sum_k \sum_{P\in P(k)} C(P)X(P)$ (and trucks $\sum_k \sum_{P\in T\subseteq P(k)} X(P)$)
(2) Linear anti-flexibility: simple $\sum_P \iota_G(P)x_P$ (or relative $\sum_P \iota_{G\setminus F}(P)x_P$)
(3) Mean robustness: $\frac{-\lambda}{|\{e\in Pr\}|}\sum_{e\in Pr}\frac{F_e}{t_2^e-t_1^e}$ Where $\lambda = .01$
(4) Customer satisfaction: $(\sum_{o\in P} s(o,t)w(o))^2$

We seek to minimize the first expression, and to maximize the last two expressions. As was stated on the assumptions section of the space-time network modeling, cost and trucks are almost equivalent, however, the reason to include trucks is to make more flexible the range of solutions that can be obtained, as will be seen in the lexicographic subsection. The second expression is to be minimized to a certain extent, and so does the anti-flexibility expression $\phi_G(F) = \sum_{P\in F, x_P>0}\sum_{e\in P}\varphi_{G\setminus F}(e,x_P)*(1-r(e))$ which is minimized via the simple and relative linear anti-flexibility.

For each of the four principal objective functions, one scalarization was created. In each of these scalarizations, there is one of the objectives used as an objective function, and the other three can be constrained. We refer to the Cost LP as the linear program on which cost is the objective function. Similarly, we have Flexibility LP, Robustness LP and Customer Satisfaction LP. This will serve as the building blocks for the rest of the methods. We say that F_1 is a base solution if the cost of F_1 is the optimal cost of the Cost LP with no constraints on the other objectives.

7.1. A priori method: Lexicographic.

Here we use a lexicographic method to obtain a Pareto solution. For the lexicographic method we need to rank the objective functions in order of importance. As was stated previously, the lexicographic method can be very stiff in some problems, since it doesn't allow for any decrease in value from the top ranked objectives to increase less important objectives. For this reason, we also consider a slight variation of the lexicographic method: **when optimizing cost, look at the value of number of trucks and constraint the problem with the respect to this number of trucks instead of cost**. Notice that, strictly, with this procedure we cannot guarantee that the solution obtained is a Pareto optimal solution, therefore, if a Pareto optimal solution is needed, then one should use the usual lexicographic method. We propose several different orderings for obtaining the (Pareto) solutions.

First lexicographic order (minimizing costs and possible unforeseen costs).

(1) Cost (Truck)
(2) Linear anti-flexibility
(3) Customer satisfaction
(4) Mean Robustness

Second lexicographic order (minimizing costs and the need to change the plan).

(1) Cost (Truck)
(2) Mean Robustness
(3) Customer satisfaction
(4) Linear anti-flexibility

Third lexicographic order (minimizing costs and maximizing customer satisfaction).

(1) Cost (Truck)
(2) Customer satisfaction
(3) Linear anti-flexibility
(4) Mean Robustness

Each of these orderings emphasizes on one of the attributes constructed. It should be observed that given the relationship between robustness and flexibility (see Section 6.1.2) these attributes are kept apart in the first and second lexicographic orders. The solutions provided by the lexicographic methods proposed will serve as a starting point for our interactive method.

7.2. **Interactive method.** As was shown in the literature review about multi-criteria optimization, there are several steps that an interactive solution method must fulfill. Here we present our interactive method by showing these steps corresponding to our case.

(1) Provide the decision maker with the range that the different objectives can take, when possible. In this case, we have the following possible values:
   (a) Cost: $\mathbb{R}^+$
   (b) Anti-flexibility: $\mathbb{R}$
   (c) Robustness: $[0, 1]$
   (d) Customer satisfaction: $[0, 1]$

Notice that, in practice, cost will have an upper bound, and the rest of the attributes might have tighter constraints.

(2) Provide a starting solution(s) to the problem.

Notice that, in contrast to usual interactive methods, we do not require the starting solution to be a Pareto optimal solution. This is because although it is always preferable to use a Pareto optimal solution, obtaining them can be more time costly (for instance, it involves solving 4 scalarization problems via lexicographic method) Depending on the size of the problem at hand, the time available to use the method and the preference of the decision maker, this step might consist of providing only the base solution F_1 or any number of lexicographic methods from the previous section can be done either as a whole or partially (that is, stop at any attribute). Notice that although the base solution may not be Pareto optimal, it can provide valuable insight. The information of these solutions is gathered and kept for further assessment. Additionally, it is useful to build one optimal solution for a scalarization of each objective (other than cost), that is, the optimal solution of the scalarization of optimizing one objective if cost is allowed a 1% increase with respect to the optimal cost (or, if more margin is given, a greater percentage). The 1% increased value of cost is rounded up before adding it to the constraint, because it was observed to facilitate the computational load on the solver (this is further discussed on Subsection 7.3).

(3) Ask the decision maker for preference information.

In this step the influence of the decision maker is crucial. The information available from the solutions of the problem found so far must be assessed and used to make decisions. This information may include, but is not limited to:

- The value of the objectives of the solutions obtained so far, for example, the value of the base solution F_1 and the influence of achieving this cost in terms of the values of other attributes. (obtained from the lexicographic methods).
- A better assessment of the range of values from the objectives done in step 1, that is, the limitations of the values of attributes.
- The approximate time for obtaining a solution, and given the time left for using the method, the number of solution extra we can expect to obtain.

From this information the following decisions are needed available:

- What objective to optimize next?
- What range to restrict the rest of the objectives to?
- Is there a minimum capacity needed for owned transport? If so, what percentage?
- Is there a specific arc whose capacity should be updated?
- Is there a path whose value should constrained?

The last decision includes whenever some trip should not be used, some path must be fixed, some departure time of an arc must be fixed, etc. When implemented, instead of building linear programs from scratch, the code modifies the existing linear program to optimize the required objective and satisfy the constraints selected, thus saving computational time.

(4) Generate new solutions, show them and other possible relevant information to the decision maker.

A new LP is solved based on the questions obtained from the last step. The information of this solution is gathered and kept for further assessment

(5) Stop, or back to 3.

Depending on whenever a satisfactory solution has been provided yet, and on the time available, we either stop the method or reassess.

Given the goal of this tool, it is very important to keep it simple so that planners do not feel overwhelmed or frustrated when using the tool (at least at this first stage of development). Therefore the command inputs proposed in this method are quite simple.

7.2.1. *Example.* We illustrate the functioning of the interactive method proposed by applying it on a specific instance. This method is meant to be used by a decision maker (which in this case is a planner) whose choices will stir the method in a certain direction, thus the method will give different solutions depending on the DM's preferences. Therefore in this example we consider the presence of a hypothetical planner and conjecture the choices that this fictional DM may make.
Instance. We generate the problem by constructing the space-time graph and the orders to be dispatched on it. This instance was created via the same problem-generating code that was used for the Subsection 5.2 with the following parameters:

- Number of terminals: 15, listed from A to O
- Time Horizon (see Definition 4.1): 200
- Number of orders (see Subsection 1.2): 400 (OD pairs uniformly randomly generated, see start of Subsection 5.2)
- Number of journeys of subcontracted transport (see Figure 5.6): 12
- Number of journeys of owned transport: 4. With 3 different allowed departure times (see Method 1.2 on Subsection 5.2)
- Allowed delay (see Subsection 6.1.3): 10

The following values of the parameters were determined via statistical data (confidential) and advice from stakeholders in transportation networks.

- Capacity of an owned transport: 154
- Capacity of a subcontracted transport: unif$\{50, 51, 52, 53, 54, 55\}$ where unif takes any of the values inside brackets with equal probability
- Number of containers per order: unif$\{1, 2, ..., 30\}$

Other values that are determined are:

- Price per owned trip (see Method 1.2 on Subsection 5.2): 80
- Truck price: 40
- Price per container in subcontracted transport: $3 + \text{unif}\{-1, 0, 1\}$

The space-time network is presented in Figure 7.1. Additionally, we require the flow variables to be integer.

To generate the values required for Customer satisfaction as discussed in Subsection 6.1.3, we generate the random values $s(o, t) \in [0, 1]$ ensuring that for each o, $s(o, t)$ is decreasing with respect to t. The weights $w(o)$ are uniformly random generated by assigning $w'(o) = \text{unif}[0, 1]$ to each order and then setting the weight $w(o) = w'(o)/\sum_o w'(o)$
Interactive method on instance. After the problem has been set, we follow the steps on the interactive method.

(1) Provide the decision maker with the range that the different objectives can take, when possible. In this case, we have the following possible values:
   (a) Cost: $\mathbb{R}^+$
   (b) Anti-flexibility: $\mathbb{R}$
   (c) Robustness: $[0, 1]$

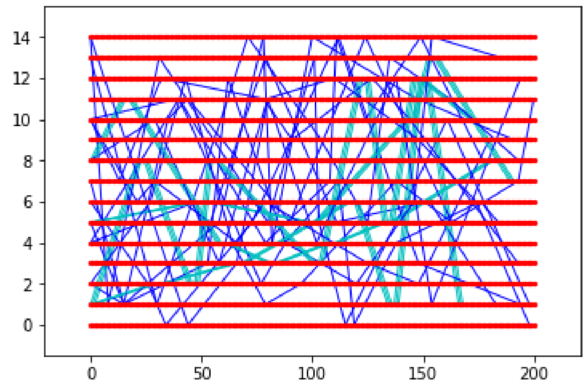

FIGURE 7.1. Example space-time network

	F_1	$F_{l,2}$	$F_{l,3}$	$F_{l,4}$
Cost	146, 387	147, 812	147, 695	147, 653
Mean Robustness	0.8778	0.8831	0.8834	0.8836
Anti-Flexibility	2365.55	442.46	439.82	442.65
Customer Satisfaction	0.8917	0.8907	0.8926	0.8926
Trucked Containers	3420	3420	3420	3420
Linear Anti-flexibility	116.79	20.73	20.73	20.73
Computational Time (seconds)	245	65	85	70

TABLE 7.1. Attribute values of the solutions of the lexicographic method

(d) Customer satisfaction: $[0, 1]$

(2) Provide a starting solution(s) to the problem.

We first obtain the base solution F_1 by solving the Cost LP with no constraint on the other objectives. This results in a solution F_1 with the characteristics shown in Table 7.1

Suppose the DM chooses to follow the first lexicographic method, then we add the constraint on trucked containers to be less than or equal to 3420 and optimize linear anti-flexibility. From this we obtain the solution $F_{l,2}$ and, following the lexicographic method, solutions $F_{l,3}$ and $F_{l,4}$ with attribute values as shown in 7.1. These solutions show a very significant decrease in terms of anti-flexibility, a slight change in mean robustness, and barely any change in terms of customer satisfaction. Notice that neither cost nor anti-flexibility follow a strictly monotonic behavior with respect to the solution number, despite the fact that this behavior is expected from a lexicographic method. For the case of cost, this is a consequence of the fact that we are using a slight variation of the lexicographic method where we do not allow trucked containers to increase, instead of cost. For anti-flexibility it is not expected to have any particular monotonic behavior since it is not constrained directly on the LP. Further analysis on the full transportation plan file corresponding to each solution reveals that despite their similarity in terms of attributes, solution $F_{l,1}$ and $F_{l,4}$ differ on the transport plan of 95 out of 400 orders.

It might not be surprising the attribute similarities between these solutions given how restrictive is the lexicographic method, but they provide us the insight of how much are the other attributes subject to change when the cost is (almost) rigid. Also, in this case, as it is often the case on lexicographic methods, the Pareto optimal solution $F_{l,4}$ is a very good proposal in terms of the attributes when compared to the other solutions obtained (that is because all the attributes have been optimized at some stage). This solution will serve as a good reference for the capabilities of the solutions in terms of the attributes.

We now calculate the set of solutions corresponding to optimizing each objective (that is, scalarization) allowing 1% increase of cost over the optimal cost. we write F_f, F_r, and F_{cs} for

	F_1	F_r	F_f	F_{cs}
Cost	$146,387$	$147,851$	$147,851$	$147,844$
Mean Robustness	0.8778	0.9056	0.8850	0.8859
Anti-Flexibility	2365.55	626.40	396.84	1498.30
Customer Satisfaction	0.8917	0.9069	0.8937	0.9619
Trucked Containers	3420	3435	3443	3454
Linear Anti-Flexibility	116.79	36.49	16.53	77.24
Computational Time (seconds)	245	60	169	729

TABLE 7.2. Attribute values of scalarization solutions

the solution corresponding to flexibility, robustness and customer satisfaction, respectively. The results are summarized in the Table 7.2

From the scalarization solutions obtained, we can see the extent at with the other attributes can be improved with as little as 1% increase on the cost: customer satisfaction can be improved .6 and mean robustness can be increased a bit less than .3. In terms of anti-flexibility, there can be a reduction of almost 2000 units (for discussion about the interpretation of the units of anti-flexibility, see the end of Subsection 6.1.2). It should be noted that the computational time to derive the solution F_{cs} is comparatively larger than the other ones.

(3) Ask the decision maker for preference information.

On this stage, the decision maker has to assimilate the information she/he has of the problem so far, provided by the previous steps. We conjecture here our fictional DM's train of thought:

The values of the attributes of the solutions provide a better idea of the range of the attributes: customer satisfaction is quite cost-effective to improve. Also, from $F_{l,2}$ and F_f we see that anti-flexibility can be reduced substantially for little cost. Additionally the DM knows that for this particular problem any plan with a customer satisfaction value over .90 is acceptable. Therefore the DM chooses the next solution to be the solution F_2 of the scalarization of optimizing cost with a constraint on linear anti-flexibility of 40 and a customer satisfaction of .9.

(4) Generate new solutions, show them and other possible relevant information to the decision maker.

We obtain a solution with the following values

	F_2
Cost	$146,395$
Mean Robustness	0.8761
Anti-Flexibility	755.53
Customer Satisfaction	0.9060
Trucked Containers	3420
Linear Anti-Flexibility	39.99
Computational Time (seconds)	229.96

This solution has just an increase of 8 in terms of cost, and it provides a very substantial decrease on anti-flexibility, as well as an increase in customer satisfaction.

(5) Stop, or back to 3.

The solution seems satisfactory, but the DM decides to try to improve the robustness of the solution without compromising the other attributes

3.

The DM decides to optimize with respect to robustness, with cost, linear anti-flexibility and customer satisfaction to be at as good as the values in F_2 (similar to a lexicographic method).

4.

We obtain a solution F_3 with the following values

	F_3
Cost	$146,395$
Mean Robustness	0.8812
Anti-Flexibility	799.44
Customer Satisfaction	0.9060
Trucked Containers	3420
Linear Anti-Flexibility	39.99
Computational Time (seconds)	2128.02

Notice that the computational time to obtain F_3 is quite long, therefore, depending on the time available, the DM may have stopped the simulation, and picked a solution from the solutions obtained so far (probably F_2). This of course depends on the importance the DM gives to improving slightly the robustness of the solution in this circumstances.

5.

Suppose the DM choose not to finish the simulation to obtain F_3 and reports F_2 as her/his solution of choice. The DM reviews the full text file (see Appendix) and is informed that solution F_2 uses a specific trip that has been canceled, which is represented by the arc $(('K', 35), ('L', 44))$ on the space-time network used for the problem. She/he is also informed that another trip from an owned transport used in F_2 will not be departing at the time the plan F_2 uses it, which is represented by arc $(('C', 48), ('I', 54))$. Additionally, a particular order has been specified to be served exclusively via truck, namely, order 2. The DM is therefore forced to go back to 3.

3.

With these new constraints, the DM has to make a choice depending on the time available: Either build a solution F_4 using constraints like the ones used to obtain the best solution so far, namely, F_2, or restart from step 1 considering the problem with this new added constraints as a new problem. Assuming a decision must be taken in a short time, the DM decides for the former

4.

The following three constraints are added to the LP (following notation on Equation 4.2):

$$\sum_k \sum_{P \in P(k)} X(P)\delta_{ij}(P) \leq 0 \quad \text{for } (i,j) = (('K', 35), ('L', 44))$$

$$\sum_k \sum_{P \in P(k)} X(P)\delta_{ij}(P) \leq 0 \quad \text{for } (i,j) = (('C', 48), ('I', 54))$$

$$X(P) = \text{number of containers on order 2, for } P \text{ the truck path of order 2}$$

We then optimize cost constraining linear anti-flexibility to 40 and a customer satisfaction of .9 and obtain the solution F_4 with attribute values:

	F_4
Cost	$148,352$
Mean Robustness	0.8761
Anti-Flexibility	801.94
Customer Satisfaction	0.9035
Trucked Containers	3474
Linear Anti-Flexibility	39.93
Computational Time (seconds)	215.33

5.

The DM is satisfied with the attribute values and proposes F_4 as a solution.

7.3. **Discussion and future work.** The above method proposes a hypothetical scenario where a planner uses this tool for the purpose of making a plan. However, the tool itself is an optimizing method and it could be used for other purposes: For example, given a particular problem, it can quantify the impact that certain attributes have on cost, such as the delayed delivery, or the minimum capacity on barges. This could illustrate how certain behaviors on the network are affecting the cost-performance of the network, or how some advantages are not being exploited.

In order to understand the tool, the user needs familiarization with the concepts used, such as space-time network, optimization, and a fair notion of the mathematics involved. Since the goal of this tool is to illustrate the benefits that can come from planners using multi-objective optimization, it is very important to keep things simple. Therefore the command inputs proposed in this method attempt to be used and understood (as much as possible) by a non-technical user. On the other hand, When compared to other interactive solution approaches in the literature, this method requires less input from the DM, and also less technical expertise. As future work, once the value of such a tool has been acknowledged by the planners, and the planners are committed to the interactive use of the tool, the complexity and usage of the tool can be increased. If more advanced interactive methods are developed, there should always be sensitivity into the context, use and level of involvement of the DM.

Also, future research must be done in order to improve visualization of results, since these may improve the understanding of the DMs on the problem and allow them to make better decisions. There are dedicated chapters about visualization in multi-objective optimization books (see chapter 8 and 9 of [4]). Perhaps the format of the output data can also be improved, according to the preferences of planners and logistic service providers.

About integral and its relationship to computational time, when calculating the solutions that allow 1% increase on cost and optimize another attribute (in example, F_r, F_f and F_{cs}), the integral of the flow variables made the code run quite slow, which was partially solved by rounding up the 1% increased cost and using this value on the constraint about cost. On the other hand, when non integer values were used to constraint robustness and cost was optimized, as in the example of Subsection 6.1.1 there seemed to be no consequence on the computational time. If the flow variables were considered continuous instead of integer, both problems performed quite fast. That is to say, the effect of integral on the computational time was sometimes a very slow solve, and sometimes inconsequential. Also, on the example of the last section, there is a drastic difference between the computational time it took to obtain F_{cs} when compared to the others. This of course depends to the workings of the solver, but it is related to the shape of the feasible set and the optimizing function; these optimizing functions are quite different for each problem, therefore one should not expect similar computational times for each problem.

8. Conclusion

This graduation project consisted on several different modeling phases of problems that are of interest in the context of synchromodal transportation problems in a tactical level. Firstly, an assessment was done on what problems could be solved via mathematical modeling, and the scope of the project was determined. This assessment consisted of literature study and, more importantly, contact with several parties interested on synchromodal transportation in both practical and theoretical terms, via meetings of stakeholders on the COMET-SP. Thanks to this constant involvement, the project could be formed into something that not only is of mathematical interest, but attempts to tackle some problems encountered on synchromodal transportation at the container-to-mode assignment level.

In order to get a clear picture of the mathematical models used in synchromodal transportation problems on a tactical-operational level, a framework was developed as part of a collaborative work on chapter 3. This framework helped by categorizing synchromodal problems via their characteristic properties, so that their approach and complexity can be identified.

Then, in order to tackle a container-to-mode assignment problem, an assessment was done in the modeling approaches of this kind of problems, emphasizing on their computational time. From this we concluded that if certain realistic assumptions are exploited, a path-based formulation of the minimum cost multi-commodity flow problem is computationally efficient. Therefore, this model was used for the main part of the project. Also, the features of owned transport and delay were added to this model.

This main part of the project consisted of an assessment of the important attributes considered when container-to-mode assignment is done, which is done in chapter 6. From this we constructed linear formulations of different attributes (robustness, flexibility and customer satisfaction) and incorporated them to the previously done model, considering the proper generalizations for the added features of owned transport and delay. With the results obtained from chapter 6, we developed a interactive multi-objective optimization tool, presented on chapter 7, which is meant to be used as a decision support tool for planners. This tool provides the user the possibility to explore solutions as she/he seems fit, and provides a range of different planning solutions for the planner to choose from, which are both properties sought for in a decision support tool. An example to illustrate the use of this tool is provided at the end of chapter 7.

With this study, we are able to address our research questions

- *How to consider different important performance indicators on a synchromodal transportation network?*

We propose a multi-objective optimization approach because the results we obtained are satisfying from different perspectives. In terms of what is important to transportation planners, it provides a range of solutions and interaction with the planner. It also conciliates the fact that the importance of the attributes is relative depending on many factors (including the instance and the planner). (Chapters 6 and 7)

- *What to optimize on a synchromodal transportation network on a container-to mode assignment?*

Both in literature and in practice many concepts are observed to be important on the performance of a synchromodal transportation network: Cost, robustness, flexibility, reliability, customer satisfaction and resilience all play a part on the performance of a synchromodal transportation network. One should be specific about the the meaning intended for each concept, since this is not always consistent, and many times context sensitive. On chapter 6 we propose a novel way to quantify attributes, on an attempt to formulate complex multi-objective problems of freight transport beyond the cost function. (chapter 2 and 6)

- *How to model container-to-mode assignment problems on a synchromodal transportation network in a time-efficient way?*

We propose a path-based minimum cost multi-commodity flow problem on a space-time network that represents the problem. (chapter 4 and 5)

- *How to categorize synchromodal transportation problems?*

We propose a framework for tactical-operational problems that distinguishes the nature of the elements of the problem considered, as well as the solution method. (chapter 3)

Appendix: Format of solution

Here we present a full transportation files of an instance like Problem 1 of 5.2, with only 10 orders to dispatch. To compensate with the little number of orders, capacity was reduces to 10 for all arcs (so that there is not a single path per commodity). As can be seen, first the value of the attributes and other relevant number are given, and then the flow per order is specified.

```
flowout - Notepad
File Edit Format View Help
Optimizing Cost
********************************
Status: Optimal
Total cost of transportation of flow = 1734.0
Total amount of containers = 151
Total amount of trucked containers = 28.0
Average value per container = 11.483443708609272
----------------------------------------------------
mean robustness = 0.9555983357658503
anti-flexibility = 106.0331883688386
linear anti-flexibility = 12.208566985966508
customer satisfaction = 0.46646964931347074

********************************

0. Origin: ('A', 5). Destination: ('E', 28)

[(('A', 24), ('E', 27))] = 5.0
[(('A', 13), ('E', 16))] = 10.0

-------------------------------
1. Origin: ('C', 7). Destination: ('A', 28)

[(('C', 11), ('E', 15)), (('E', 15), ('A', 19))] = 5.0
trucked = 8.0
[(('C', 23), ('A', 28))] = 10.0
[(('C', 15), ('D', 17)), (('D', 17), ('B', 21)), (('B', 21), ('A',
24))] = 2.0

-------------------------------
2. Origin: ('C', 9). Destination: ('E', 16)

[(('C', 11), ('E', 15))] = 5.0
[(('C', 9), ('D', 15)), (('D', 17), ('E', 22))] = 1.0

-------------------------------
3. Origin: ('B', 12). Destination: ('F', 27)
```

```
flowout - Notepad
File Edit Format View Help
-------------------------------
3. Origin: ('B', 12). Destination: ('F', 27)

[(('B', 13), ('C', 15)), (('C', 20), ('F', 24))] = 10.0
[(('B', 21), ('A', 24)), (('A', 28), ('F', 30))] = 5.0

-------------------------------
4. Origin: ('D', 0). Destination: ('E', 29)

[(('D', 0), ('A', 6)), (('A', 24), ('E', 27))] = 4.0
[(('D', 24), ('C', 26)), (('C', 27), ('E', 29))] = 10.0

-------------------------------
5. Origin: ('A', 12). Destination: ('B', 25)

[(('A', 15), ('B', 20))] = 10.0
trucked = 10.0

-------------------------------
6. Origin: ('D', 8). Destination: ('F', 20)

[(('D', 17), ('E', 22)), (('E', 27), ('F', 29))] = 9.0
[(('D', 15), ('F', 17))] = 10.0
[(('D', 9), ('B', 13)), (('B', 21), ('A', 24)), (('A', 28), ('F',
30))] = 3.0

-------------------------------
7. Origin: ('E', 0). Destination: ('B', 20)

[(('E', 6), ('F', 10)), (('F', 13), ('D', 17)), (('D', 17), ('B',
21))] = 8.0

-------------------------------
8. Origin: ('A', 8). Destination: ('C', 21)

[(('A', 11), ('F', 13)), (('F', 17), ('C', 20))] = 6.0
```

References

[1] Bai, R., Wallace, S.W., Li, J., Chong, A.Y.L. *Stochastic service network design with rerouting.* Transp. Res. Part B 60, 50–65, 2014.

[2] Behdani, Behzad, et al. *Multimodal schedule design for synchromodal freight transport systems.* Available at SSRN 2438851, 2014.

[3] Beuthe, M., Bouffioux, C.H., *Analysing qualitative attributes of freight transport from stated orders of preference experiment.* Journal of Transport Economics and Policy 42, 105–128. 2008.

[4] J. Branke, K. Deb, K. Miettinen, R. Slowinski. *Multiobjective optimization.* Springer, 2008.

[5] M. Conforti, G. CornuÃ©jols, G. Zambelli. *Integer Programming.* Springer, 2014.

[6] W. Cook, W. Cunningham, W. Pulleyblank, A. Schrijver. *Combinatorial Optimization*,Wiley, 1997.

[7] T.G. Crainic and G. Laporte. *Planning models for freight transportation.* European Journal of Operational Research, 97(3):409-438, 1997.

[8] T.G. Crainic and J. Roy. *Tools for tatical freight transport planning.* European Journal of Operational Research, 33(3): 290-297 1988.

[9] M. Fonoberova. *Algorithms for Finding Optimal Flows in Dynamic Networks.* Handbook of Power Systems II: 31-54, 2010.

[10] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G. Kan Rinnooy *Optimization and approximation in deterministic sequencing and scheduling: a survey.* Annals of discrete mathematics, Elsevier, 1979.

[11] J. Husdal, *A Conceptual Framework for Risk and Vulnerability in Virtual Enterprise Networks.* on Managing Risk in Virtual Enterprise Networks: Implementing Supply Chain Principles. Business Science Reference. 2010.

[12] Ishfaq, R., Sox, C.R. *Intermodal logistics: the interplay of financial, operational and service issues.* Transportation Research Part E: Logistics and Transportation Review 46(6), 926–949, 2010.

[13] D.G. Kendall. *Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain.* The Annals of Mathematical Statistics, JSTOR, 1953.

[14] K. Kooiman. *A Classification Framework for Time-stamp Stochastic Assignment Problems and an Application to Inland Container Shipping.* Master Thesis, Erasmus School of Economics, 2015.

[15] Le Li. *Coordinated Model Predictive Control of Synchromodal Freight Transport Systems.* TRAIL thesis series, 2016.

[16] X. Lin, R. Negenborn, G. Lodewijks. *Towards Quality-aware Control of Perishable Goods in Synchromodal Transport Networks.* IFAC-PapersOnLine, Elsevier, 2016.

[17] M. RK Mes, M.E. Iacob. *Synchromodal transport planning at a logistics service provider.* Logistics and Supply Chain Innovation, Springer, 2016.

[18] K. Miettinen, F. Ruiz, A.P. Wierzbicki. *Introduction to Multiobjective Optimization: Interactive Approaches.* Lecture Notes in Computer Science, 2008.

[19] E. Miller-Hooks, X. Zhang, R. Faturechi. *Measuring and maximizing resilience of freight transportation networks.* Elsvier, 2012.

[20] A. PÃ©rez. and M. Mes. *Service and Transfer Selection for Freights in a Synchromodal Network.* Computational Logistics: 227-242, 2016.

[21] F. Phillipson. *Multi Objective Approach for Tactical Capacity Management of Distributed Generation.* Innovations for Community Services: 17th International Conference I4CS, 2017.

[22] S. Pfoser, et. al. *Critical success factors of synchromodality: results from a case study and literature overview.* Elsevier, 2016.

[23] B. van Reissen, R. Negenborn, et al. *Synchromodal Container Transportation: An Overview of Current Topics and Research Opportunities.* Computational Logistics: 386-397, 2015.

[24] B. van Reissen, R. Negenborn, et al. *Service network design for an intermodal container network with flexible due dates/times and the possibility of using subcontracted transport.* Int. J. of Shipping and Transport Logistics, Springer, 2013.

[25] Steadie Seifi, M., et. al. *Multimodal Freight Transport Planning: A Literature Review.* European Journal of Operational Research 233.1: 1-15, 2014.

[26] C. Theys, W. Dullaert, T. Notteboom. *Analyzing cooperative networks in intermodal transportation: a game-theoretic approach.* Nectar Logistics and Freight Cluster Meeting, 2008

[27] M. Zhang, AJ Pel. *Synchromodal hinterland freight transport: model study for the port of Rotterdam.* Journal of Transport Geography, Elsevier, 2016.

[28] SynchroMania the game https://time.tno.nl/en/articles/synchromania-teaches-students-the-art-of-synchromodal-thinking/

[29] Network Optimization course https://ocw.mit.edu/courses/sloan-school-of-management/15-082j-network-optimization-fall-2010

[30] C. Caplice, F. Jauffred. *Balancing Robustness and Flexibility in Transportation Networks.* MIT Center for Transportation and Logistics: http://ctl.mit.edu/sites/ctl.mit.edu/files/caplice-SCB-MarApr2014.pdf

[31] Lean and Green Synchromodal http://www.synchromodaliteit.nl/case/lean-and-green-barge/

[32] Synchromodaily http://www.synchromodaliteit.nl/case/synchromodaily/

[33] Synchromodal Control Tower http://www.synchromodaliteit.nl/case/synchromodale-control-tower/
[34] Synchromodale Cool Port control http://www.synchromodaliteit.nl/case/synchromodale-cool-port-control/
[35] Rotterdam Tilburg Case www.synchromodaliteit.nl/download/een-stip-op-de-horizon/