Structural uncertainty in supply chain simulation models

An approach to account for structural uncertainty in supply chain simulation models

Bruno Hermans

Delft University of Technology

TUDelft

Structural uncertainty in supply chain simulation models

An approach to account for structural uncertainty in supply chain simulation models

by

Bruno Hermans

to obtain the degree of Master of Science in Engineering and Policy Analysis at the Delft University of Technology, to be defended publicly on August 22.

Supervisors:	Jan Kwakkel, Yilin Huang, Isabelle van Schilt
Institution:	Delft University of Technology
Place:	Faculty of Technology Policy and Management, Delft
Project Duration:	January 2022 - August 2022

Cover Image: https://pixabay.com/illustrations/background-structure-lines-3581188/

TU Delft

Preface

This thesis marks the end of a remarkable and enjoyable journey at TU Delft. Delft and TU Delft have always been special to me. Delft is the place where I was born and where I grew up. Ever since I was young, I cycled through the university campus, seeing it as a prestigious and reputable. When I entered high school, I started in lower general secondary education. Academic education was something that seemed unattainable to me. However, after hard work, I moved on to the highest level of secondary education in the Netherlands. Slowly, but surely, academia became a realistic option for me. After finishing high school, I was admissible to TU Delft. I was proud, and felt privileged that I was allowed to study here.

My parents have always helped me in this journey to TU Delft. Particularly during my time at high school, they have been very helpful to get the best out of myself. The opportunities to move to the highest level of secondary education were shaped by them. They have alarmed the teachers and me whenever they noticed that I was not getting the best out of myself. To this day, I am still very thankful to them for doing that.

The choice to study at TU Delft is one that was ultimately self-evident. However, without my grandfather, my interest in technology would never have developed to such an extent. Throughout my life, my grandfather has continuously inspired me with his machines, model aeroplanes, model boats, 3d printers, and smart tools he developed himself. The stargazer, steam engines, and model kits that he gave me ensured that I had a strong basis in technology even before I went to TU Delft. Thanks grandpa.

The past five years at TU Delft were full of joyful learning experiences, adventures and personal development. I have learned a wide variety of things, from programming, to transport, to economics. I have made lots of new friends, and gained confidence in my own abilities.

Writing a thesis is something that costs time and that can not be done without support. I want to thank Jan Kwakkel for his invaluable comments on my work. His comments greatly helped me to shape my work and also reflect on my work. I want to thank Yilin Huang, for her useful feedback. Her feedback challenged me to make my work better, and made working on my thesis a lot more fun. I want to thank Isabelle van Schilt for her feedback, and the discussions that we had about my work. Without the weekly meetings and discussions that we had, I would have never finished writing this thesis.

Other than that, I want to thank my family and friends who have been supportive during my thesis. I especially want to thank Floris. From the first day of our studies, we have been friends. We have been collaborating in every project possible. Floris supported me during my Bachelor and Master, and has always been very helpful for me. I am also very grateful to have been working with Bram, Wesley and Lieuwe, whom have motivated and helped me during my studies. Especially during the global pandemic. Last, my thanks go to Nienke who has been of great support.

Bruno Hermans
Delft, August 2022

Summary

Illicit supply chain networks are not well mapped. Regulators do not know how goods flow from supplier to retailer. There is uncertainty about whom is involved, where goods originate from, and what quantities are being shipped. Simulation models are effective tools to find measures against the distribution of illicit goods such as personal protective equipment. However, simulation models often solely handle uncertainty through the variation of parameters. Structural uncertainty, which is uncertainty in the structure of the model, is often neglected. This study focuses on accounting efficaciously for structural uncertainty in supply chain simulation models using model-driven exploratory modelling.

Model composability, which is a specific form of model driven exploratory modelling, is used in this study. The methodology is applied to a supply chain of illicit personal protective equipment. Using a model composer, many plausible models are generated of this supply chain. A model composer works by coupling model components in different configurations, while complying to preset constraints. Model components are submodels of a supply chain actors, for example, a retailer. Constraints help to restrict the way the model components can be coupled, making sure that every model generated by the model composer is plausible.

A ground truth is established to test the model composer on its efficacy to account for structural uncertainty. A ground truth is a simulation model of an illicit supply chain that functions as a benchmark. Five sets of 100 models are generated by the model composer to estimate the ground truth. Each set of models is generated with a different set of constraints. A constraint set consists of elements such as the maximum number of suppliers, the locations of supply chain actors, and the maximum number of customers of a supplier. These sets reflect different perspectives on an illicit supply chain.

Results show that structural uncertainty can result in significantly different simulation outcomes. The time in system, the production time, and the international transport time depend the most on changes in the constraints of the model composer. The time in system, the production time, and the international transport time of the models generated by the model composer are significantly different from the ground truth. The distributions of these outcomes have a different shape and have a wider range of possible values. Therefore, this study shows that model composability, a specific form of model-driven exploratory modelling, is efficacious in accounting for structural uncertainty in supply chain simulation models.

In the future, the methodology shown in this study can be used to model structural uncertainty in other fields such as water pipes networks, gas pipes networks, and telecom networks. Furthermore, the methodology can be used to identify robust measures to tackle the problem of illicit supply chains. Another recommendation is to use model composability for the individual components of the model. For example, a component such as a retailer can be build from several components: a cash register, a shelf, and a distribution area.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
DAG	Directed Acyclic Graph
DEVS	Discrete Event System Specification
DSOL	Distributed Simulation Object Library
PES	Pruned Entity System
PPE	Personal Protection Equipment
Pydsol	Python Distributed Simulation Object Library
SES	System Entity Structure

List of Figures

List of Tables

1

Introduction

This chapter introduces the core concepts and outlines the research approach used in this thesis. First, 1.1 introduces the problem of illicit supply chains. In this section the relevance of providing better insight in such supply chains is outlined. Second, in section 1.2 the role of structural uncertainty in supply chain simulation models is covered. Within this section the research gap, and the research goals are stated. Finally, the structure of this thesis is presented in section 1.3.

1.1. Problem formulation

Supply chains are the backbone of our global world economy. Every year, close to a billion containers are transported through ports worldwide (UNCTADstat, n.d.). Besides, about 2000 million tons of crude oil are transported overseas (Statista, 2021). The vast majority of these supply chains are legal, nonetheless some are illicit. The Organisation for Economic Co-operation and Development estimates that a total of 250 billion USD of counterfeit or pirated goods were traded in 2007 (OECD, 2009).

Illicit supply chains are groups of multiple organizations engaging in the distribution of goods, while participating in one or more illegitimate activities such as sourcing, procurement, production, logistics or distribution (Basu, 2013, 2014). The prevalence of such illicit supply chains has a number of negative social economic effects (Basu, 2014; Jabarzare et al., 2020). The illegitimate activities may have effects on public health, security, the economy, and government income. Products distributed in illicit supply chains range from drugs, illegal weapons, wildlife (for instance ivory), illicit tobacco and illegally produced goods (Basu, 2013). Regulators aim to disrupt such supply chains, because of the negative social economic effects the prevalence of these supply chains have. However, disrupting illicit supply chains proves to be a challenge, and depends on the product being distributed.

A literature study by Staake et al. (2009), revealed that few is known, both in practice and in theory, about the structure of illicit supply chains. Studies focus on the legal aspects (Jiao et al., 2021) of an illicit supply chain, or the transaction costs between supply chain actors (Basu, 2014). Others focus on how to interdict illicit supply chains, using network models (Jabarzare et al., 2020). To the authors best knowledge, simulation is not leveraged as a method to better map the structure of an illicit supply chain. Mapping the structure of an illicit supply chain is relevant, as it will ultimately help regulators and governments disrupt illicit supply chains.

Although an uncertain supply chain configuration is most relevant when it comes to illicit supply chains, legal supply chains sometimes also have an uncertain configuration. For example, in supply chains of cacao and coffee the supply chain structure is frequently undocumented. In these types of supply chains, companies participate without proper IT abilities, such as cacao or coffee farmers. Therefore, mapping the supply chain structure is relevant in both illicit and legal supply chains. This study aims to use simulation to provide better insights in the supply chain structure.

1.2. Scientific relevance

The use of simulation in the field of supply chains is not unique and new. Often, simulation models are used to boost a supply chains' efficiency. In many cases, simulation models are used to design a supply chain system to function as optimal as possible. Simulation models are used to determine the optimal configuration of assets such as production sites, distribution centres, and stores (Bittante et al., 2018; Fumarola et al., 2010; Gargalo et al., 2017).

Current supply chain simulation models fail to model a partly unknown supply chain configuration. The partly unknown supply chain configuration, causes uncertainty in the structure of the simulation model. This type of uncertainty, is known as structural uncertainty. In many supply chain simulation models, the structure of the simulation model is assumed to be fixed and known. In fact, most knowledge driven models are presented as reality, rather than an array of hypothesis of how the system might work or as the actors view the system (Keller & Hu, 2019).

Assuming a single model structure to be correct, when in fact it is not, can be problematic. This can be problematic, because researchers and companies come to a different conclusion when they study the same system with a different model. Li et al. (2017), Refsgaard et al. (2006), and Vautard et al. (2013) show that researchers construct different models, even when the question they try to answer, and the data they use, is the same. For example, Li et al. (2017) compare 37 published models on the spread of Ebola. Leveraging the ensemble of models, they try to deduce which measure is most effective against the spread of Ebola. They found that depending on which model structure the model had, the recommended measure was different. Besides that, Refsgaard et al. (2006) provide another example in the context of pollution. They compare five models of aquifer vulnerability towards nitrogen pollution. The models were used to identify specific vulnerable areas. The estimated vulnerability differed per model, despite the fact that the same spatial resolution and observational data were used. Furthermore, Vautard et al. (2013) show that regional climate models have a broad prediction range when it comes to simulating heat waves. The models differ in their prediction of temperature and precipitation.

There are many ways to account for structural uncertainty in (simulation) models. There are three types of accounting for structural uncertainty: the compensating strategy, the expert judgement strategy, and exploratory modelling.

Compensating strategy: If the compensating strategy is used, the error caused by structural uncertainty is compensated by for example a statistical Gaussian process (Brynjarsdóttir & O'hagan, 2014). In this way, the prediction error is compensated by means of an uncertainty bandwidth.

Expert judgement strategy: Another strategy to account for structural uncertainty is the expert judgement strategy (Winsberg, 2010). Webster et al. (1998) show that expert judgement can form the basis of parametrizing structural uncertainties. In their study, the structural uncertainties are first represented by parameters. This is possible, as the parameters can be used to change the model's characteristics. Afterwards, these uncertain parameters are estimated by experts. These subjective expert judgements are mapped into a probability density function, which is used to run the model. The model then generates a set of outcomes, to which a probability is linked.

Exploratory modelling: The idea of this approach is to create multiple possible model structures that are all plausible. Exploratory modelling is a modelling approach that is based upon the exploration of multiple different model structures (Bankes, 1993). Using this approach, an ensemble of models is generated to study a problem.

This thesis focuses on the exploratory modelling approach. The exploratory modelling strategy has several benefits over the other two strategies. The compensating strategy focuses on compensating for the error introduced by structural uncertainty. However, this strategy does not take into account that the distribution of the errors might have a shape that is not easily reproduced with a statistical distribution. The expert judgement strategy is based on opinions of experts. The exploratory modelling strategy does not require such opinions, within this strategy as many as possible models are generated. Therefore, this study attempts to account for structural uncertainty in supply chain models leveraging exploratory modelling.

Bankes (1993) defines three forms of exploratory modelling: data-driven, question-driven, and model driven exploratory modelling. The focus in this thesis is on model-driven exploratory modelling. Model-driven exploratory modelling is the studying of an ensemble of models without having a data set or a particular question up front. When studying supply chains with an unknown configuration, this type of modelling is relevant because empirical data is scarcely available. Besides, the goal is foremost to better map (illicit) supply chains. The analysis is thus not started with a particular policy question known up front. Therefore, data-driven exploratory modelling is not applied because it requires data sets, and question driven exploratory modelling is not suitable because it is not aimed at better mapping phenomena.

So far, relatively few studies have combined supply chains, transport, or logistics with exploratory modelling (Gruchmann et al., 2019). Besides, studies that did combine exploratory modelling with supply chains, transport, or logistics focused on parametric uncertainties rather than structural uncertainties (Halim et al., 2016; Moallemi & Köhler, 2019). Halim et al. (2016) use exploratory modelling in the context of the global container shipment system. In their study, nine uncertain parameters are used and varied. Model structural uncertainty was not examined. Moallemi and Köhler (2019) use exploratory modelling in the context of mobility. However, they solely use model parameters to represent uncertainty and structural uncertainty is not modelled.

Concluding, few studies pay attention to structural uncertainty when studying supply chains. Besides, few studies account for structural uncertainty using exploratory modelling in the domain of supply chains. Therefore, this study aims to efficaciously account for structural uncertainty leveraging exploratory modelling. To achieve this, a number of sub-goals are defined. The first sub-goal is to formulate a clear conceptualization of structural uncertainty in simulation models. The second sub-goal is to formulate a specific exploratory modelling method that accounts for structural uncertainty. The third and final sub-goal of this research is to apply and test the method.

To achieve these objectives formulated, the following research question is formulated:

How to efficaciously account for structural uncertainty in supply chain simulation models using model-driven exploratory modelling given limited data?

This main question is split into multiple sub-questions:

1. *How to conceptualize structural uncertainty in the context of supply chain simulation models?*
2. *What model-driven exploratory modelling approaches are used to account for structural uncertainty in simulation models used in other fields?*
3. *How can model-driven exploratory modelling be used to account for structural uncertainty in the context of supply chain simulation models?*
4. *How does a change in perception of the target system change the method's efficacy?*

An approach to each of the sub-questions is explained in the next paragraphs.

1. How to conceptualize structural uncertainty in the context of supply chain simulation models?
To answer this sub-question, a clear conceptualization of structural uncertainty is made in chapter 2. By answering this sub-question, terminology for the rest of this thesis is laid out. There is special attention for the difference between structural and parametric uncertainty. Furthermore, this chapter focusses on what constitutes a valid (simulation) model. This will be approached from several philosophical perspectives to formulate a well-considered, practical definition.

2. What model-driven exploratory modelling approaches are used to account for structural uncertainty in simulation models used in other fields?
There are many approaches to account for structural uncertainty. However, the focus in this thesis is on model composability. Model composability is one form of model-driven exploratory modelling. Departing from the theoretical fundaments of simulations, chapter 3 highlights how model composability can be used to model structural uncertainty. In this chapter, concepts such as Discrete Event System Specification (DEVS), coupled DEVS models, the System Entity Structure (SES) and their relation are explained.

3. How can model-driven exploratory modelling be used to account for structural uncertainty in the context of supply chain simulation models?
This sub-question is answered in chapter 4. Within this chapter, the theoretical concepts of chapter 2 and 3 are applied to a real world case. This shows how the theoretical concepts can be applied to a real world illicit supply chain. A composable discrete event simulation model is developed to demonstrate its abilities to account for structural uncertainty. The implementation of this composable discrete event simulation model is explained, and an introduction to how it is used will also be provided.

4. How does a change in perception of the target system change the method's efficacy?
The final sub-question of this thesis focuses on the efficacy of the composable model to account for structural uncertainty. The answer to this question is laid out in chapter 6. The focus of this chapter is to discover which assumptions of the model structure are of significance to the simulation outcomes. To do this, the composable model is tested with different sets of assumptions.

1.3. Structure of this thesis

The structure of this thesis is shown in figure 1.1. This chapter (chapter 1) introduced the topic of thesis, and the research questions. Chapter 2 is aimed at clarifying what structural uncertainty exactly is. In chapter 2, structural uncertainty is conceptualized, and distinguished from other types of uncertainties. Leveraging these definitions, chapter 3 is aimed at explaining how structural uncertainties can be modelled using exploratory modelling. Within this chapter, it is explained how model composability can be used to model structural uncertainty in supply chain simulation models. Chapter 4 focusses on demonstrating the techniques explained in chapter 3. A practical case study of an illicit supply chain is conducted, to demonstrate the method. A so-called model composer is developed to model structural uncertainty in supply chain simulation models. In chapter 5, an experimental setup is discussed using this model composer. The experiments are aimed at testing the model composer's efficacy in modelling structural uncertainty. Chapter 6 presents the results of the experiments. Chapter 7, contains a discussion. Finally, chapter 8 contains a conclusion of this research.

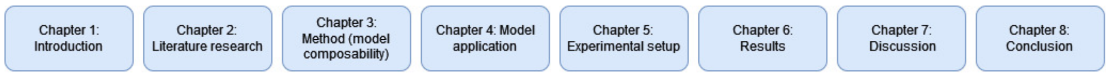

Figure 1.1: Structure of thesis

1.4. Link to EPA programme

This thesis is related to the EPA program. First of all, the thesis is not only about structural uncertainty in supply chain models. The approaches used to account for structural uncertainty are generalizable, and could be used within other domains. For example, the developed method is also relevant to climate models. Similar to illicit supply chains, climate models are prone to structural uncertainty because both types of models are hard to validate using empirical data. Secondly, this thesis recognizes the relevance of the involvement of many actors' perspectives. Within this thesis, we aim to include a diversity of perspectives, rather than presenting a single solution as most optimal.

2

Conceptualizing structural uncertainty in supply chain simulation models

This chapter is about what structural uncertainty is, how it differs from other types of uncertainty, and why it arises when modelling. Section 2.1 includes a discussion on when models are considered to be true, useful or valid. Section 2.2 sheds light on what structural uncertainty is, and where it originates from. Subsequently, section 2.3 is aimed at distinguishing structural uncertainty from other types of uncertainty. Afterwards, section 2.4 applies the concept of structural uncertainty in supply chain simulation models. Finally, section 2.5 provides an overview of this chapter.

2.1. What is a valid simulation model?

Models are used for many purposes, and whether a model is valid depends upon where the model is used for. Models are used to describe the world, to predict the weather, to navigate ships, to recommend content on Youtube, and to give insight in the worlds future climate (Page, 2018). Models are powerful, but are simplifications of reality; usually they are not a replication of the system under study. Consider the example of maps. To have a perfectly detailed map, the world would essentially have to be replicated. For example, a map showing highway routes in France does not contain details like water pipes or water depth. Since perfect replication is impossible, geographers have made maps to describe the world. However, not all maps are suited for every purpose. Some maps, like the Mercator map (the projection system used by google maps), are perfectly suitable to navigate. However, the Mercator map should not be used to determine the size of a country or to calculate the distance between two points. If one were to measure the distance between two points using the Mercator projection, the distance measured would be wrong. Instead, one should use an equal area projection to determine the size of a county and an equidistant projection for the distance between two points. Geographers have described which map should be used given a certain purpose. Yet, this is not always the case. In many fields it is unclear what model should be applied in a particular case. Consider the example of modelling the spread of Ebola. Li et al. (2017) demonstrate that 37 models recommend different measures against the spread of Ebola. In such a situation, there is uncertainty about the structure of the model. It is unclear which model is valid and all the models have a different structure.

The above examples illustrate why the purpose and the validity of a model are of relevance when it comes to structural uncertainty. Before continuing, a definition of a model is provided to make sure that it is clear what is meant by a model. In the rest of this thesis, the following definition of a model is used:

”Any system A is a model of a system B if the study of A is useful for the understanding of B without regard to any direct or indirect causal connection between A and B. A must be like B in some respects. The resemblance is in terms of the pattern or order exhibited in each system.” (Kaplan, 1964, pp. 258-291)

When validating a simulation model, the modeller argues that there is a strong relation between system A and B. Often, system B is a real world system, but it can also be a simple model of a complex model that we aim to understand. For example, a flowchart of a complex simulation model is a model of a model. In the rest of this text, system B is seen as a real world system.

There are different approaches when it comes to modelling. This thesis does not focus upon every approach and type of modelling, therefore it is useful to further specify which approach of modelling in this thesis is used. In general, there are three approaches towards modelling (Page, 2018). One of them is the embodiment approach, which tries to capture as much as possible from reality (for instance, a map). A second approach is the analogy approach, in which an analogy is used to explain a certain phenomenon (for example, using a bird to understand a plane). The final approach is the alternative reality approach. Examples of such models are the game of life, or models that try to help us understand the implications of unrealistic scenarios such as flying energy. In this thesis, the focus is on models that take the embodiment approach. The embodiment approach is taken, because it is most useful in modelling structural uncertainty. Moreover, this thesis focuses on supply chain simulation models, which typically are models developed using the embodiment approach.

In simulation modelling, developing such a model usually starts by formulating a problem, followed by conceptualization of the system (Dam et al., 2013). Afterwards, the conceptual model is formalized, and the formalized model is used to experiment. Finally, the model is validated and verified using several tools. In this validation and verification process, the model is tweaked untill the model has become sufficiently credible or representative. If this is the case, a model is deemed valid by the modeller.

It is precisely this validation process that is of interest when it comes to structural uncertainty. Depending on which philosophical world-view is adopted, the meaning of validation changes (Andreas Tolk, 2013; Barlas, 1996). If a strict positivist approach is taken, a model is considered valid when the model output corresponds to the empirical real world value. If a more constructivist approach is taken, a model is considered valid when it is fit for purpose or credible.

An uncertain structure of a supply chain simulation model usually means that there is a lack of data. When a simulation model is constructed in a data sparse environment, the positivist approach to validity seems rather obsolete (Winsberg, 2010). This is because there is no empirical data to which a simulation model can be compared. A more constructivist approach to validity seems more useful in simulation modelling in data sparse environments.

In sparse data environments, a model is thus valid, when it is fit for purpose, or it is credible to the user. This raises the question of what credibility really is. According to Winsberg (2010), a simulation model is credible whenever the results it produces fit well into the web of previously accepted data, our observations, our pen and paper analysis, and whenever their predictions are successful. It is thus a matter of our (shared) perception whether a simulation model is valid or not.

Such a notion of validation has several implications. From the perspective of positivism, a model is seen as an absolute representation of reality, and therefore models are either true or false (Barlas & Carpenter, 1990). This implies that for each phenomenon there is only one ideal and valid model. However, from the perspective of constructivism, a valid model is seen as only one of the possible ways of describing reality. Not a single model can claim absolute objectivity, because every model is based on the modellers' world-view. Not a single model is better than another, however one can prove to be more effective. In constructivism, models lie on a continuum of usefulness, and are not true or false.

Pluralism is a world-view that is somewhere in between constructivism and positivism. Pluralism is a world view in which there can be more than one model of a phenomenon. Parker (2006) discusses two types of pluralism: competitive pluralism and compatible pluralism. In competitive pluralism, two models of the same phenomena can coexist, even when their assumptions of the world are conflicting. However, in competitive pluralism, the coexistence is only temporarily. The existence of multiple models is there to ultimately select one best representation of reality. The models solely exist to stimulate debate. In compatible pluralism, the models have compatible underlying assumptions. In compatible pluralism, models can coexist permanently. The model's assumptions can be true at the same time.

Mitchell (2009) presents integrative pluralism as an alternative to compatible and competitive pluralism. Integrative pluralism maintains the belief that there are multiple correct ways to parse reality into models and theories. However, in integrative pluralism models can be compatible even when they have conflicting assumptions. This is because all descriptions are always partial. Descriptions are always partial, because there exists no such model that completely captures all relevant aspects of the world. A linguistic or a mathematical representation of the world is never complete and always leaves room for ambiguity. The partiality of representations forces us to simplify processes at higher levels that in fact should not be simplified to fully understand the phenomena. For example, Mitchell (2009) provides an example of models trying to predict depression in humans. She demonstrates that depression often is the result of a combination of causes at different levels. Both a person's trauma and a lack of serotonin seem to play a role in developing a depression, however the models that we have are unable to combine these causes in a single representation.

Considering the partiality of representations, the possibility that two incompatible models are both correct at the same time should not be neglected. In integrative pluralism, there ultimately is a single true explanation of a particular natural phenomenon. However, representations of the world might seem incompatible, because the level of abstraction of our representations is determined pragmatically by a combination of our cognitive ability and the scientific objective.

In the rest of this work, an integrative pluralist view of validation is adopted. Like mentioned before, this means that there are many descriptions of reality and that there is no such thing as the most accurate model. Because of the partiality of representations, models can coexist while having conflicting assumptions, but at the same time, there is just one true explanation for a particular phenomenon. The integrative pluralist view of validation is adopted, because of pragmatic concerns. The positivist approach is not suitable in the context of supply chains with an uncertain supply chain configuration, because empirical data is scarcely available. On the contrary, the constructivism provides relatively little guidance to validity. Since constructivism is based on credibility, there is a risk of accepting any model as valid. The integrative pluralist approach is favoured over the other two types of pluralism, because both competitive and compatible pluralism exclude the possibility that models with conflicting assumptions might coexist permanently.

2.2. A conceptualization of structural uncertainty

The notion of validity is closely connected to conceptualize structural uncertainty. In this section, a conceptualization of structural uncertainty from the perspective of integrative pluralism is presented. To explain what structural uncertainty is, we will first briefly touch upon the concept of a modelling relation, as defined by Rosen (1991). A modelling relation exists between a target system (natural system) and a model (formal system) representing that target system. The target system consists of physical matter, and the model is a simulation model or a different mathematical representation of the target system. The relation between the model, and the target system is called the modelling relation. In integrative pluralism, there is not just a single model that faithfully represents the target system. There even exists a plethora of models that each represent the target system. Therefore, it is possible to establish multiple modelling relations, as seen in figure 2.1. In many cases, it is unsure which model should be used given a certain purpose. Such a lack of sureness, which implies a lack of knowledge, can be seen as a synonym of uncertainty (Walker et al., 2013). Structural uncertainty relates to the uncertainty about how well a model represents its target system (Baldissera Pacchetti, 2021; Winsberg, 2010). Often, it is defined as the uncertainty in the structure of the mathematical model.

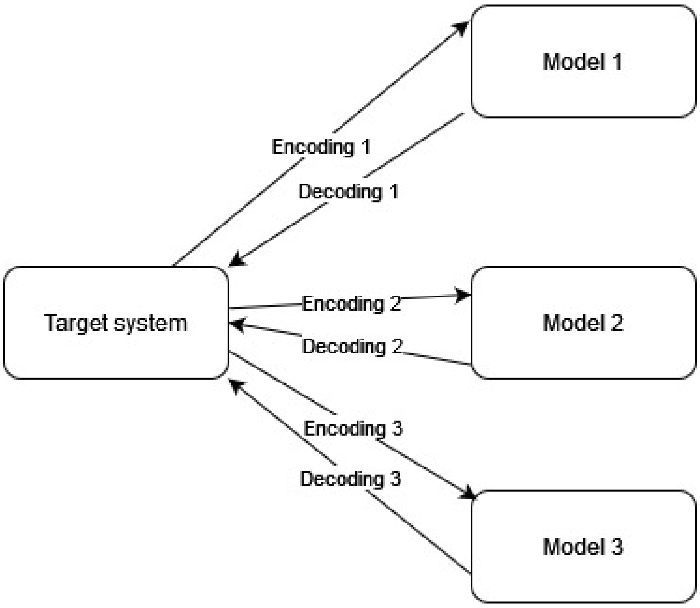

Figure 2.1: Structural uncertainty visualized, based on the modelling relation of Rosen (1991)

Philosophers have defined many explanations for the existence of structural uncertainty (Baldissera Pacchetti, 2021). Parker (2006) mentions that structural uncertainty occurs due to inability of comparing the model's output to empirical data. For example, in climate modelling, projections of 50 years into the future cannot be compared to empirical data. In such a case, it is impossible to validate the model with observational data, because of the inexistence of future data and the inaccuracy of historical observational data. Winsberg (2010) argues that structural uncertainty originates from value-driven priorities to certain prediction tasks. For example, he demonstrates that economic or political reasons might drive scientists to prioritize certain predictions tasks in coupled climate models. For instance, he shows how climate models estimate temperature projections to be less uncertain compared to precipitation projections. Frigg et al. (2014) argue that slight differences in the model structure of a non-linear target system might cause the model to inaccurately represent its target. Pacchetti (2018) argues that structural uncertainty occurs because modellers are unable to unequivocally distinguish the target system. She argues that modellers make implicit scale separation assumptions, causing structural uncertainty. Many models assume a phenomenon to happen at a certain scale. However, she argues that phenomena occur at different scales. For example, climate change is an interaction of molecular processes as well as processes that happen at a world scale like El Niño. Because scientists ignore some processes at different scales, due to the partiality of representations, their models are prone to structural uncertainty.

In literature, structural uncertainty is mentioned under many names, such as model discrepancy, model inadequacy, model error, model form error, model structure uncertainty, conceptual uncertainty and model bias (Brynjarsdóttir & O'hagan, 2014; Refsgaard et al., 2006; Walker et al., 2013; Webster et al., 1998). In the rest of this research, the term structural uncertainty will be used, by which the same concept is meant.

2.3. Structural uncertainty and other types of uncertainty

This section focuses on distinguishing structural uncertainty from other types of uncertainty. First, structural uncertainty is distinguished from parametric uncertainty. Afterwards, structural uncertainty is distinguished from contextual uncertainty. Finally, it is distinguished from methodological uncertainty.

Both structural uncertainty and parametric uncertainty are partially epistemic in nature (Kiureghian & Ditlevsen, 2009). Epistemic uncertainties can be resolved by gathering more data or refining the model. This means that epistemic uncertainties can eventually be mitigated using extra knowledge and resources. By contrast, aleatory uncertainties are uncertainties caused by the inherent randomness of a natural phenomenon. Even though both types of uncertainty are epistemic in nature, in classifications of uncertainty, structural uncertainty is often distinguished from parametric uncertainty (Bojke et al., 2009; Jackson et al., 2011; Webster et al., 1998; Winsberg, 2010). Whenever there solely is parametric uncertainty and no structural uncertainty, the ideal model structure is known, but its parameter values are uncertain. When there is structural uncertainty, the mathematical structure of the model is unsure. However, a distinction between parametric and structural uncertainty should be made carefully.

Only if a model is an adequate representation of reality, reviewing parametric uncertainty makes sense (Pilkey & Pilkey-Jarvis, 2007). However, if a model is not an adequate representation of reality, accounting for parametric uncertainty is meaningless. The reason for this is that the parameters of an invalid model are different from the parameters in the target system. In this sense, both parameters and models are means to depict certain phenomena. However, if a model is an inadequate representation of reality, its parameters values are also likely wrong and meaningless. And therefore, parametric and structural uncertainty are connected.

Structural and parametric uncertainty do not manifest themselves independently, because they can affect each other's magnitude (Van Zelm & Huijbregts, 2013). An increase in model complexity can increase parametric uncertainty and reduce structural uncertainty, and vice versa. Although the way structural uncertainty and parameter uncertainty affect each other might differ from study to study, figure 2.2 shows the tradeoff identified in the study of Van Zelm and Huijbregts (2013).

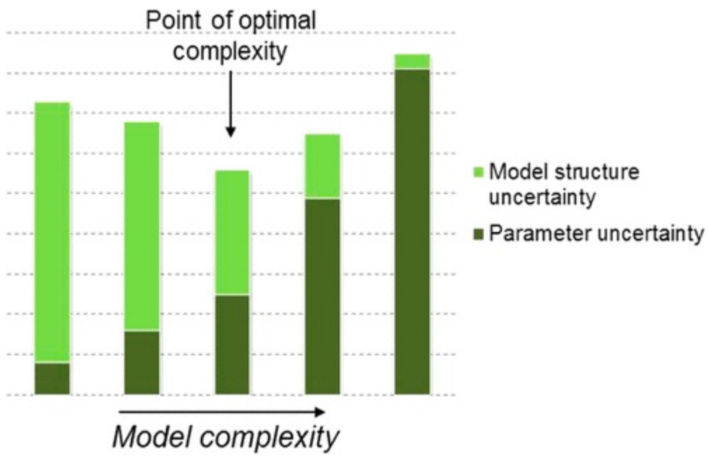

Figure 2.2: Tradeoff between parameter and structural uncertainty, image from Van Zelm and Huijbregts (2013)

Hyperparameters make distinguishing between structural uncertainty and parametric uncertainty even harder. In machine learning, hyperparameters are parameters that "guide" the model in learning its optimal form (Probst et al., 2019). Hyperparameters are used to construct are particular subset of models or are used to select an optimal model. Suppose that two simple equations 2.1 and 2.2 could be used to represent an arbitrary target system. Both could be valid representations, because the actual behaviour of the target system is unknown.

$$f(x) = x^2 \tag{2.1}$$

$$f(x) = x - 1 \tag{2.2}$$

In such a case, a single model structure can be created by using an extra parameter that "switches" between the two model structures. Combining the example models 2.1 and 2.2, results in model 2.3.

$$f(x) = z * (x^2) + (1 - z) * (x - 1)\ with\ z \in \{0, 1\} \tag{2.3}$$

Using hyperparameter z, the model enables structure 2.1 or 2.2. In this thesis, hyperparameter z is seen as a parameter helps to grasp some structural uncertainty. Parameters x and y are seen as parametric uncertainties, because they can vary within a range without changing the model structure.

In the rest of this research, structural uncertainty and parametric uncertainty are considered to be partially overlapping terms. Because of the dependence between parametric and structural uncertainty, parametric uncertainties should be treated carefully. In this study, parametric uncertainties are seen as uncertainties that are represented by a single variable. By this, we mean variables that are not used to enable or disable certain parts of the model. Parameters that are used to enable or disable certain parts of the model are called hyperparameters. Examples of parametric uncertainties in the context of supply chain models are changes in price, demand, costs, delays in production, and willingness to return a product (Govindan et al., 2015).

Structural uncertainty can be distinguished from context uncertainty (Walker et al., 2010). Context uncertainty refers to the boundaries of the target system, or the formulation of the problem. Whenever there is context uncertainty, the target systems' boundary is unclear, and multiple formulations of the problem are thus possible. This is visualized in figure 2.3. When solely facing structural uncertainty, the system boundary of the target system is thus fixed and unambiguous. Warmink et al. (2010) highlights the importance of making a distinction between structural uncertainty and contextual uncertainty explicitly, because many papers do not properly distinguish between these two types of uncertainties.

This distinction is of high relevance, as this study considers only structural uncertainty and no contextual uncertainty. This means that the system boundaries of the target system are not subject to discussion. The reason that this study excludes contextual uncertainty, is because it is out of scope.

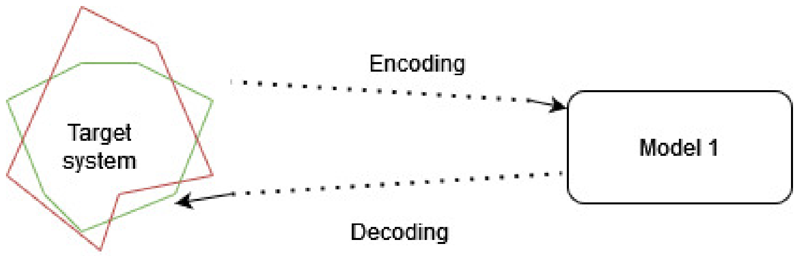

Figure 2.3: Context uncertainty visualized, based on Walker et al. (2010)

Furthermore, structural uncertainty is distinguished from methodological uncertainty (Jackson et al., 2011). Methodological uncertainty arises when it is unsure which type of (simulation) model, is most appropriate to represent the system of interest. For instance, a system might be modelled by using a system dynamics model, an agent based model or a discrete event simulation model. All of which have their own strengths and downsides. In this research, structural uncertainty is seen disjoint from methodological uncertainty. The choice of a specific modelling method is thus not seen as part of structural uncertainty.

2.4. Location and sources of structural uncertainty

Besides distinguishing structural uncertainty from other types of uncertainty, it is helpful to locate structural uncertainty in the model. A useful framework to describe the location of structural uncertainty is the XLRM framework of Lempert et al. (2003). The framework is depicted in figure 2.4. "L" refers to decision levers, which are variables that the decision maker can control. "X" stands for the externalities in the model, and represents all parametric uncertainties in the model. "M" refers to metrics, these are observables that are used to determine the effects of a particular strategy. "R" stands for relationships and refers to the relations between L and X within the model. Structural uncertainty mostly originates from uncertainty about the relations (R) in the model (Walker et al., 2013).

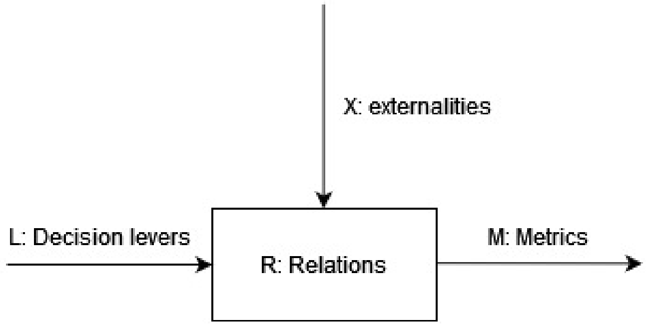

Figure 2.4: XLRM framework, based on Lempert et al. (2003) and Walker et al. (2013)

To explain relevant sources of structural uncertainty in supply chains simulation models, we first define a supply chain. A supply chain is an integrated system, in which raw materials are transferred into finished products and in which the goods are distributed amongst suppliers, manufacturers, distributors, retailers and costumers (Min & Zhou, 2002).

Structural uncertainty is uncommonly examined in the context of supply chains (Farrokh et al., 2018). More commonly, it is studied in the context of climate modelling or hydrological modelling (Gupta & Govindaraju, 2019; Webster et al., 1998; Winsberg, 2010). Literature on approaches to account for structural uncertainty in the context of supply chains is limited. However, two studies have identified two important sources of structural uncertainty in supply chains. The first one relates to uncertainty about the physical supply chain structure (Vilko et al., 2014). Secondly, structural uncertainty can originate from unexpected or unpredictable events (Farrokh et al., 2018).

The physical structure is the way distributors, retailers and other relevant actors are arranged, and is sometimes named the supply chain configuration. A supply chain configuration consists of the following aspects (Cigolini et al., 2014):

1. The number of nodes (locations) in the supply chain.
2. The amount of storage at each location.
3. The distance between the locations.
4. The number of levels (echelons) in the supply chain.

In literature, the supply chain configuration is rarely treated as an uncertainty. Frequently, it is seen as either given, or as a configuration to choose from. Previous studies focus on the relation between the configuration and the performance or the optimal configuration given a certain key performance indicator.

Previous work has demonstrated that the supply chain configuration has an effect on the supply chains' performance. Tombido et al. (2020) show that the supply chain configuration affects the size of the bull whip effect. The bull whip effect is caused by variability in demand for final products. This causes an upstream distortion of the supply chain. Furthermore, Cigolini et al. (2014) show that the supply chain configuration affects the overall supply chains' performance. Furthermore, Cardoso et al. (2015) show that the supply chains' configuration influences the supply chains' resilience. They show that some structures are more susceptible to disruptions than others.

Other studies demonstrate how to optimize the supply chain configuration to reduce cost or transport time (Bittante et al., 2018; Ding et al., 2004; Gargalo et al., 2017). These studies show that the supply chain configuration can be optimized given one or more key performance indicators.

Although many studies show the effect of having a different supply chain configuration or show how to optimize the configuration, none of them treat the supply chain configuration as a true uncertainty.

Besides structural uncertainty originating from the configuration, structural uncertainty can originate from unexpected or unpredictable events (Farrokh et al., 2018). These are events that have an influence on the supply chain, for example the introduction of a new competitor or a crisis.

2.5. Definition of structural uncertainty

This chapter discussed the main concepts surrounding structural uncertainty. It started by stating that models in this study are created using an embodiment approach. When it comes to validation, this study takes the view of integrative pluralism. This means that multiple models of the same target system may exist, even when they have conflicting assumptions. In the rest of this thesis, structural uncertainty is considered to be the uncertainty in the structure of a simulation model. Structural uncertainty is distinguished from contextual, parametric, and methodological uncertainties. In the context of supply chain models, the supply chain configuration is considered as the main source of structural uncertainty.

3

Modelling with structural uncertainty

In this chapter exploratory modelling and its relation to structural uncertainty is explained. The structure of this chapter is as follows. Section 3.1 covers the types of exploratory modelling. Section 3.2 covers a particular type of exploratory modelling: model-driven exploratory modelling. A particular type of model-driven exploratory modelling is model composability. In section 3.3 the fundamental concepts of model composability are explained. The focus is on explaining Discrete Event System Specification (DEVS) and the System Entity Structure (SES). In section 3.4 the relation between the two concepts is explained. Afterwards, section 3.5 explains the simulation framework used throughout the rest of this thesis. Finally, section 7.3 provides some limitations of model composability.

3.1. Exploratory modelling

Exploratory modelling is a modelling approach that is based upon the exploration of multiple different model structures (Bankes, 1993; Bankes, 2011). The creation of this family of models can either be done by sampling over the input space of a single model, or by generating several alternative model structures. This family of models is generated to study a problem. Exploratory modelling is applied in situations where there is significant uncertainty. In general, there are three types of exploratory modelling: data-driven, question-driven, and model-driven exploratory modelling. Each of them will be explained in the following alineas.

In data-driven exploratory modelling, a set of models is fitted to a dataset. Fitting a regression model to a dataset, is an example of this type of exploratory approach. Another example of data-driven exploratory modelling is the work of Keller and Hu (2019). They use a genetic algorithm to generate a set of models in a crowd management context. The genetic algorithm is used to fit several model structures on a dataset and requires big volumes of data. In the end, one particular model structure that best predicts the dataset is the result of these types of analysis.

Another type of exploratory modelling is question-driven exploratory modelling. In this type of modelling, a family of models is used to answer a particular policy question. An example of such a study is the study Li et al. (2017). In their study, a family of models is used to determine which measure is the most effective against the spread of Ebola. Using this type of exploratory modelling, a fit with a data set is not necessarily needed.

The final type of exploratory modelling is model-driven exploratory modelling. This type of exploratory modelling starts without a dataset or a particular type of policy question. In such an approach, a family of models is generated by coupling several sub-models or components in different manners. The general idea is to generate a set of alternative model structures, without fitting it to a dataset or without a particular question at first.

Regardless of what type of exploratory modelling is applied, the models will always result in a broad range of predictions. To make these predictions more precise, model averaging or selection can be used to reduce the range of predictions (Grainger et al., 2018). Both of these techniques start by using each model to compute the outcomes of interest. When applying model averaging, the average for each variable of interest is computed based upon every model. If model selection is applied, the model which represent the target system most accurately is selected.

All types of exploratory modelling are valid approaches to account for structural uncertainty. However, in this thesis, the emphasis will be on the model-driven exploratory modelling approach. The model-driven approach is favoured over other types of exploratory modelling, because the model-driven approach does not depart from empirical dataset (Bankes, 1993). This is an advantage, as supply chain data is usually sparsely available. Besides, this form of exploratory modelling is not applied often in literature, as discussed in chapter 1.

3.2. Model-driven exploratory modelling

Bankes (1993) provides a somewhat broad definition when it comes to model-driven exploratory modelling. He defines it as a form of experimental mathematics, in which a family of models is explored without a dataset or question known up front. Model-driven exploratory modelling encompasses many approaches. For instance, multi-resolution modelling is mentioned as one type of model-driven exploratory modelling. Multi-resolution modelling is the modelling of a particular phenomenon on several levels of scale. Another common approach is to use the exploratory modelling and analysis workbench from Kwakkel (2017). The workbench consists of many useful techniques, such as sampling techniques and optimization algorithms. However, the techniques are limited to varying the parameters of a single model structure. When desiring to evaluate structural uncertainties, or alternative model structures, the alternative model structures should be transformed to exogenous parameters. Currently, the workbench does not contain any techniques to create alternative model structures. The final model-driven exploratory modelling approach is model composability (Rodriguez & Yilmaz, 2020). Model composability is the idea of coupling several sub models in different combinations.

Model composability is the only type of model-driven exploratory modelling techniques that is capable of creating multiple alternative model structures. Despite the fact that model composability is capable of creating alternative model structures, so far it is rarely applied in the context of structural uncertainty. Other than Rodriguez and Yilmaz (2020), who applied model composability to account for structural uncertainty in modelling plume containment, model composability has not been applied to model structural uncertainty.

3.3. Model composability

Model composability is a type of model-driven exploratory modelling that can be used to vary the structure of a simulation model (Davis & Anderson, 2003). The structure of the model can be varied by the creation of various components (sub-models) that can coupled in various ways. Each component has inputs and outputs that can be connected to inputs and outputs of other components. Through different ways of coupling these components, multiple models can be created.

To create a valid family of coupled component systems, a SES, a set of constraints, a simulator and a model composer are needed (Yilmaz, 2019). Figure 3.1 provides a visual overview of these elements. The model composer is paramount in creating a family of models. The model composer couples all model components, while the coupling complies to the constraints. Each model is created in such a way that it is able to interact with the simulator. A SES is needed to describe all model components. Within the context of supply chains, components such as retailers, manufacturer and supplier are relevant. They can be described by using the SES or any equivalent ontology. The constraints contain information on how the elements in the SES might be pruned, and how they might be coupled. To simulate a model, a model also needs parameters. These are needed to control the behaviour of the control the behaviour of the model components.

Like stated before, the model composer is paramount to the model creation process. To illustrate how the model composer works, the DEVS formalism is used to illustrate that several simulation components can be coupled to form a higher level structure. There are several forms of the DEVS formalism. The most basic version of the DEVS version is introduced in section 3.3.1. This version of DEVS does not support the coupling of individual model components, but it is the basis of more sophisticated versions of DEVS. After the basic DEVS formalism is introduced, the DEVS formalism is expanded to the parallel DEVS formalism in section 3.3.2. Parallel DEVS contains the possibility to couple the model structure to another model structure. However, it is not yet a full specification of a coupled model. Therefore, the parallel DEVS coupled models are explained in section 3.3.3. Subsequently, section 3.3.4 introduces the SES. Afterwards, the simulation framework used in this thesis is discussed in section 3.5. Thereupon, this chapter concludes with section 3.4 in which the relation between DEVS and SES is covered. Finally, some limitations of the method are introduced in section 7.3

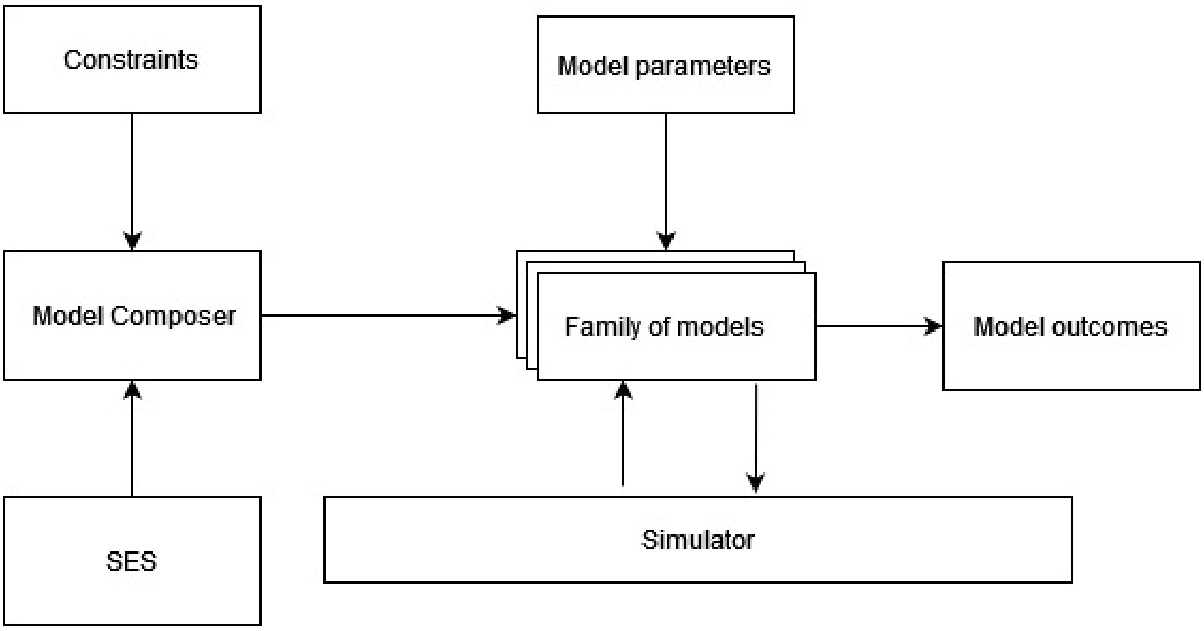

Figure 3.1: Using model composability to generate multiple simulation models. Figure based on (Yilmaz, 2019)

3.3.1. DEVS

To explain how model composability works, it is useful to discuss DEVS. DEVS is a formalism by Zeigler et al. (2019b) which can be utilized to specify a simulation model or a simulation model component. DEVS is used to explain how several model components can be coupled, forming the basis of model composability (Zeigler et al., 2019b). DEVS provides a strong computational basis for simulations and enables users to specify simulation models on paper. It can be utilized to specify any type of simulation, because other formalisms, such as system dynamics and agent-based models, can be embedded in the DEVS formalism (Vangheluwe, 2002). A basic DEVS structure can be expressed by structure 3.1.

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \tag{3.1}$$

with:

X refers to the set of all input events of the model.

S refers to the set of all internal states of the model.

Y refers to the set of all output events of the model.

δ_{int} is a function called internal transition function that is used to internally transform the state from the previous to the next one. It looks like $S \rightarrow S$.

δ_{ext} is a function called the external transition function. It is used whenever external of this particular DEVS model, for example the user triggers the model to do something else. It looks like: $Q * X \rightarrow S$ in which Q is $Q = \{(S, e) | S \in S, 0 \leq e \leq ta(s)\}$ in which e is the time elapsed since the last transition.

λ is the output function: $S \rightarrow Y$. This function maps the state to the output of the model.

ta is the time advance function: $S \rightarrow [\mathbb{R}^{+}_{0,+\infty}]$.

The DEVS formalism is visualized in figure 3.2 to clarify what the symbols in the DEVS formalism mean. The emphasis in this figure is on symbols X, Y and S. X denotes a set of input events, Y a set of output event and S denotes the internal state of the model. The other aspects of DEVS are equally important, but are less relevant when it comes to explaining the basic ideas of model composability.

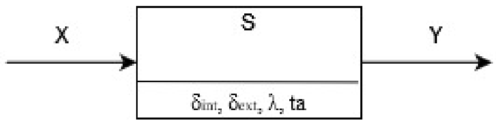

Figure 3.2: DEVS visualized

DEVS models can be coupled by the linking of an output of a DEVS model (from the set of outputs Y) to an input of another DEVS model (from the set of inputs X). This can be specified more elegantly by using the parallel DEVS formalism. The coupling itself can be specified by the coupled parallel DEVS formalism.

3.3.2. Parallel DEVS

Parallel DEVS is different from the basic DEVS formalism (Zeigler et al., 2019b). It is an extension of the basic DEVS formalism. The use of parallel DEVS has two benefits. First, it has a build in specification of what should happen when an internal an external event occur at the same time. Second, it has the notion of ports. The ports make it easy to couple specific outputs to specific inputs of another model. In parallel DEVS, a DEVS model is expressed by equation 3.2. This equation is similar to equation 3.1, but the sets X and Y are replaced by ports and the confluent transition function is added. No longer does the DEVS model only have a set of in- and outputs. The confluent transition function is added to specify what should happen if an external event and an internal event happen at the same moment.

$$M = \langle X_M^+, Y_M^+, S, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle \tag{3.2}$$

with:

$X = \{(p, v) | p \in InputPorts, v \in X_p\}$.
$Y = \{(p, v) | p \in OutputPorts, v \in Y_p\}$.
S is the set of states.
$\delta_{ext} = Q * X_M^+ \rightarrow S$.
$\delta_{int} = S \rightarrow S$.
δ_{conf} is the confluent transition function. This function is triggered whenever δ_{ext} and δ_{int} are triggered at the same moment. It looks like: $Q * X_M^+ \rightarrow S$ in which $Q = \{(S, e) | S \in S, 0 \leq e \leq ta(s)\}$. e refers to the time elapsed since the last transition.
$\lambda = S \rightarrow Y^+$ this is the output function. This function maps the state to the output of the model.
$ta = S \rightarrow [\mathbb{R}_{0,+\infty}^+]$ this is the time advance function.

3.3.3. Parallel DEVS Coupled models

Model composability is the idea of coupling several parallel DEVS models to each other. Parallel DEVS models can be coupled to form a parallel DEVS coupled model (Zeigler et al., 2019a). A parallel DEVS coupled model can be expressed by equation 3.3.

$$N = (X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,d}\}) \tag{3.3}$$

with:

X refers to the set of input events of the coupled model N.
Y refers to the set of output events of the coupled model N.
D is a set of names for each individual component.
$\{M_d\}$ refers to a set of parallel DEVS models.

$\{I_d\}$ is the influencer set of d with $I_d \subseteq D \cup \{N\}, d \notin I_d$. This set is used to specify which component influences another component.

$\{Z_{i,d}\}$ is a set that consists of the in- and output relations between model component i and d. In which $Z_{i,d}$ can be specified as follows:

$X \to X_d$, if $i = N$
$Y_i \to Y$, if $d = N$
$Y_i \to X_d$, if $d \neq N$ and $i \neq N$

A coupled parallel DEVS system is depicted in figure 3.3. The figure consists of three parallel DEVS components, which are coupled together. In this example, three different components were thus coupled in one way to form a single model (N). The interesting thing, is that the parallel DEVS models could theoretically be coupled differently to form a different coupled model (N). This is the property of model composability that is used to account for structural uncertainty in the rest of this thesis.

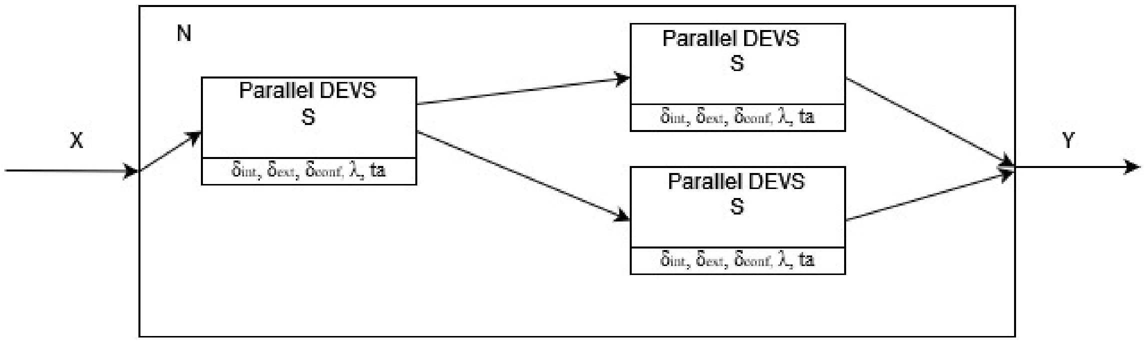

Figure 3.3: Coupled DEVS visualized based on Zeigler et al. (2019a)

Coupled component systems (like seen in figure 3.3) can be specified by using more than one formalism at the same time (Sarjoughian, 2006). In the rest of this thesis the focus is on model composability with few compatibility issues. This is because model composability becomes increasingly complex when the components use different formalisms. In this section, the coupling between several model components in the same formalism is described. Coupling model components in the same formalism is the simplest form of model composability. This type of model coupling is called monolithic coupling. It is also possible to write different components in different formalisms, such as system dynamics combined with an agent based component. However, the coupling then becomes increasingly harder. To account for structural uncertainty it is sufficient to make use of monolithic coupling, because a supply chain can be modelled using a single formalism.

3.3.4. Referential ontology: The System Entity Structure

A referential ontology supports the development of a simulation model. It helps to describe the state of the world, or changes of the state of the world (Zeigler & Hammonds, 2007). In simulation modelling, they are particularly useful to describe and communicate the elements in the model. Examples of referential ontologies are Extensible Markup Language (XML), Unified Modelling Language (UML) and SES. The SES is an ontological framework that is specifically designed for simulation engineering.

Referential ontologies can be distinguished from methodological ontologies (Hofmann, 2013). Referential ontologies refer to the description of real world entities (for example: figure 3.4). By contrast, methodological ontologies refer to the modelling world view that is applied. Examples of methodological ontologies are event-oriented, activity-oriented, and process-oriented world views (Nance, 1981). In this study, only one world view is applied because the type of model coupling that is applied is monolithic. Recall that this means that all components are specified in the same formalism. Within this thesis, an event-oriented world view is applied. This means that changes in state of model objects are caused by events. This world view is chosen because it is inherent to the DEVS formalism.

One particular type of referential ontology that is applied throughout this thesis is the SES. The SES is specifically designed for modelling and simulation, and can be used to specify a family of models (Zeigler & Hammonds, 2007). The SES can be used to specify a system with respect to a specific system boundary. It can be used to specify the relevant entities in a system, and the relations between these entities. The SES consists of a web of entities, and makes use of three types of relations:

- Aspect-relation
- Multi-aspect relation
- Specializations

To illustrate how the SES can be used to specify multiple simulation models, a minimal example of a train simulation model will be given. To demonstrate the SES, the example includes a relation of each type. In a SES, aspect relations are used to reduce entities into parts. For example, a train has wagons and a locomotive. This is an example of a physical aspect relation. An aspect relation can also be non-physical, for example when discussion the relation between cultural concepts. Multi-aspect relations are used when there are many entities of the same kind. For example, a set of wagons consist of many (identical) wagons. Finally, specializations are used to describe the variants of an object. For example, a locomotive can be either green, red or blue. This is an example of a colour specification relation. There are other types of specification relations, such as the specification of a wagon's shape. The examples given in this section are shown graphically in figure 3.4.

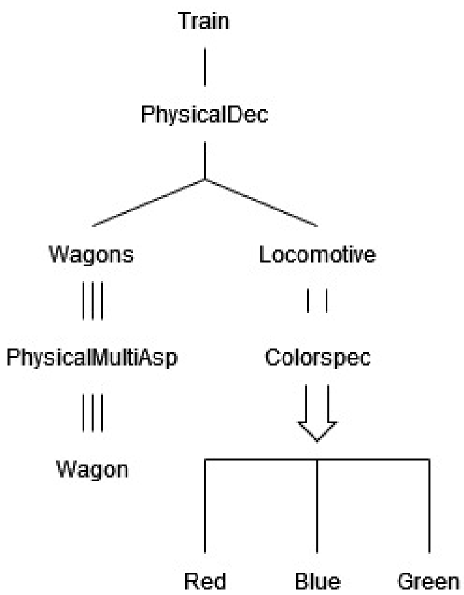

Figure 3.4: An example of a SES

The SES in figure 3.4 can be used to generate multiple simulation models by using a process called 'pruning' (Folkerts et al., 2020). Pruning is the process of deriving a single model configuration from a SES. Using the example from figure 3.4, the SES can be pruned into different trains. For example, based on this SES, an instance of a train can have a red locomotive with ten wagons. It is also possible to prune a train with a blue locomotive with nine wagons. Such a train blue or red train can be called a pruned entity system, which has a non-isomorphic relation with the SES. The process of pruning can be restricted by a table of constraints. For example, a constraint might be that a train with a red locomotive may never have more than five wagons. This prevents pruning from creating invalid instances.

3.4. The relation between DEVS and SES

The relation between DEVS and SES is illustrated in figure 3.5, this figure is based on Zeigler and Hammonds (2007). In this figure, a distinction is made between the ontological level and the implementation level. The ontological level includes the conceptual SES model and the mathematical DEVS models. The implementation level includes the computer implementation entities. From the ontological perspective, the SES provides a static snapshot of the world, from which pruned entity systems can be generated. These pruned entity systems provide a static description of the world, valid within a particular pragmatic frame. The pruned entity system can be transformed to a DEVS model by adding dynamics to the pruned entity system. From the implementation perspective, the SES can be implemented using object-oriented programming. Entities from the SES can be represented using classes. From these classes, typically instances can be generated, representing pruned entity systems.

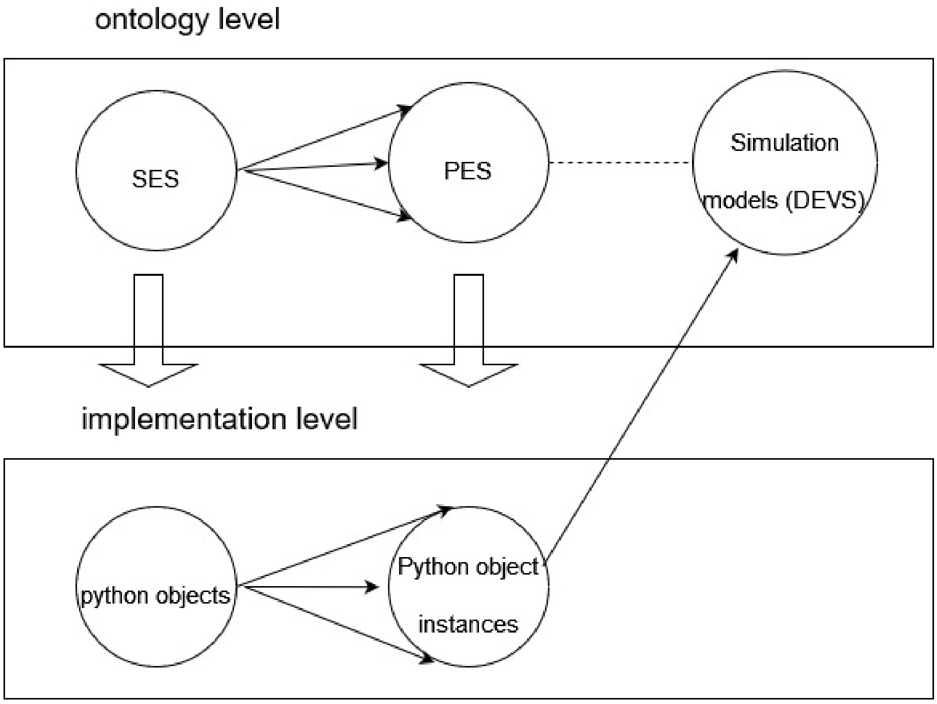

Figure 3.5: Relation between SES and DEVS based on Zeigler and Hammonds (2007)

3.5. Simulation software: Pydsol

All models in this thesis are implemented in a modified version of pydsol. Pydsol is a python implementation of DSOL (a Java-based simulation framework). Pydsol is a relatively new simulation framework, which has some advantages and limitations.

A major advantage of pydsol is that it is based on python. Python is easy to read and has a lot of libraries which developers can make use of. For instance, some libraries provide access to machine learning algorithms and other libraries provide access to algorithms that handle spatial data and so on. Another advantage of pydsol is that it is modular in nature. Because pydsol is modular, it is much more suitable for model composability than other simulation frameworks. It supports the coupling relation that was demonstrated in DEVS.

Pydsol has some disadvantages, however most of its disadvantages can be mitigated. Python is in interpreted language, and can therefore be slow. The low speed is especially an issue when conducting large numbers of experiments. Besides, pydsol does not have any visualization utilities. This makes it hard to visually validate and verify the execution of a simulation model. However, both disadvantages can be mitigated. In this thesis, a modified version of pydsol is used to increase its speed and improve its visualization abilities.

The disadvantage of the slow execution time is mitigated by changing the event list of the pydsol simulation framework. An event list is used to store the actions that should be executed by the simulator. Event lists are commonly used in discrete event simulation frameworks. The underlying data structure of an event list predominately determines the speed of the simulation, especially when there are many events scheduled. Pydsol depends upon an external python library PyTreeMap, which is a python implementation of a red black tree. Pytreemap becomes slow when event lists becomes long. Therefore, an alternative to PyTreeMap was implemented.

The PyTreeMap module is replaced by a C++ module, resulting in speed improvements up to 96%. C++ is a programming language that has a build in priority list, which is much more efficient than PyTreeMap. Priority lists are conceptually based on heaps, which are tree structures that efficiently point to the elements first in queue. C++ is much more efficient due to the fact that C++ is a lower-level programming language, and is a compiled language. The C++ event list is connected to python by using Cython to make it compatible with pydsol. Cython is a python library that facilitates communication between python and C++.

The absense of visualization utilities is mitigated by developing a VUE.js web application. This application shows the static elements of a simulation in an interactive map. Some details about this visualization utility can be found in appendix C.

3.6. Limitations to model-driven exploratory modelling

There are some drawbacks to the use of model composability in modelling and simulation. Most limitations originate to the use of a referential ontology. For certain domains, it is hard to construct an objective referential ontology (Hofmann, 2013). This is especially the case in social and sociotechnical domains. For example, it is almost impossible to construct a referential ontology of concepts such as culture, democracy, and society. Another limitation originates from the fact that a model is created by one modeller, or a group of modellers. Salt (2008) mentions that there is a risk that these modellers see themselves capable of perfectly observing the world. He mentions the risk of modellers seeing their ontology as perfect, and that they do not acknowledge that there are different interpretations of the world. Although these are severe limitations to account for structural uncertainty, model composability might be very successful in the context of supply chain systems. These limitations are mitigated by making a referential ontology based on literature as much as possible. This makes sure that the SES used is grounded in literature. Furthermore, it also mitigates the risk of creating a biased model.

4 Testing model composability with a ground truth

In this chapter, model composability is used to efficaciously account for structural uncertainty in supply chain simulation models. The process of testing model composability on its efficacy to account for structural uncertainty is depicted in figure 4.1. First, a particular model structure is chosen as a ground truth. This model structure acts as a benchmark. Consequently, the ground truth is estimated with a model composer. To limit the amount of plausible model structures, a careful selection of simulation models is simulated. Finally, experiments are conducted to test the method on its efficacy. This chapter focusses on the first two steps: establishing the ground truth and estimating the ground truth with a model composer. Chapter 5 focusses on the last two steps, the selection of simulation models and the comparison to the ground truth (testing).

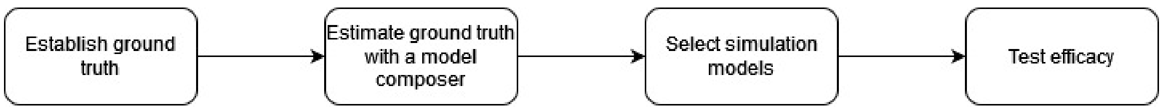

Figure 4.1: Process of testing model composability

4.1. Ground truth

The first step of testing model composability, is to introduce a ground truth. A ground truth is a perfect replication of a target system, which is unequivocally understood by its observer (Khondoker et al., 2016). In essence, it is a model of reality that is perceived as reality. In this thesis, a simulation model of an illicit supply chain of personal protective equipment is considered to be the ground truth. The ground truth can be seen as some sort of 'benchmark'. It provides a baseline to which estimated plausible models can be compared.

The goal of comparing the estimated models to a ground truth is not to select a perfect model. Estimated models do not need to have similar outcomes like the ground truth. The ground truth rather functions as a benchmark that helps to understand which uncertainty is added by removing information from the ground truth.

As a ground truth, a supply chain of illicit personal protective equipment between the Vietnam and Netherlands is chosen. The model is based on a model authored by Isabelle van Schilt. The model represents a supply chain of one product from suppliers to retailers. It is a discrete event simulation model made in pydsol.

4.1.1. Conceptual model

Goods flow from supplier to retailer, trespassing a number of other supply chain actors. This general flow of goods is depicted in figure 4.2. This figure shows the supply chain actors and the links between them. The figure contains several types of simulation elements: sources, servers and sinks. These elements are model components that are build into the simulation framework. In discrete event simulation packages, sources are model components that create entities, in this case products. Servers are model components that process entities. Sinks remove entities from the simulation and delete them. In the models, suppliers are sources and retailers are sinks. Other supply chain actors are servers.

Three suppliers deliver raw goods to three manufacturers in Vietnam. The manufacturers deliver their finished goods to an export port in Vietnam. This export port is located south of Hanoi. From here on, the goods are transferred to a transit port located closely to Hanoi. From the transit port, a larger ship departs to the Netherlands. The ship arrives in the port of Amsterdam (import port), where it is unloaded. The finished goods are trucked from the port of Rotterdam to a wholesaler located in Dordrecht. From here on, the goods are delivered to retailers in Amsterdam, Hoofddorp, Utrecht, Breda and in Rotterdam.

In figure 4.2, green arrows represent land connections and blue arrows represent naval connections. On land connections, either small or big trucks are used to transport goods. Between a supplier and a manufacturer, a small truck delivers the raw materials to the manufacturer. A big truck delivers the finished products to the export port. Feeders (small ships), transfer the goods to the transit ports. Through an intercontinental journey, a large vessel takes the goods to the import port. Afterwards, a large truck takes the goods to the wholesaler. In turn, a small truck delivers the goods from the wholesaler to the retailers.

At each supply chain actor, such as a manufacturer, or a port, the product has to be processed. Processing involves the handling of products, such as storage, transformation, transshipment, and sales. The product is processed for a specific duration, depending on the type of actor. If the actor is occupied, meaning that it is processing another product, the product has to wait before it can be processed. The product is then added to a queue. It is processed whenever the actor is idle again. The queue, including the products that are being processed, represents the stock of the supply chain actor.

The processing times of the supply chain actors are stochastic. They are drawn from statistical distributions which are inspired by a real world fashion supply chain. Arrival processes such as the creation and deletion of products at suppliers and retailers are assumed to be distributed exponentially. Processing times at a manufacturer are assumed to be normally distributed. Ports processing times are controlled by a skewed normal distribution. At wholesalers, a triangular distribution is used. The parameters for these distributions are controlled by model parameters with fixed values.

After the processing time is over, the product leaves the supply chain actor. The product is then coupled to a vehicle, which transfers the product to the next supply chain actor. The speed at which this happens is controlled by the vehicle type. The speeds are stochastic, and are controlled by triangular distributions. Feeders and vessels are assumed to be slower than trucks. Large trucks are assumed to be a bit slower than smaller trucks. This is because large truck with a trailer are often limited to 90 km/h.

In figure 4.2, the supplier in the middle is connected to two manufacturers. In such a case, a unit produced by this supplier could go to both manufacturers. In the ground truth, both manufacturers have an equal probability of receiving the package. Both links thus have an equal selection weight.

One time unit in the model represents a day in reality. A total number of 700 days is simulated. This is chosen, as it takes some days before vessels complete their intercontinental journey. Other than that, the model needs some time to stabilize the outcomes.

The above paragraphs describe the ground truth simulation model. The components, processing times, and simulation time are similar in the generated models mentioned in the rest of this text. Specific connections and configuration are specific for the ground truth.

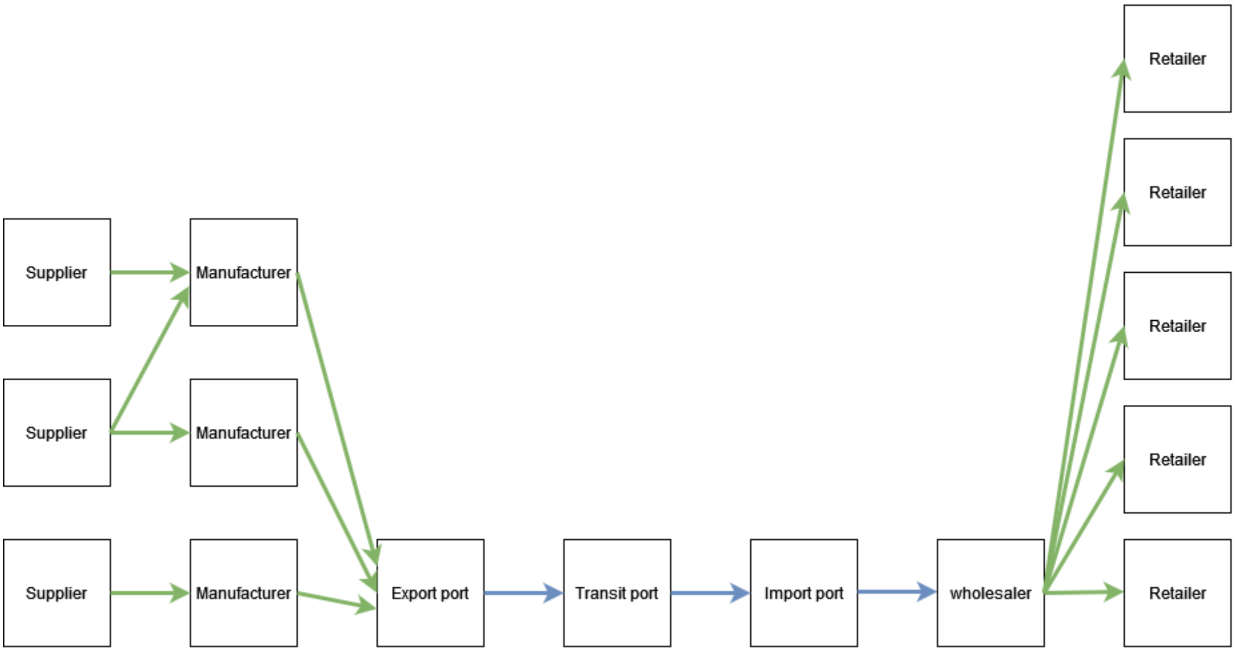

Figure 4.2: Supply chain flow of goods

4.1.2. Model outcomes

The model has several tally model outcomes, e.i. averages of a model run. The outcomes of the model are listed below:

- Average time in system
- Average production time
- Average transfer time
- Average International transport time
- Demand side time
- Average time at wholesalers

The tally statistics are shown in figure 4.3. In this figure, a red dot symbolizes processing and storage time. The arrows symbolize the transport times. The time in system measures the time of the whole chain, other outcomes focus on specific parts of the chain.

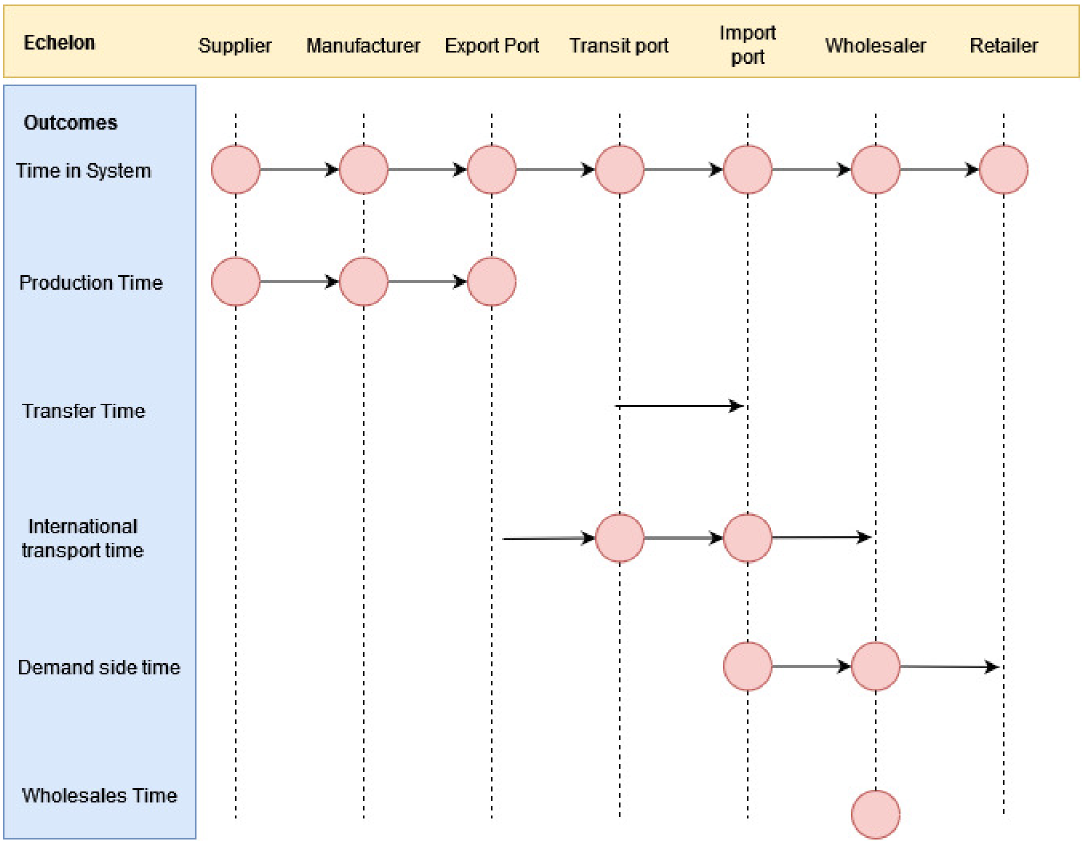

Figure 4.3: Outcomes of simulation models

4.2. Estimating the ground truth with a SES

The SES is used to provide a backbone for the model family for estimating the ground truth. It contains all possible entities in the simulation model. Figure 4.4 shows the system entity structure for a supply chain network. Within in the diagram, all entities that are considered within this analysis are depicted. The diagram contains all types of nodes of the supply chain network, such as suppliers, retailers, wholesalers and ports. These supply chain actors are chosen based on Basu (2013). Other than that, the diagram contains all types of links within the network, such as sea links and land links. The figure consists of several type specifications. Type specifications are inheritance relations that make an object more specific. The figure also has a physical multi aspect relation. This relation is used whenever an object contains several physical objects of the same kind.

The SES can be pruned to a pruned entity system (Folkerts et al., 2020). An example of a pruned entity system is a supply chain network composed of multiple suppliers, a single manufacturer, a transit port, and a set of retailers. Pruning should be done with subject to constraints, because neglecting constraints might cause the composer to generate invalid pruned entity systems. An example of such an invalid pruned entity system is a supply chain network without a single supplier or retailer.

In order to make a plausible family of simulation models, a set of constraints should be specified. The constraints that are needed to generate a plausible family of simulation models vary from parameters to datasets. The constraints are related to the number of supply chain actors that could be in the simulation model, to the type of coupling between the supply chain actors, and to the spatial distribution of supply chain actors, among others specified in section 4.3. To generate a valid pruned entity system, the constraints are considered by the model composer. An overview of all constraints used in this thesis is discussed in section 4.3.

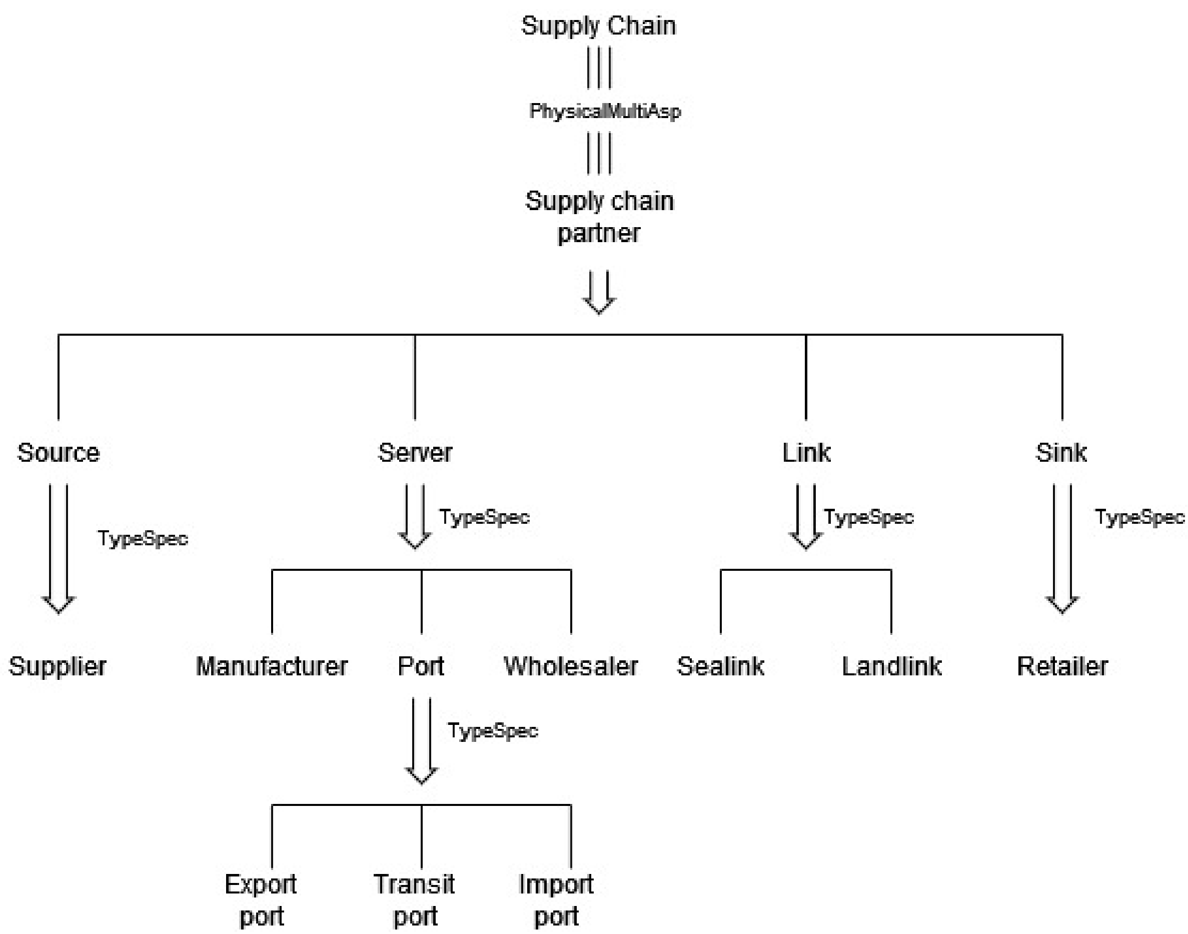

Figure 4.4: System entity structure of a supply chain

4.3. Constraints

The SES is pruned while accounting for the constraints. Constraints are restrictions for pruning the SES. The constraints can be specified using parameters or datasets. The constraints range from the number of supply chain actors, to a specification of which supply chain actors can be coupled to each other, to the spatial distribution of the supply chain actors. All types of constraints used in this research are listed in table 4.1. A more detailed overview of the constraints can be found in appendix A. The appendix consists of all the exact constraints. Besides, this appendix also contains all datasets that are part of the constraints.

In this thesis, constraints are subjective and can not be determined objectively. Constraints restrict the number of models that can be generated. A decision maker should set these constraints based on their beliefs of the actual supply chain. For instance, a decision maker that acts at the end of the supply chain might have different beliefs about the number of retailers than a decision maker involved at the beginning of the supply chain.

Table 4.1: Constraints

Constraint	Description	Example
Number of supply chain actors	Restriction of number of supply chain actors of each type.	There should be at least five retailers, and at most 20 retailers.
Coupling order of supply chain actors	Certain types of supply chain actors can not precede of other types of supply chain actors.	Wholesalers can deliver goods to retailers, but suppliers can not.
Indegree and outdegree of supply chain actors	For some supply chain actors it is unrealistic if they serve many other supply chain actors.	A supplier can not supply thirty manufacturers.
Land use restrictions	Some supply chain actors should be located on a particular type of land use.	Ports should be located close to water. Retailers should be in build-up environments. Build-up locations are areas filled with buildings.
Proximity restrictions	The location of a supply chain actor depends upon the location of another.	A retailer can not be separated from a wholesaler. They should be somewhat proximate to each other.

Note. View appendix A to see an exhaustive list of all restrictions.

4.4. Model composer

The model composer generates model structures to estimate the ground truth. The model composer generates a model structure by coupling predefined model components of the SES in a specific way. The model composer creates a pruned entity system from the SES.

The model composer is conceptually based on graph theory. A Directed Acyclic Graph (DAG) is used to represent the components in a model, and the coupling relations between them. A directed graph is a graph, with nodes and directed edges. Directed edges are edges that have a specific direction. Graphs are acyclic when they do not contain any circular paths. The choice for a DAG is substantiated by the fact that a supply chain is linear with a directed flow of goods.

A DAG is created in five steps by the composer (visualized in figure 4.5). Step 1 and 2, determining the number of nodes per type and establishing the edges are explained in section 4.4.1. Step 3, the assignment of the locations is introduced in section 4.4.2. Finally, section 4.4.3 includes how the edge weights are computed, and how the DAG is converted to a pydsol simulation model.

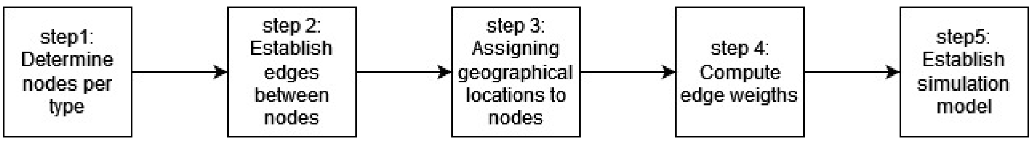

Figure 4.5: Working of the model composer

The model composer is implemented in python in order to ensure that it can easily interact with the python based simulation package pydsol. The python package Networkx is used to create the network structures and to evaluate them. The python packages Rasterio and Geopandas are used to facilitate the geospatial computation.

To make sure that everything is implemented as outlined in the rest of this chapter, over fifty tests are conducted to validate the behaviour of the model composer. Some tests are developed to test the algorithms that return the distance between two points. Others are developed to test whether validation algorithms are implemented as intended. Validation algorithms are algorithms that check if the hyperparameters of the model comply to the constraints. Other tests are used to test whether the spatial algorithms work as assured. All tests can be found in appendix B. Furthermore, a visualization utility was used to verify if the models were successfully implemented. More about this visualization utility can be found in appendix C.

4.4.1. Step 1 & 2: Creating nodes and edges

In the first two steps of composing a model, a number of nodes and edges is created. Firstly, the number of nodes is determined, and secondly the edges are created. The generation of nodes and edges is subject to constraints. An algorithm is used to assure that a valid DAG can be generated, while complying to the constraints.

To create the nodes, a random number of supply chain actors for each type of actor is generated. This is done based on a uniform distribution, for each actor type. The model composer assures that the number of supply chain actors per type is within a predefined range. There should be at least one supply chain actor of each type. This is an assumption that is part of the constraints. Secondly, the model composer validates whether this number of supply chain actors can be composed to a valid graph by checking whether the minimum and maximum indegree and outdegree of the supply chain actors match with each other. For example, if there is just one supplier, and a supplier has a maximum outdegree of three, there should be no more than three manufacturers. The first two steps are repeated until a valid number of supply chain actors per type is found.

In the second step, the model composer starts randomly drawing edges between the nodes, while accounting for the maximum and minimum in and outdegrees. Again, this process is repeated, until a valid DAG is found. An example of a valid DAG is shown in figure 4.6. In this figure, each colour refers to a type of supply chain actor.

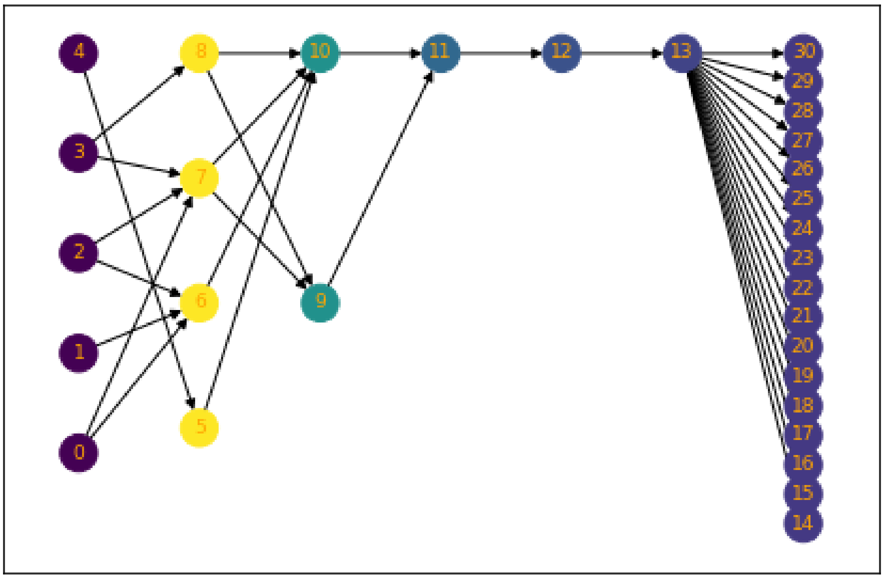

Figure 4.6: A valid directed acyclic graph of a supply chain

4.4.2. Step 3: Geographical location of the nodes

The generated DAG forms the basis for the supply chain structure, but it does not yet accurately resemble reality. This is because the nodes do not have a spatial location, and the distances between them are not determined in previous steps. To ensure that the nodes get a valid geographic position, an algorithm is used to make sure that supply chain actors are placed within a plausible position. A difficulty is that the location of the supply chain actors are conditional to each other. For example, a retailer can not be separated hundreds of kilometres from its wholesaler, as that is likely to be untrue. Other than that, ports should be located near the water, and manufacturers should be close to their port(s) etc. Like stated before, such requirements are called constraints.

To assign locations to the nodes, an algorithm is developed to comply to these constraints. In this algorithm, first port locations are sampled from a dataset with ports in Vietnam and the Netherlands. Import ports are assumed to be in the Netherlands, Belgium or Germany, whereas export and transit ports are assumed to be in Vietnam. After the ports are sampled, the rest of the locations are sampled. The rest of the locations, those of retailers, wholesalers, manufacturers and suppliers are assumed to be in build-up environments. A location is considered to be build-up, i.e. filled for the most part with buildings, whenever it is classified as such in the Copernicus land cover dataset (Buchhorn et al., 2020). Besides the requirement that these supply chain actors should be in build-up environment, the supply chain actors should also be located near their predecessor or successor. For example, a retailer should be close to its wholesaler(s) (its predecessor). By contrast, a manufacturer should be located near the port(s) it is delivering too (its successor). To ensure that this requirement is met, buffers are drawn around the predecessor or successor supply chain actors. A zone is then generated by intersecting the buffer with the administrative borders of the country the supply chain actor should be situated in. The supply chain actor is then placed within the buffer, on a random location that classifies as a build-up location. Buffers are zones around a spatial feature, in this case they look like circles. If there is more than one predecessor or successor, several buffers are drawn, and consequently they are intersected. The supply chain actor is then placed within the intersection of the buffers. If there is no intersection, the buffers are enlarged until there is an intersection.

An example of the location allocation process is shown in figure 4.7. In this figure, the geometric assets needed to locate a wholesaler are shown. The map shows where a wholesaler connected to the port of Antwerpen can be located. This wholesaler can be placed on a random build-up location indicated by red. These build-up locations are identified in three steps. First, a buffer is drawn around the port of Antwerpen. This buffer is indicated by the blue circle. Second, the buffer is intersected with the administrative boundaries of the Netherlands. This is indicated by the green area. Finally, the resulting intersection is used to clip a land-use dataset. This land-use dataset contains all build-up locations.

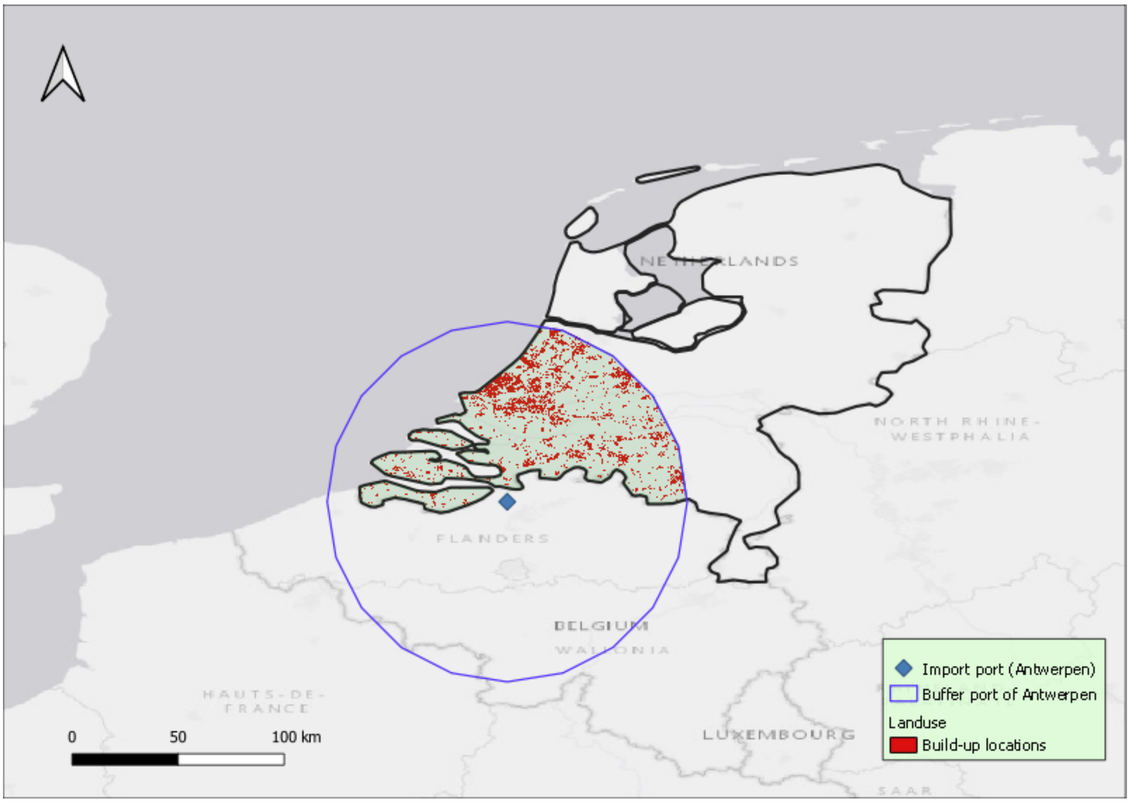

Figure 4.7: Example of location placement of a wholesaler

4.4.3. Step 4 & 5: Computation of edge weights and conversion to simulation model

The fourth step of the model composer is to compute edge weights. Edge weights are determined by the distance between the two nodes. How the distance between two nodes is computed depends on the edge type. There are two types of edges: those which represent an overseas connection and those which represent a land connection. The type of the edge is determined by the type of nodes it is connecting to. An edge between a wholesaler and a retailer is a land connection (land edge), while a connection between two ports is a sea connection (sea edge).

The length of a land connection is computed using a 'straight' (Euclidean) line between two points. The coordinates of the nodes that are connected by the edge are projected in an equidistant worldwide map projection. This projection system is suited to compute the Euclidean distances between two points. It corrects for the round shape of the earth, which guarantees that distances can be computed with relatively small distortion.

Figure 4.8 shows the locations of the supply chain actors and the land connections between them. The locations shown in this map are generated using the DAG depicted in figure 4.6. As can be seen, the routes between the locations are indicated using straight lines (arrows). The lengths of these lines are computed and saved as attributes within the DAG.

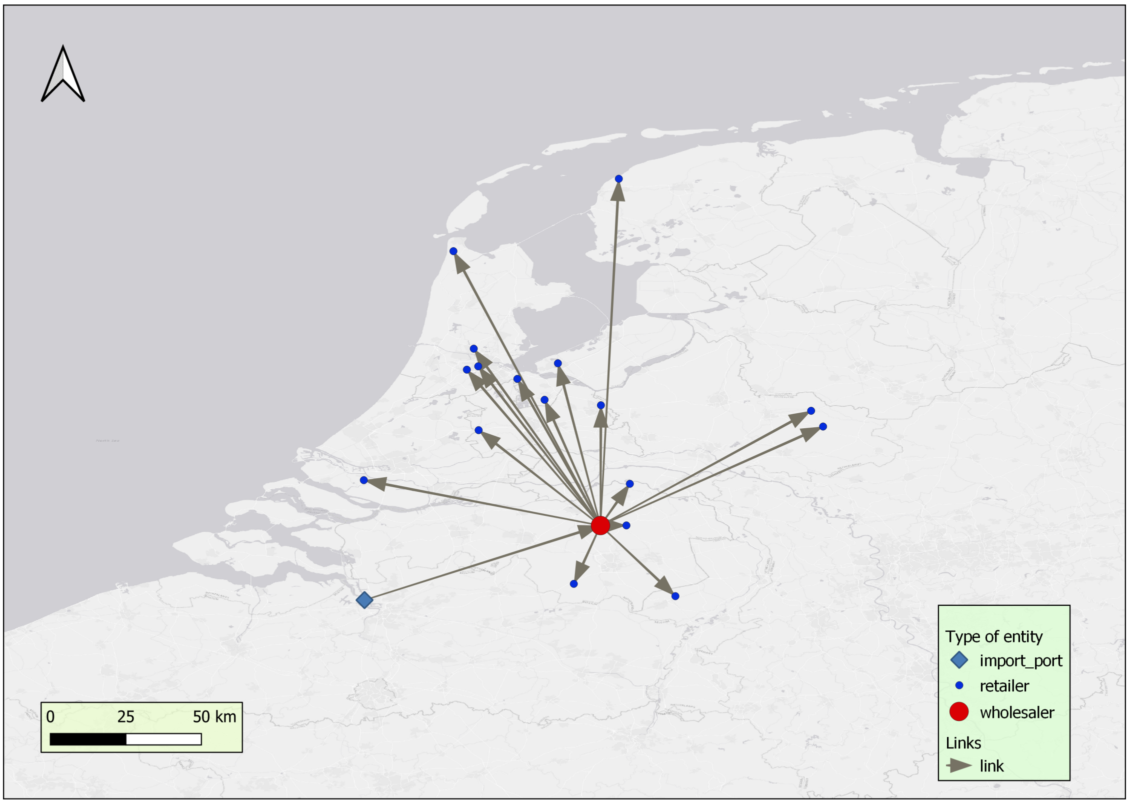

Figure 4.8: Physical supply chain structure (Netherlands), starting from the import port

To compute the length of a sea connection, the route between two locations is computed with a cheapest route algorithm. The Copernicus land cover dataset (Buchhorn et al., 2020) is used to determine the ship routes. The land cover dataset is a grid type of dataset (commonly called raster dataset) which has a land cover classification for each cell. The land cover dataset has over 30 classes indicating whether land is classified as urban area, agricultural area or forests. This classification scheme is manipulated by classifying each type of water as water, and each different type of land as land. The resulting dataset is a binary classified dataset with land and water. The dataset is converted from an image to an array. The route was computed using an algorithm that searched for the cheapest route through an array. Water cells are assigned a low cost, and land cells are assigned a high cost. An example of a route computed using this approach is shown in figure 4.9.

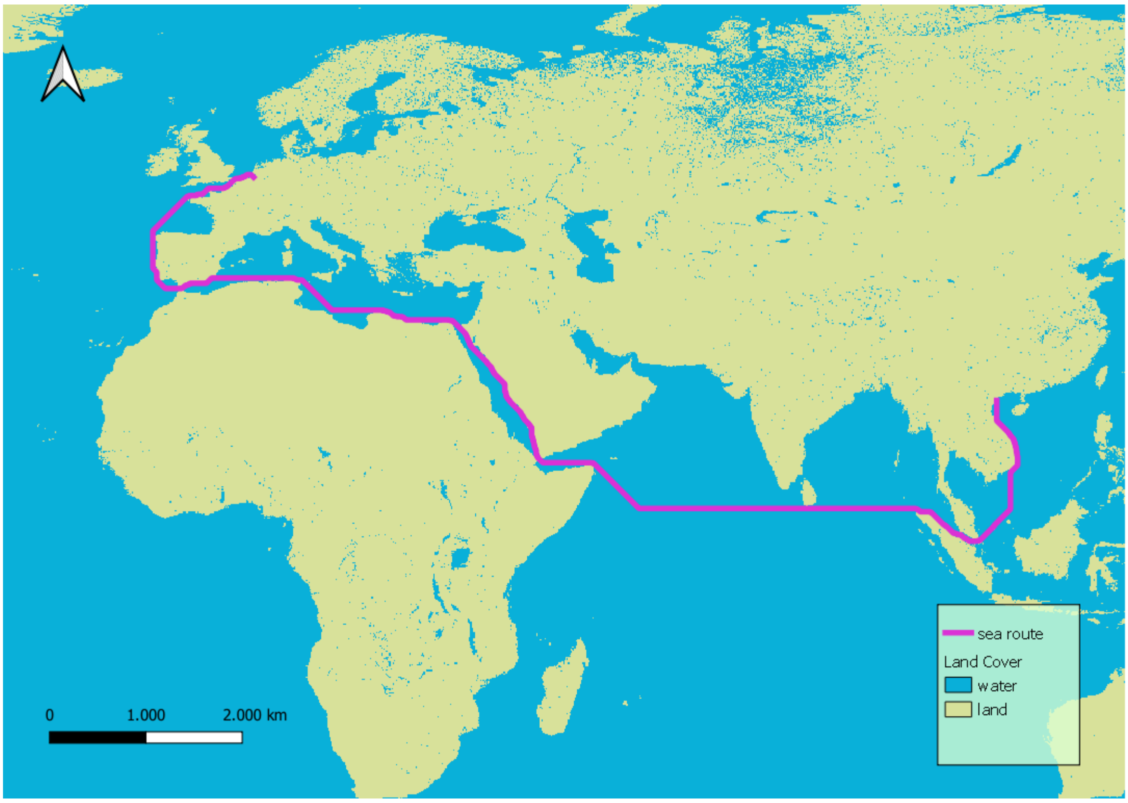

Figure 4.9: Sea route between a port in Vietnam and a Dutch port

After a DAG has been generated, the final step is to convert a DAG into a simulation model. Nodes in the DAG are converted to python class instances. The python classes comply to the SES described in section 4.2. Edges between nodes are represented by python class instances named links. A Link establishes a coupling relation between two components. The class instances are all part of a pydsol simulation model. Due to of the modular nature of the simulation environment pydsol, the components function together to form a simulation model.

5 Experimental setup

This chapter continues on the concepts introduced in chapter 4. Whereas chapter 4 focusses on the first two steps of the testing process, this chapter focusses on the last two steps. In the previous chapter, a ground truth is established, and the model composer is explained.

This chapter first outlines how a selection in simulation models is made. Second, this chapter highlights how the selected models are used in an experiment. In section 5.1, the process of selecting simulation models is substantiated. In section 5.2, an analysis that substantiates how many times each model should be replicated is presented. In section 5.3, an experimental setup is laid out to test the efficacy of the method.

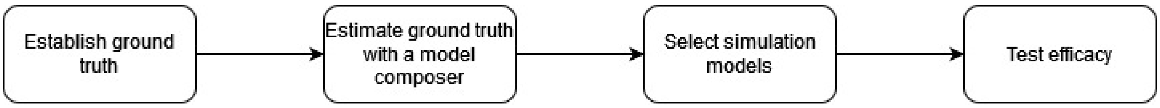

Figure 5.1: Process of testing model composability

5.1. Selection of plausible models

Many simulation models are generated by the model composer. None of these models should be seen as wrong or invalid, because all of them comply to the constraints given to the model composer. However, it is not feasible to simulate all of them, due to a high computational burden. Besides, it is not interesting to simulate all models, because models become more and more similar if more models are generated. Therefore, this chapter first focuses on making a selection in simulation models that are simulated and analysed. It is of importance, that the chosen simulation models differ as much as possible, to make sure that the results are as meaningful as possible.

A selection in simulation models is made in three steps. First, a set of topological features to describe a DAG is selected. The second step is to generate a set of DAG's and determine a suitable number of DAG's. The third step is to sample from this set of simulation models. This is done using the k-means algorithm.

The first step is to identify topological features. Topological features are needed to characterize a DAG. Topological features are properties of a DAG, expressed in numerical values. Topological features are needed to compare DAG's, due to the fact that it computationally expensive to compare a DAG to another DAG. Aside of the fact that such a comparison is computationally expensive, many DAG's are only slightly different. For instance, two DAG's are considered completely different, when in fact only the distance between two nodes might differ slightly. Even when the link length is not considered when comparing network structures, but only the number of nodes and their edges, the total number of DAG's that can be generated is still too high.

Transportation networks have a number of topological features that can be leveraged to make a selection in networks (Calatayud et al., 2017; Lin & Ban, 2013; Wang et al., 2011). Examples of such topological features are mean degree centrality, efficiency and mean betweenness. In this study, the topological features shown in table 5.1 are used to make a selection in simulation models. These are chosen, because they are commonly used in research about transportation networks (Calatayud et al., 2017; Lin & Ban, 2013; Wang et al., 2011). In addition to that, a regression analysis was conducted to validate whether these topological features can describe a DAG successfully. In this regression analysis, the topological features of 700 models are compared to the simulation model outcome 'time in system'. The time in system outcome was chosen, as it measures the 'whole' supply chain. The time in system is the average time of a product that flows from supplier to retailer. In the regression analysis, the topological features are used as independent variables. The time in system outcome of the simulation model was used as the dependent variable. The explained variance of the regression model (r^2) is 0.391, as seen in appendix D. Albeit that can be higher, a significant proportion of the variance is caused by the inherent randomness of the simulation model. Therefore, the chosen topological features sufficiently describe a DAG.

Table 5.1: Typology of directed acyclic graphs

Metric	Description
Number of nodes	Total number of nodes (supply chain actors) of the supply chain simulation model.
Number of edges	Total number of edges in the supply chain simulation model.
Mean degree centrality	Average number of edges that connect to a node.
Mean betweenness centrality	Average extent to which a node lies on the shortest path between other nodes.
Mean closeness centrality	Average length of the shortest path between all pairs of nodes.

Note. Metrics are based on Calatayud et al. (2017), Lin and Ban (2013), and Wang et al. (2011)

The second step is to generate a suitable number of DAG's. It is not feasible to generate all DAG's because this is computationally expensive. Therefore, a limited number of DAG's is generated. By selecting a limited number of DAG's some plausible models are not operationalized. Because of this, it is vital to make sure that the generated structures are similar to the total set of DAG's. To make sure that the DAG's generated are as different as possible, Shannon's entropy measure is used. Using Shannon's entropy measure, the variety of a random variable (entropy) can be computed (Shannon, 2001). Shannon's entropy measure is used, because it helps to quantify the variety of a dataset. If the variety does not change anymore, enough models are generated. The higher the entropy, the more variable a dataset is. For example, the entropy of a random variable with two values: heads (p = 0.5) or tails (p = 0.5) is lower than the entropy of a random variable with three values: red (p = 0.33), blue (p = 0.33) or green (p = 0.33).

The original measure can be used to compute the entropy of a discrete variable, such as gender. Because the topological features edges and weights are discrete, the original Shannon entropy measure is used. The centrality metrics are continuous variables. Therefore, differential entropy for continuous variables is used for these variables. There are several methods to estimate differential entropy for continuous variables, all of them compare similar in terms of performance and outcomes (Alizadeh Noughabi, 2015). Because the performance and outcomes of all these approaches are similar and the ebrahimi method is implemented in python, the ebrahimi method is chosen.

The estimated entropy values for each topological feature are shown in figure 5.2. The x-axis shows the model set size, which is the number of DAG's generated. The figure shows that after generating about 700 DAG's, the entropy value does not change. Therefore, in all experiments the number of DAG's generated is 700.

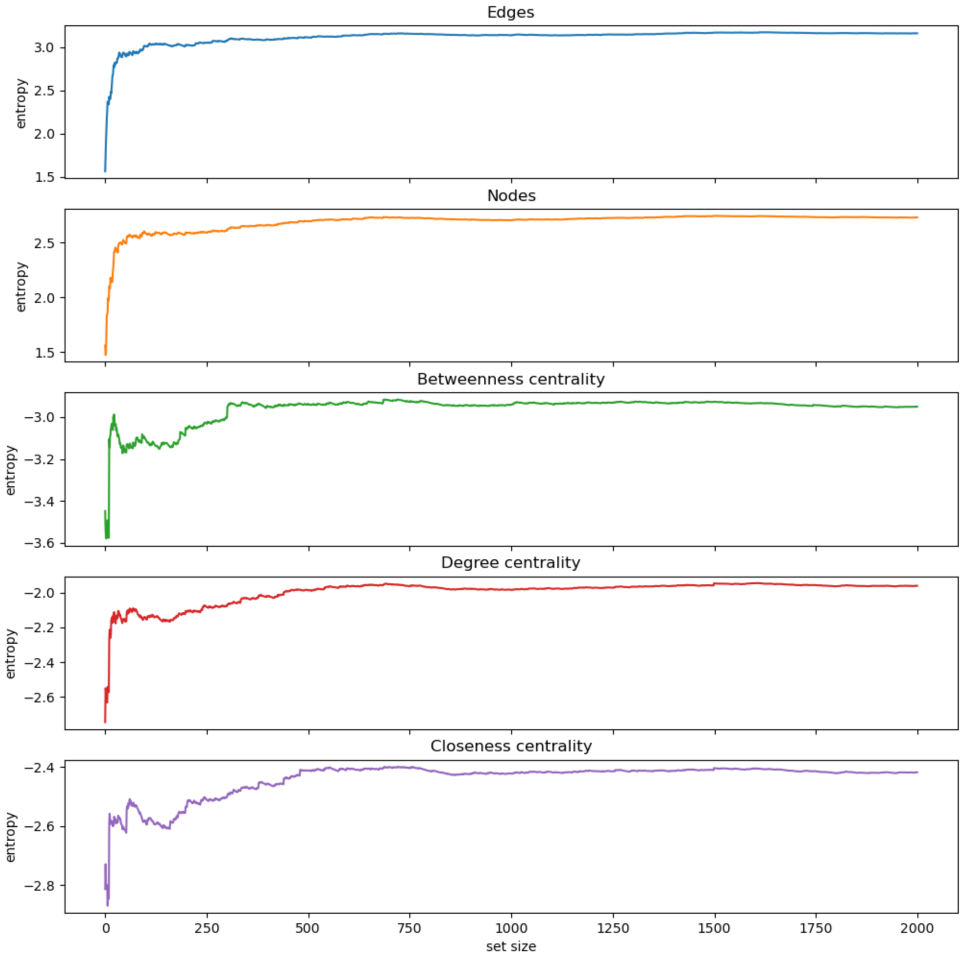

Figure 5.2: Entropy computed for each set size

The third and final step is to use the k-means algorithm to reduce the total number of DAG's. The number of DAG's is reduced because it is computationally unfeasible to simulate all 700 model structures, while doing enough replications. To keep the simulated models as diverse as possible, it is important to keep a subset of models with as diverse topological features as possible. The k-means algorithm is used to select a diverse subset of these 700 DAG's. The k-means algorithm is an unsupervised machine learning algorithm that makes k clusters. The algorithm tries to minimize (euclidean) distances within a cluster, and maximizes distances between other clusters. This makes sure that the DAG's within each cluster are as similar as possible, but the clusters are as different as possible. Selecting a DAG from each cluster, makes sure that the models are as different as possible. A cluster has a value for each of the topological features described in table 5.1.

K is chosen by the number of structures that should be sampled from the set of DAG's. From each cluster, one random DAG is selected. In this manner, the k-means algorithm ensures that the selected DAG's differ as much as possible from each other, with respect to their topological features. The choice to select a random model from a cluster is made deliberately, because in this way models are retained from being unselected based on their topological features. Every model still has a probability greater than zero of being selected.

5.2. Seed analysis

Using the process described in section 5.1 a selection of models is used in further analysis. The selected models are stochastic in nature, this means that their outcomes are not deterministic. Each simulation generates different outcomes. Therefore, it is of importance to replicate the simulation several times.

To determine a suitable number of replications for each model, a seed analysis was conducted. Every model has a seed, which controls the stochastic behaviour of the model. Whenever the seed of a model is fixed, the outcomes of the model are always the same. In a seed analysis, a model is replicated several times with a different seed. Every time a new replication is done, the average of all replications is computed. Replications of a model are done, until the average stabilizes. In a seed analysis, the goal is to find a suitable number of replications for each model run.

Figure 5.3 shows the seed analysis that is conducted with a constant model structure. In this figure, it can be seen that the average outcome value fluctuates whenever the number of replications increases. The figure shows the average outcome value for six different outcomes: average time in system, average international transport time, average time at wholesaler, average production time, average transfer time, and average demand side time. In figure 5.3, the mean value for all outcomes tends to stabilize after 60 replications. Therefore, a simulation is replicated 60 times for each model.

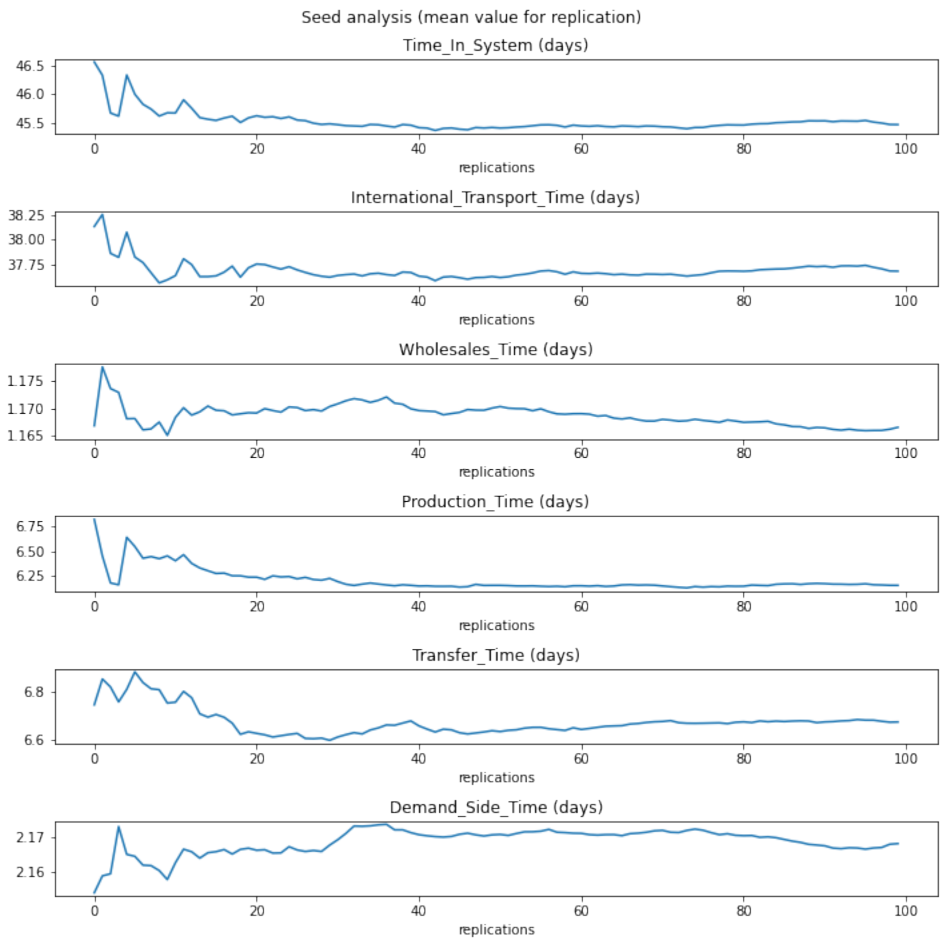

Figure 5.3: Mean value of four kpi's with increasing number of replications

5.3. Comparing the ground truth to the plausible structures

The final step of testing model composability is to estimate the efficacy of model composability to account for structural uncertainty. The efficacy of developed model composer is evaluated by comparing the plausible structures to the ground truth. Testing the efficacy is done in three steps:

1. Simulate ground truth.
2. Estimate ground truth using model composer and model selection process.
3. Estimating the relation between the model outcomes and the model structure.

In the first step, the ground truth is simulated to provide a benchmark. In the second step, this benchmark is estimated by generating and simulating plausible model structures. The aforementioned selection process is relevant in this stage. In the final step, a regression analysis is conducted to see which elements of the model structure contribute to different simulation model outcomes.

In the first step, the ground truth is simulated. The ground truth is a fixed model structure, introduced in chapter 4. This model structure is replicated 100 times.

In the second step, the ground truth is estimated by varying the constraints that the model composer uses. As the structure of an illicit supply chain is unknown and uncertain, there is no objective way in which these constraints can be set. Therefore, five constraint sets are chosen to reflect different perspectives on an illicit supply chain:

- **Base set (set 1):** Base set of constraints
- **Different transit ports (set 2):** Different dataset for transit ports.
- **Larger retailer area (set 3):** Different boundaries for locations wholesalers and retailers.
- **Larger retailer network (set 4):** Larger number of retailers and wholesalers.
- **More suppliers (set 5):** Higher number of suppliers and manufacturers

Constraint set 2 to 5 are all variations to the base set of constraints. Variations are restricted to one aspect of the constraints to see what a change in perception of these factors do. Ideally, a full factorial would have been conducted. However, due to computational constraints this is not possible. Therefore, only a selection of the constraints are varied, not all constraints are varied. The modifications are summed below:

- In constraint set 1, the base set of constraints is used. In this constraint set, transit and export ports are located in Vietnam, and wholesalers and retailers are located in the Netherlands. Furthermore, relatively low numbers of entities are used. Besides, entities do not have high in and out degrees.
- In constraint set 2, a different dataset for the transit ports is used. In this constraint set, transit ports can be located outside of Vietnam. The transit ports in this dataset are located in China, Indonesia, and several other deep sea terminals.
- In constraint set 3, the location in which wholesalers and retailers may be placed is changed. In this constraint set, retailers and wholesalers may be located in the west of Germany, the North of Belgium, and the Netherlands. Furthermore, the range they can be allocated in, is expanded from 80.000 to 200.000 meters.
- In constraint set 4, a larger number of retailers and wholesalers is assumed, compared to the base set of constraints. In the base set, a maximum of 10 retailers and 3 wholesalers is assumed. In this constraint set, this number is increased to 20 retailers and 3 wholesalers.
- In constraint set 5, the maximum number of suppliers and manufacturers is changed. The number of suppliers is raised from 5 to 15, and the number of manufacturers is increased to 10.

The above constraint sets are chosen to vary as much of the constraints as possible, while keeping the experiment's computational costs relatively low. Furthermore, all changes are related to the structure of the supply chain. Vilko et al. (2014) identify that the structure of an (illicit) supply chain is uncertain. The exact values for these constraint sets can be found in appendix A.

The final step of testing the efficacy of model composability is to estimate the relation between model outcomes and constraints. This is done using several regression analyses, conducted using all the models generated in the previous step. In this final analysis, the focus is on what specific elements of a model structure cause the model outcome to change.

The independent values of these regression models are highly related to the constraints of the model composer. For example, the number of supplier and retailers are included in the analyses. Furthermore, the average length of a (sea) link is also included. Besides, the (summed) in degrees and out degrees of each supply chain actor type are also included.

6 Results

In this chapter, the results of the simulation experiments are presented. The results shed light on the efficacy of the model composer in accounting for structural uncertainty. First, the influence of various constraint sets on the outcomes of the simulation models are covered in section 6.1. Afterwards, section 6.2 covers the relation between the model structure and the simulation outcome.

6.1. Influence of the different constraint sets

The aim of this analysis is to find differences and similarities between various perspectives on an illicit supply chain. Each perspective contains one hundred models, each generated by using a unique constraint set. Recall that there are five constraints sets, and multiple simulation outcomes.

The five constraint sets are compared to the ground truth, which functions as a benchmark model. The ground truth helps to understand how a single simulation model relates to a set of simulation models. It helps to see the influence of varying the structure on the simulation model outcomes.

To prove that structural uncertainty affects the simulation outcomes, it is of relevance to test whether each outcome is distributed differently within each constraint set. Therefore, several two sample kolmogorov-smirnov tests are conducted. A two sample kolmogorov-smirnov test statistically tests whether two samples are drawn from the same (unknown) distribution or not. An advantage of a two sample kolmogorov-smirnov test is that it tests the difference between two distributions, rather than the mean, the median or the standard deviation.

The p-values of the two sample kolmogorov-smirnov tests are shown in figure 6.1. In this figure, six heatmaps are presented, one for each simulation outcome. Each cell represents a single two sample kolmogorov-smirnov test. If there is a significant difference between the two constraint sets, the cell is coloured white. If not, the cell is coloured blue. The figure shows that most constraint sets differ significantly from each other and from the ground truth, with a few exceptions. This means that all constraint sets and the ground truth yield different distributions for each simulation outcome, with a few exceptions. The exceptions relate to the transfer time and the wholesales time, indicating that a different constraint set has less effect on these simulation outcomes.

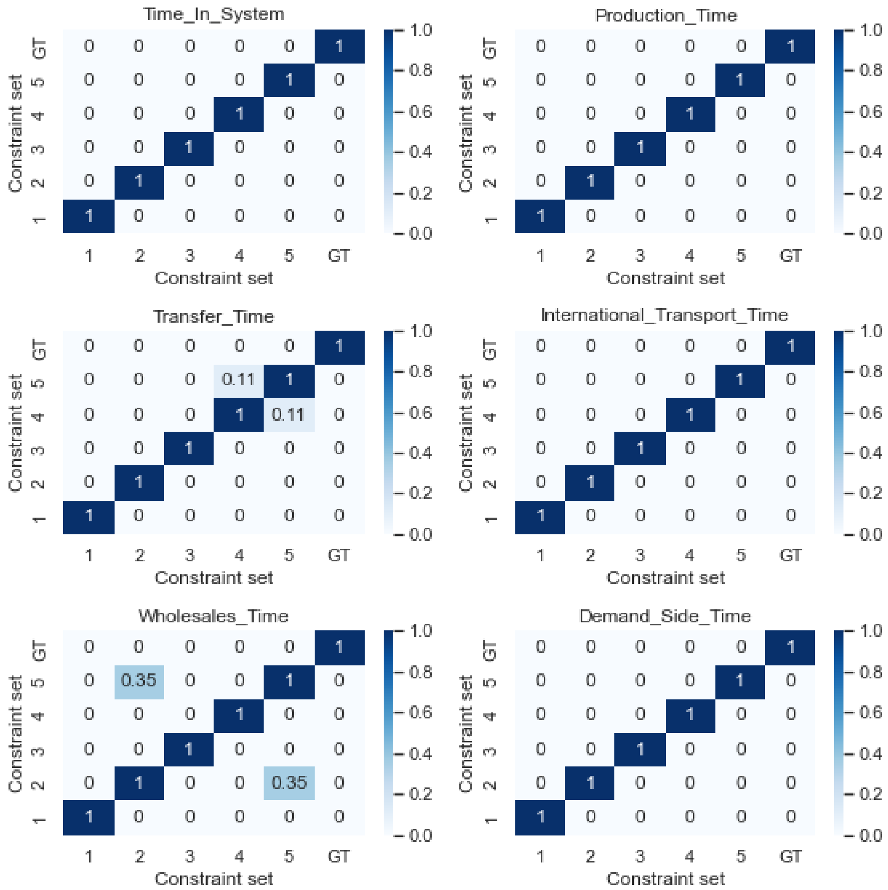

Figure 6.1: Pvalues of two sample kolmogorov-smirnov test, coloured by significance level of 5%

The two sample kolmogorov-smirnov tests indicate that the distributions of the outcomes within each constraint set differ. However, the tests do not reveal the nature and the magnitude of the differences. To provide insight in the magnitude of the difference between the constraint sets, KL-divergence is computed. KL-divergence is a measure that can be used to compute the relative difference between two distributions. The higher a KL-divergence value is, the bigger the difference between two distributions is.

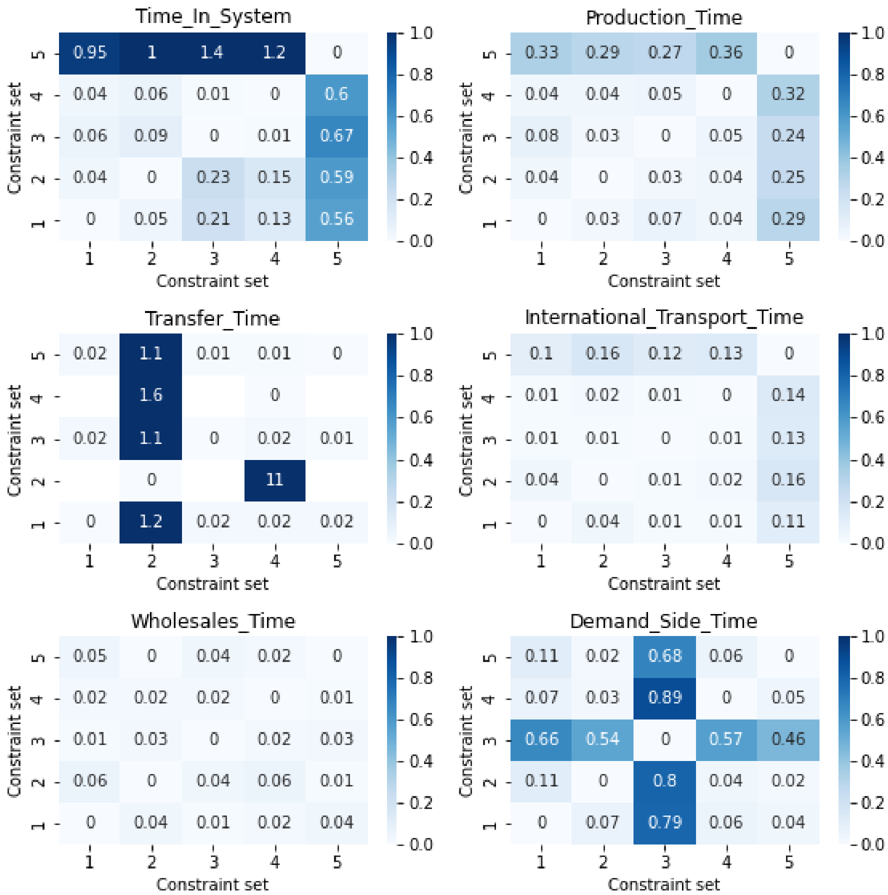

Figure 6.2: KL-divergence values of constraint sets kernel density functions. No value represents infinite.

Figure 6.2 shows the KL-divergence values of the constraint sets. The figure indicates that constraint set 5 has a different distribution when it comes to the time in system, the production time, and the international transport time. Constraint set 2 differs marginally from the other constraint sets when it comes to the transfer time. Constraint set 3 differs with respect to the demand side time. To explain and inspect the differences, several histograms and kernel density figures are plot throughout the rest of this analysis.

Figure 6.3 shows histograms and a kernel density estimates of the time in system. Recall that the time in system portrays the average total time a single product takes from the beginning of the supply chain to the end. Within figure 6.3 constraint set 1, 5, and the ground truth are depicted. These sets are chosen, because the KL-divergence values indicate that constraint set 5 differs the most from the base set of constraints. The histogram shows that a higher time in system is more frequent in constraint set 5 compared to constraint set 1. Recall that constraint set 5 contains a higher restriction for the number of suppliers. This higher number of suppliers is causing congestion in the system, causing many products to delay. The delay in its turn leads to a higher time in system. Figure 6.3 also shows that the constraint sets 1 and 5 differ significantly from the ground truth. The ground truth is distributed in a much smaller range than the constraint sets. A single model structure thus provides a far less uncertain perception of the time in system.

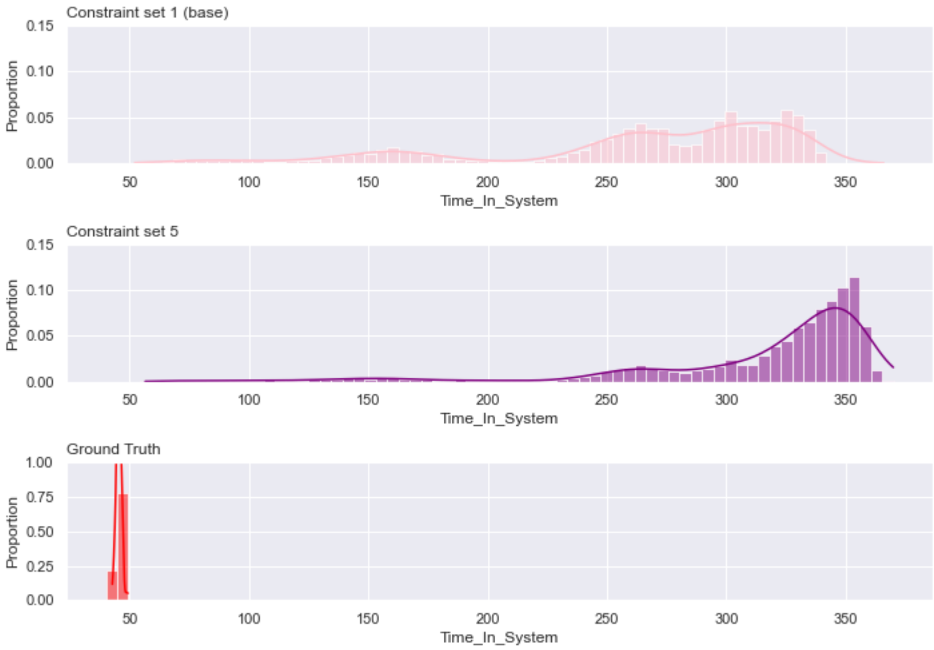

Figure 6.3: Histogram and kernel density distributions of the time in system

Figure 6.4 shows the distributions of the demand side time. The demand side time is the time it takes to ship a package from an import port to a retailer. Within this figure, the ground truth and constraint sets 1 and 3 are shown. The peak of constraint set 3 much more centred to the right than the base set of constraints (set 1). Recall that constraint set 3 has a larger area for wholesalers and retailers. In constraint set 3, retailers and wholesalers may be located in the Netherlands, parts of Belgium, and parts of Germany. In the other constraint sets, retailers and wholesalers may only be located inside the Netherlands. The figure indicates that the median travel times are larger in constraint set 3 than in base set of constraints. This figure combined with the two sample kolmogorov-smirnov tests shows that a different area for wholesalers and retailers result in higher demand side times.

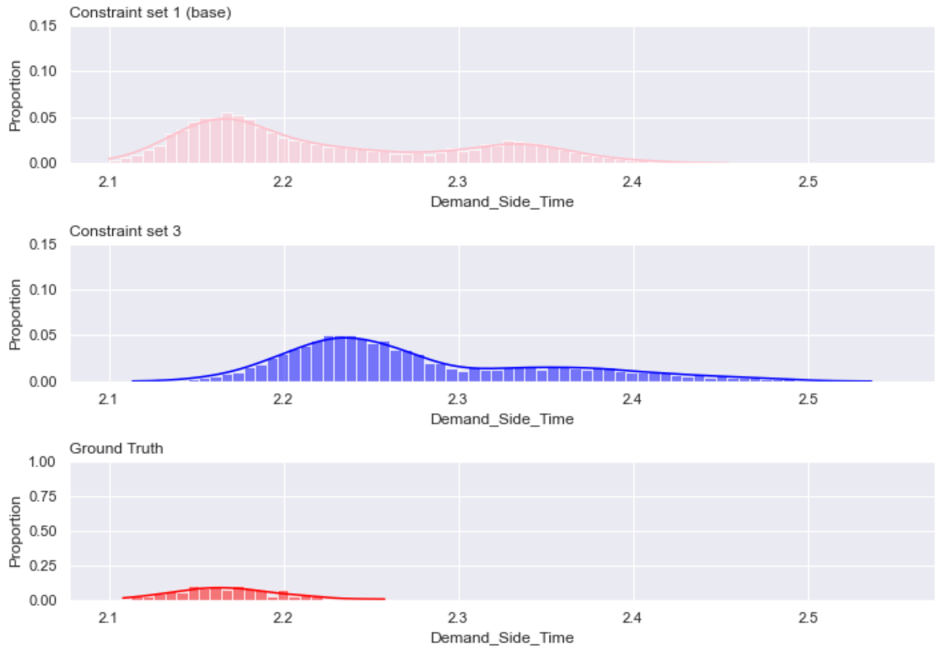

Figure 6.4: Histogram and kernel density distributions of the demand side time

Figure 6.5 shows the production time of constraint set 1 and constraint set 5. The production time is the time it takes to process and ship a product from a supplier to an export port. The figure shows that the median of production times is higher in constraint set 5 compared to constraint set 1. Constraint set 5 allows models to have more suppliers, resulting in more congestion. Due to the congestion, products have to wait longer before they can be processed and shipped.

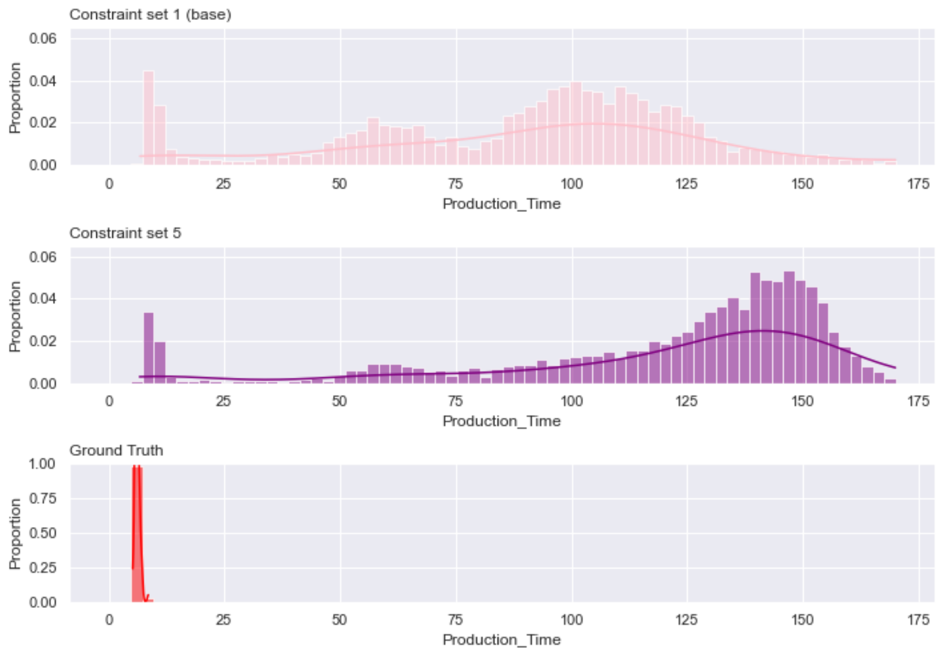

Figure 6.5: Histogram and kernel density distributions of the production time

Figure 6.6 shows the transfer time of constraint set 1 and constraint set 2. The transfer time is the time a product takes to ship a product from an export port to a transit port. Recall that constraint set 2 allowed the model composer to create models with a different transit port dataset. The transit ports in constraint set 2 are located outside Vietnam, spread around Asia. Whereas in constraint set 1, all transit ports are located inside Vietnam. Figure 6.6 shows that the transfer times of constraint set 2 have a longer range. Transfer times range from 7 to 12, instead of 7 to 9.

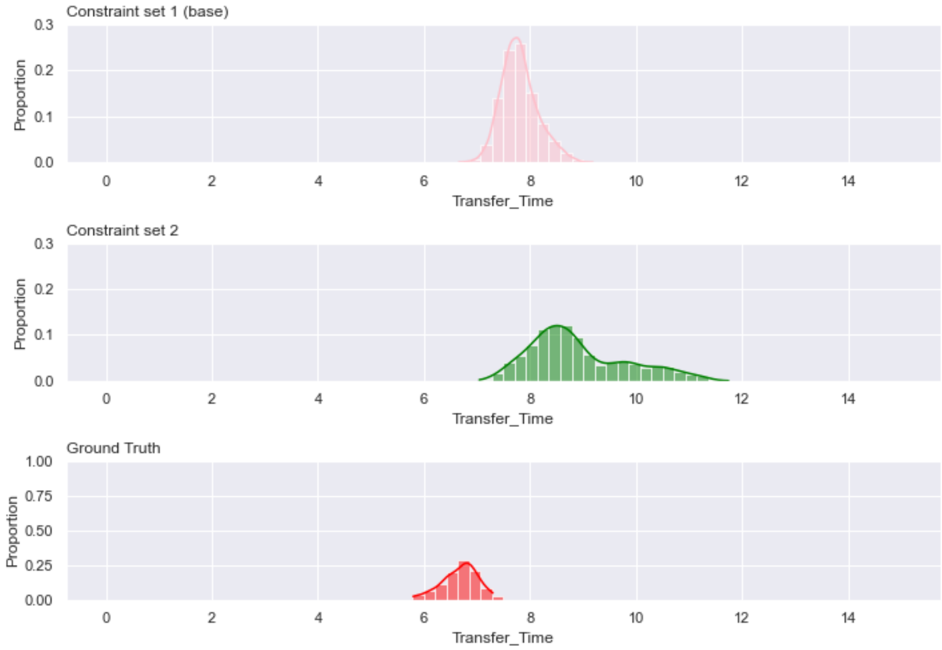

Figure 6.6: Histogram and kernel density distributions of the transfer time

6.2. The relation between model structure and model outcomes

The aim of this analysis is to find which parts of the model structure are associated with the simulation model outcomes. To do this, six regression analysis between the model structure elements and the model outcomes are conducted. The independent variables of these regression analyses are model structure variables, which are values based on python model objects. For instance, the number of suppliers and retailers. The dependent variables of the regression models are the simulation outcomes.

The standardized regression coefficients of the regression models are depicted in figure 6.7. The coefficients provide an insight in which model structural elements affect the simulation model outcomes most significantly. In this figure, the columns indicate simulation model outcomes, and the rows indicate the model structural elements. The cells of the heatmap are coloured blue if the relation is negative, and red if the relation is positive.

The explained variance of each model is reported in table 6.1. It shows that the models of the time in system, the production time, the international transport time and the wholesales time have high explained variance. The models of the transfer time and the demand side time have relatively low explained variance. This is possible due to the fact that these outcomes are effected by constraints coded in datasets. Because of this, these constraints were not considered in the regression analysis.

The regression model of the time in system has a relatively high explained variance. The heatmap (figure 6.7) shows that the time in system is associated positively by the number of suppliers. On the contrary, the number of manufacturers, export ports, transit port, import port and wholesalers are associated with a lower time in system. Accordingly, addition of extra supply chain actors reduces the time in system, except for suppliers. Furthermore, the heatmap shows that the addition of extra links increases the time in system. Possibly due to more inefficient routes in such models.

A higher number of export ports and manufacturers is associated with a shorter production time. The more suppliers a model has, the more products are created in the model. An overflow of products might lead to congestion. The more manufacturers, and exports ports a model has, the less time it takes to process and ship a product from a supplier to an export port.

The wholesales time correlates positively with the number of import ports. The more import ports there are, the more products are delivered to the wholesaler. Addition of extra wholesalers reduces the wholesales time, because there is more capacity to process products.

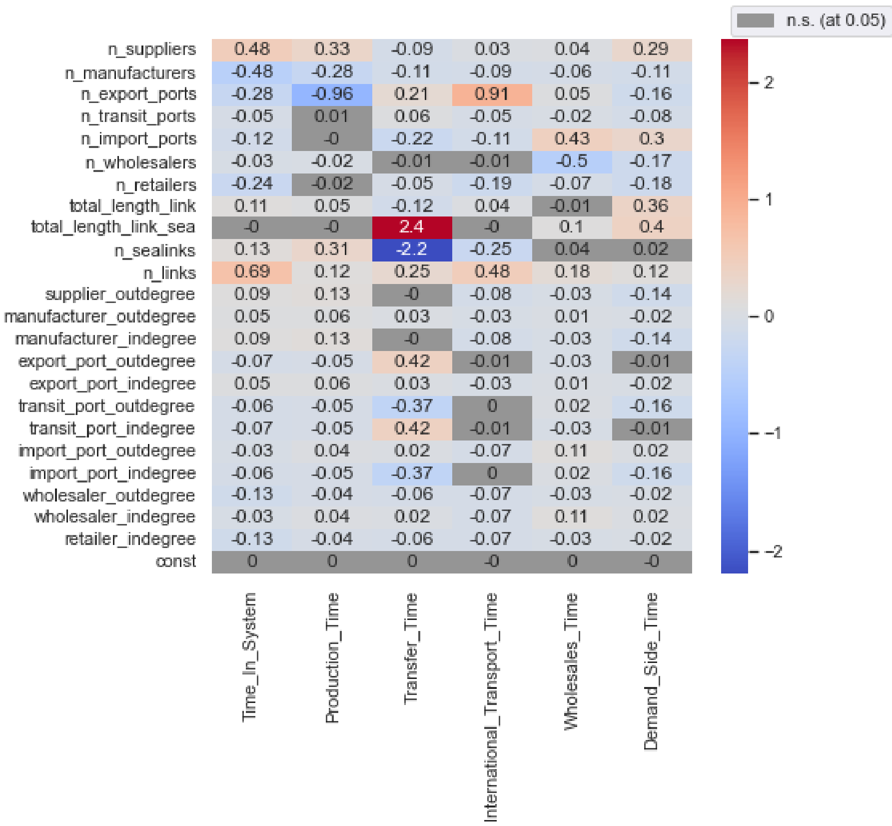

Figure 6.7: Standardized regression coefficients, coloured by size

Table 6.1: Explained variance of each regression model

Regression model	Dependent variable	Explained variance (r^2)
Model 0	Time in system	0.571
Model 1	Production time	0.803
Model 2	Transfer time	0.234
Model 3	International transport time	0.847
Model 4	Wholesales time	0.517
Model 5	Demand side time	0.269

7
Discussion

In this chapter, a discussion is presented. First, section 7.1 discusses the results. Thereafter, section 7.2 highlights the generalizability of this thesis. Afterwards, section 7.3 introduces the limitations of this study.

7.1. Discussion of results

The aim of this study is to efficaciously account for structural uncertainty in supply chain simulation models using model driven exploratory modelling. Exploratory modelling is often applied in studies that deal with a large extent of epistemic uncertainties. In this thesis, an illicit supply chain was modelled using exploratory modelling. Exploratory modelling helps to model such uncertainties.

Understanding structural uncertainty, starts by strictly defining when a model is deemed valid. Traditionally, a model is deemed valid whenever it perfectly matches empirical data. If the model matches the empirical data it is deemed valid, and otherwise it is 'false'. Only a single model can be valid at any time.

In many cases, it is impossible to establish such a true model. Systems are too complex to understand, or too few empirical data is available, to formulate a perfect model (Winsberg, 2010). An alternative to the traditional positivistic view of validation is integrative pluralism. In integrative pluralism multiple models of the same phenomena can be valid, while having conflicting assumptions (Mitchell, 2009). When multiple valid models of the same phenomena can be created, one faces structural uncertainty.

Exploratory modelling is a modelling approach, that departs from a family of models, rather than a single model. Many studies that use exploratory modelling to account for uncertainty, focus on parametric uncertainty rather structural uncertainty (Halim et al., 2016; Moallemi & Köhler, 2019). Models are build and validated, and afterwards its parameters are varied to map the effects of uncertainty. The model structure is assumed fixed and is not subject to discussion. Previous studies thus fail to account for structural uncertainty.

Not accounting for structural uncertainty can be problematic, especially when models are hard to validate. If the model is not a valid representation of the target system, varying its parameters is meaningless (Pilkey & Pilkey-Jarvis, 2007). This thesis shows that by changing the model building process, this pitfall can be overcome. By using model composability a plethora of plausible model structures can be generated. This enables the modeller to see the target system from a variety of perspectives, rather than a single representation. In this thesis, model composability is applied to generate a set of simulation models of an illicit supply chain. Standard model components are coupled in a dissimilar fashion to generate several model structures. The developed models all have a different configuration to represent structural uncertainty. Models generated by the model composer differ in number of actors, routes, and locations.

The results of this thesis show that changing the model building process helps in accounting for structural uncertainty. By employing model composability, several model components are combined in different combinations. This enables the modeller to create models with different model structures, without manually specifying each model. This is a pragmatic advantage over other approaches that compare several models of different authors, such as Li et al. (2017).

Important patterns and results can be missed when studying a single representation of the target system. This study proves this, because the ground truth (a single model structure) differentiates largely from the model sets generated by model composer, with respect to all studied simulation outcomes. The uncertainty bandwidths are significantly different. Additionally, this thesis shows that changing the model structure results in uncertainty bandwidths without varying parameters.

Furthermore, this study highlights the importance of a modellers' perception on the target system. In this study, a modellers' perception is operationalized by generating models with different constraints sets. Constraint sets define 'the degree of freedom' of the model composer to generate simulation models. The results show that by employing a different perspective on the illicit supply chain, different distributions of simulation outcomes are found. These results suggest that modellers should be aware of the influence of their perspective of the simulation results. Model composability can help modellers to record such effects of different perspectives.

The ability of generating models from several perspectives can help to stimulate debate in the policy arena. Model composability can potentially help involved actors to generate models in line with their own perspective. This facilitates communication, and makes models more useful in decision-making.

7.2. Generalizability

This research focussed on accounting for structural uncertainty in simulation models of illicit supply chains. The focus is on illicit supply chains, because decision makers face a substantial amount of structural uncertainty when regulating these types of supply chains. Nevertheless, the results also hold for similar legal supply chains with a substantial amount of uncertainty. For example, legal supply chains of coffee, cocoa or tobacco. The results hold, considering the fact that the model represents the physical flow of the supply chain. The physical flow of an illicit supply chain is relatively similar to a legal supply chain (Basu, 2013). The models do not contain any flow of information, and are not considered with any economic transactions.

The models generated in this study are generated specifically for a supply chain that is situated between Vietnam and the west of Europe. The results are reported specifically for this illicit supply chain. Despite that, the constraints of the model composer can easily be changed to model a supply chain that is situated elsewhere. The model composer can thus easily be applied to other uncertain illicit supply chains.

The results cannot be generalized to all types of supply chains. All models in this thesis represent single product supply chains, rather than complex multi product supply chains. Complex multi product supply chains might be modelled similarly, however the conclusion drawn differ.

7.3. Limitations

The results might be affected by five limitations: (1) the limited number of models generated, (2) the ontology used, (3) the assumptions of the constraints, (4) the simplicity of the individual simulation components, and (5) the experimental setup.

The first limitation of this study is the limited number of simulation models that are generated. Per perspective (constraint set) 700 models are generated. These are not all models that could have been generated by the model composer. It is therefore possible that some useful representations were not generated by the model composer. However, this study used Shannon's entropy to make sure that the generated models are as diverse as possible.

The second limitation of this study is related to the ontology used (the SES), which specifies which types of components are present in the model structure. Not every modeller creates the same SES, even when they try to model the same phenomena. The SES used in this research has seven types of supply chain actors, because the author believes these seven are of most relevance. However, the specification of these seven types of supply chain actors might be too limited or advanced. Other modellers might specify different components, or components with different specification relations. For instance, one might believe that the system has three types of ports (export, transit and import), while someone else might believe that there is no relevant difference between these types of ports. However, the ontology used in this study is based on literature. Therefore, the risk of such disagreement is mitigated.

The third limitation is related to the constraints used by the model composer. Constraints can be specified more specific, or more 'advanced'. For instance, a buffer around a port to place a wholesaler could potentially result in a wholesaler being located closely to the port. Such a constraint can be improved by specifying a minimum and a maximum range. Another example is the fixed order of supply chain actors in a simulation model. The model composer links supply chain actors with others, by selecting one actor of the next tier. For example, a supplier is always followed by a manufacturer. As a consequence, supply chain actors cannot be skipped. Each model structure has a transit port, while in fact such a port might not always be needed in a transport network. Besides, transport is assumed to be by boat and truck.

The fourth limitation of this study is the focus on structural uncertainty, originating from the composition of individual simulation model components. The behaviour of the individual components, such as retailers and manufacturers is fixed. In supply chain systems, this is a reasonable assumption because the (structural) behaviour of the components is known and not subject to structural uncertainty. Furthermore, this thesis does not account for structural uncertainty originating from unpredictable events.

The fourth limitation of this study is that the model components used in this study are relatively simple. For instance, supply chain actors have unlimited storage capacity. There is no maximum number in the number of products that a supply chain actor can store in their storage. Furthermore, supply chain actors handle just one type of good.

The final limitation of this study are related to the experimental setup of this research. Model parameters are not changed due to computational constraints. This might be of relevance, because Van Zelm and Huijbregts (2013) show that there is interaction between the complexity of the model structure and the size of parametric uncertainty. For instance, an increase in the number of raw goods produced by the suppliers, will lead to more congestion in the rest of the system regardless of the model structure. This is because high numbers of products will always lead to queues at other supply chain actors. However, having one set of parameters enables to focus on structural uncertainty.

8
Conclusion

The goal of this research was to efficaciously account for structural uncertainty in supply chain simulation models using model driven exploratory modelling. To accomplish this goal and answer the main question, four sub questions were formulated. In the following paragraphs, these four questions are answered one by one.

How to conceptualize structural uncertainty in the context of supply chain simulation models?
How structural uncertainty can be conceptualized largely depends on the modellers view of model validation. This is important because it determines the number of models of a phenomenon that can be valid at the same time. A modellers view on validation depends on the research philosophy they use.

From a traditional positivistic view, structural uncertainty does not exist. A model is either false or true, and therefore there cannot be any uncertainty in the structure of the model. The model is simply false if its structure is uncertain. Another view is compatible pluralism. Compatible pluralism allows for multiple models at the same time, however their assumptions should be compatible. Rival explanations are not allowed. From this view, structural uncertainty exists, but only complementary assumptions are allowed.

This study advocates integrative pluralism as research philosophy. This research philosophy allows having multiple models with conflicting assumptions of the same phenomenon. This research philosophy allows for a diversity of perspectives.

What model-driven exploratory modelling approaches are used to account for structural uncertainty in simulation models used in other fields?
Model-driven exploratory modelling encompasses a diversity of approaches. Approaches range from multi-resolution modelling to the usage of parameter sampling techniques, machine learning algorithms and optimization algorithms. In this study, the relevance of model composability to the modelling of structural uncertainty is argued. Utilizing DEVS, it was discussed how a variety of model structures can be generated using fixed model components. Furthermore, it was shown how the theory of model coupling in DEVS can be brought to practice, by describing its relation to the SES and by describing the process of pruning.

How can model-driven exploratory modelling be used to account for structural uncertainty in the context of supply chain simulation models?
This study has demonstrated how structural uncertainty of an illicit supply chain can be modelled using model composability, which is a specific form of model-driven exploratory modelling. A family of models can be generated by defining a standard set of model components, a set of constraints and a model composer. The model composer couples the standard model components, while complying to the constraints. Constraints include the order of coupling, the number of components per type, and the available locations. Model components were tiny submodels of suppliers, manufacturers, ports and other supply chain actors.

How does a change in perception of the target system change the method's efficacy?
A change in perspective of the target system changes the assumptions of the model composer. It changes the constraints that the model composer uses to generate models. Using a particular set of constraints, a set of models can be generated and simulated. These models combined show a particular distribution of a simulation outcome. This thesis shows that these distributions differ when different constraint sets are used to generate the models. The perspective from which a simulation model is generated, matters to the distribution of simulation outcomes.

Some assumptions had a larger effect on the simulation results than others. When the supply chain is assumed to have a higher number of manufacturers, ports, wholesalers and retailers, the time in system is shorter. If the system's number of suppliers is increased, the time in system becomes longer. Assumptions such as the indegree and outdegree of supply chain actor had a demonstrable relation with the simulation results, but their relations were smaller.

Concludingly, this study has shown that model composability, a form of model driven exploratory modelling, can be utilized to account for structural uncertainty in supply chain simulation models. By coupling standardized model components in different configurations, several model structures can be created. This study shows the international transport time, the production time and the time in system are affected most significantly by the structural uncertainty. In the case of illicit supply chains, the number of suppliers corresponds strongly to time in system, international transport time and the production time. A different perception on the model structural elements affects the outcomes of the simulation model.

Four recommendations for further research are specified. First, model composability could be used to account for structural uncertainty in other fields. The technique can be useful in modelling any type of system with fixed types of components, but an uncertain composition of them. For example, the technique can be useful in modelling water pipes networks, gas pipes networks, electricity networks, telecom networks. Especially when the data quality about these systems are bad, the technique can help to model several possible networks of the networks. For instance, the technique can possibly help in modelling water pipes system with an uncertain location of the pipes and pump stations. It is interesting to see if similar results can be achieved in these fields.

Besides that, further research could also focus on changing the behaviour of individual components, while varying the composition of these components. For instance, multiple variants of a wholesaler could be introduced, each having different behaviour. One with fixed capacity, one solely based a processing time, and another based on an advanced sub-model of a distribution centre. These variants can also be created by a dedicated model composer. Furthermore, it is also interesting to see if the structure of the simulation model can be varied over time. For example, introducing transit ports at certain times in the simulation, or removing or adding links during the simulation time.

Third, further research can also focus on the robustness of several policies given an uncertain illicit supply chain structure. For example, it is useful to try and test which policies are effective given an uncertain configuration of the illicit supply chain. To do this, both the parameters and the structure should be varied.

Finally, further research could also aim to discover the most optimal supply chain structure. Because the model composer generates several models, further research could focus on finding the most optimal supply chain structure in terms of processing and production time. Several optimization algorithms can be tested to see which supply chain is the most effective. It would also be interesting to see if the constraints of the model composer can be changed by the optimization algorithm to generate more optimal supply chain structures.

References

Alizadeh Noughabi, H. (2015). Entropy Estimation Using Numerical Methods. *Annals of Data Science*, *2*(2), 231–241. https://doi.org/10.1007/S40745-015-0045-9

Andreas Tolk. (2013). Ontology, Epistemology, and Teleology for Modeling and Simulation: Philosophical Foundations for Intelligent M&S Applications. In A. Tolk (Ed.), *Ontology, epistemology, and teleology for modeling and simulation* (pp. 1–26). Springer. https://doi.org/10.1007/978-3-642-31140-6

Baldissera Pacchetti, M. (2021). Structural uncertainty through the lens of model building. *Synthese*, *198*(11), 10377–10393. https://doi.org/10.1007/S11229-020-02727-8

Bankes, S. (1993). Exploratory Modeling for Policy Analysis. *Operations Research*, *41*(3), 435–449. https://doi.org/10.1287/OPRE.41.3.435

Bankes, S. (2011). The Use of Complexity for Policy Exploration. In P. Allen, S. Maguire, & B. McKelvey (Eds.), *The sage handbook of complexity and management* (pp. 590–603). SAGE Publications Inc. https://doi.org/10.4135/9781446201084.N34

Barlas, Y. (1996). Formal aspects of model validity and validation in system dynamics. *System Dynamics Review*, *12*(3), 183–209. https://doi.org/10.1002/(SICI)1099-1727(199623)12:3<183::AID-SDR103>3.0.CO;2-4

Barlas, Y., & Carpenter, S. (1990). Philosophical roots of model validation: Two paradigms. *System Dynamics Review*, *6*(2), 148–166. https://doi.org/10.1002/SDR.4260060203

Basu, G. (2013). The role of transnational smuggling operations in illicit supply chains. *Journal of Transportation Security*, *6*(4), 315–328. https://doi.org/10.1007/S12198-013-0118-Y

Basu, G. (2014). Concealment, corruption, and evasion: A transaction cost and case analysis of illicit supply chain activity. *Journal of Transportation Security*, *7*(3), 209–226. https://doi.org/10.1007/S12198-014-0140-8

Bittante, A., Pettersson, F., & Saxén, H. (2018). Optimization of a small-scale LNG supply chain. *Energy*, *148*, 79–89. https://doi.org/10.1016/J.ENERGY.2018.01.120

Bojke, L., Claxton, K., Sculpher, M., & Palmer, S. (2009). Characterizing Structural Uncertainty in Decision Analytic Models: A Review and Application of Methods. *Value in Health*, *12*(5), 739–749. https://doi.org/10.1111/J.1524-4733.2008.00502.X

Brynjarsdóttir, J., & O'hagan, A. (2014). Learning about physical parameters: the importance of model discrepancy. *Inverse Problems*, *30*(11), 114007. https://doi.org/10.1088/0266-5611/30/11/114007

Buchhorn, M., Smets, B., Bertels, L., Roo, B. D., Lesiv, M., Tsendbazar, N.-E., Herold, M., & Fritz, S. (2020). Copernicus Global Land Service: Land Cover 100m: collection 3: epoch 2019: Globe. https://doi.org/10.5281/ZENODO.3939050

Calatayud, A., Mangan, J., & Palacin, R. (2017). Connectivity to international markets: A multi-layered network approach. *Journal of Transport Geography*, *61*, 61–71. https://doi.org/10.1016/J.JTRANGEO.2017.04.006

Cardoso, S. R., Paula Barbosa-Póvoa, A., Relvas, S., & Novais, A. Q. (2015). Resilience metrics in the assessment of complex supply-chains performance operating under demand uncertainty. *Omega (United Kingdom)*, *56*, 53–73. https://doi.org/10.1016/J.OMEGA.2015.03.008

Cigolini, R., Pero, M., Rossi, T., & Sianesi, A. (2014). Linking supply chain configuration to supply chain performance: A discrete event simulation model. *Simulation Modelling Practice and Theory*, *40*, 1–11. https://doi.org/10.1016/J.SIMPAT.2013.08.002

Dam, K. H., Nikolic, I., & Lukszo, Z. (Eds.). (2013). *Agent-Based Modelling of Socio-Technical Systems* (Vol. 9). Springer Netherlands. https://doi.org/10.1007/978-94-007-4933-7

Davis, P. K., & Anderson, R. H. (2003). *Improving the Composability of Department of Defense Models and Simulations* (tech. rep.). Defense Technical Information Center. Santa Monica, CA. https://apps.dtic.mil/sti/citations/ADA451702

Ding, H., Benyoucef, L., & Xie, X. (2004). A simulation-based optimization method for production-distribution network design. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, *5*, 4521–4526. https://doi.org/10.1109/ICSMC.2004.1401244

Farrokh, M., Azar, A., Jandaghi, G., & Ahmadi, E. (2018). A novel robust fuzzy stochastic programming for closed loop supply chain network design under hybrid uncertainty. *Fuzzy Sets and Systems*, *341*, 69–91. https://doi.org/10.1016/J.FSS.2017.03.019

Folkerts, H., Pawletta, T., Deatcu, C., & Zeigler, B. P. (2020). Automated, Reactive Pruning of System Entity Structures for Simulation Engineering. *Proceedings of the 2020 Spring Simulation Conference, SpringSim 2020*. https://doi.org/10.22360/SPRINGSIM.2020.MOD4SIM.001

Frigg, R., Bradley, S., Du, H., & Smith, L. A. (2014). Laplace's Demon and the Adventures of His Apprentices. *Philosophy of Science*, *81*(1), 31–59. https://doi.org/10.1086/674416

Fumarola, M., Seck, M., & Verbraeck, A. (2010). A DEVS component library for simulation-based design of automated container terminals. *SIMUTools 2010 - 3rd International ICST Conference on Simulation Tools and Techniques*. https://doi.org/10.4108/ICST.SIMUTOOLS2010.8670

Gargalo, C. L., Carvalho, A., Gernaey, K. V., & Sin, G. (2017). Supply Chain Optimization of Integrated Glycerol Biorefinery: GlyThink Model Development and Application. *Industrial and Engineering Chemistry Research*, *56*(23), 6711–6727. https://doi.org/10.1021/acs.iecr.7b00908

Govindan, K., Soleimani, H., & Kannan, D. (2015). Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *European Journal of Operational Research*, *240*(3), 603–626. https://doi.org/10.1016/J.EJOR.2014.07.012

Grainger, M. J., Aramyan, L., Piras, S., Quested, T. E., Righi, S., Setti, M., Vittuari, M., & Stewart, G. B. (2018). Model selection and averaging in the assessment of the drivers of household food waste to reduce the probability of false positives. *PLoS ONE*, *13*(2). https://doi.org/10.1371/JOURNAL.PONE.0192075

Gruchmann, T., Eiten, J., De La Torre, G., & Melkonyan, A. (2019). Integrating optimization and simulation analysis to enhance strategic supply chain decision-making. In A. Melkonyan & K. Krumme (Eds.), *Sustainable logistics and transportation systems* (pp. 265–279). Springer International Publishing. https://doi.org/10.1007/978-3-319-98467-4_12

Gupta, A., & Govindaraju, R. S. (2019). Propagation of structural uncertainty in watershed hydrologic models. *Journal of Hydrology*, *575*, 66–81. https://doi.org/10.1016/J.JHYDROL.2019.05.026

Halim, R. A., Kwakkel, J. H., & Tavasszy, L. A. (2016). A scenario discovery study of the impact of uncertainties in the global container transport system on European ports. *Futures*, *81*, 148–160. https://doi.org/10.1016/J.FUTURES.2015.09.004

Hofmann, M. (2013). Ontologies in Modeling and Simulation: An Epistemological Perspective. In A. Tolk (Ed.), *Ontology, epistemology, and teleology for modeling and simulation* (pp. 59–87). Springer. https://doi.org/10.1007/978-3-642-31140-6

Jabarzare, Z., Zolfagharinia, H., & Najafi, M. (2020). Dynamic interdiction networks with applications in illicit supply chains. *Omega*, *96*, 102069. https://doi.org/10.1016/J.OMEGA.2019.05.005

Jackson, C. H., Bojke, L., Thompson, S. G., Claxton, K., & Sharples, L. D. (2011). A framework for addressing structural uncertainty in decision models. *Medical Decision Making*, *31*(4), 662–674. https://doi.org/10.1177/0272989X11406986

Jiao, Y., Yeophantong, P., & Lee, T. M. (2021). Strengthening International Legal Cooperation to Combat the Illegal Wildlife Trade Between Southeast Asia and China. *Frontiers in Ecology and Evolution*, *9*, 105. https://doi.org/10.3389/FEVO.2021.645427

Kaplan, A. (1964). Chapter VII: Models. *In the conduct of inquiry: Methodology for behavioral science* (pp. 258–291). CA: Chandler.

Keller, N., & Hu, X. (2019). Towards data-driven simulation modeling for mobile agent-based systems. *ACM Transactions on Modeling and Computer Simulation*, *29*(1). https://doi.org/10.1145/3289229

Khondoker, M., Dobson, R., Skirrow, C., Simmons, A., & Stahl, D. (2016). A comparison of machine learning methods for classification using simulation with multiple real data examples from mental health studies. *Statistical Methods in Medical Research*, *25*(5), 1804–1823. https://doi.org/10.1177/0962280213502437

Kiureghian, A. D., & Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, *31*(2), 105–112. https://doi.org/10.1016/J.STRUSAFE.2008.06.020

Kwakkel, J. H. (2017). The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. *Environmental Modelling & Software*, *96*, 239–250. https://doi.org/10.1016/J.ENVSOFT.2017.06.054

Lempert, R. J., Popper, S. W., & Bankes, S. C. (2003). *Shaping the next one hundred years : new methods for quantitative, long-term policy analysis and bibliography*. RAND. https://www.worldcat.org/title/shaping-the-next-one-hundred-years-new-methods-for-quantitative-long-term-policy-analysis-and-bibliography/oclc/559502047

Li, S. L., Bjørnstad, O. N., Ferrari, M. J., Mummah, R., Runge, M. C., Fonnesbeck, C. J., Tildesley, M. J., Probert, W. J., & Shea, K. (2017). Essential information: Uncertainty and optimal control of Ebola outbreaks. *Proceedings of the National Academy of Sciences*, *114*(22), 5659–5664. https://doi.org/10.1073/PNAS.1617482114

Lin, J., & Ban, Y. (2013). Complex Network Topology of Transportation Systems. *Transport Reviews*, *33*(6), 658–685. https://doi.org/10.1080/01441647.2013.848955

Min, H., & Zhou, G. (2002). Supply chain modeling: past, present and future. *Computers & Industrial Engineering*, *43*(1-2), 231–249. https://doi.org/10.1016/S0360-8352(02)00066-9

Mitchell, S. D. (2009). *Unsimple Truths*. The university of Chicago Press.

Moallemi, E. A., & Köhler, J. (2019). Coping with uncertainties of sustainability transitions using exploratory modelling: The case of the MATISSE model and the UK's mobility sector. *Environmental Innovation and Societal Transitions*, *33*, 61–83. https://doi.org/10.1016/J.EIST.2019.03.005

Nance, R. E. (1981). The time and state relationships in simulation modeling. *Communications of the ACM*, *24*(4), 173–179. https://doi.org/10.1145/358598.358601

OECD. (2009). Magnitude of counterfeiting and piracy of tangible products – November 2009 update. https://www.oecd.org/sti/ind/magnitudeofcounterfeitingandpiracyoftangibleproductsnovember2009update.htm

Pacchetti, M. B. (2018). *Spatiotemporal scales in modeling: identyfing target systems* (Doctoral dissertation). University of Pittsburgh. http://d-scholarship.pitt.edu/34707/1/BaldisseraPacchetti_ETD_final_1.pdf

Page, S. E. (2018). *The Model Thinker* (1st ed.). Basic Books.

Parker, W. S. (2006). Understanding pluralism in climate modeling. *Foundations of Science*, *11*(4), 349–368. https://doi.org/10.1007/S10699-005-3196-X

Pilkey, O. H., & Pilkey-Jarvis, L. (2007). *Useless arithmetic : why environmental scientists can't predict the future*. Columbia University Press.

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3). https://doi.org/10.1002/WIDM.1301

Refsgaard, J. C., van der Sluijs, J. P., Brown, J., & van der Keur, P. (2006). A framework for dealing with uncertainty due to model structure error. *Advances in Water Resources*, *29*(11), 1586–1597. https://doi.org/10.1016/J.ADVWATRES.2005.11.013

Rodriguez, B., & Yilmaz, L. (2020). Learning Rule-Based Explanatory Models from Exploratory Multi-Simulation for Decision-Support under Uncertainty. *Proceedings - Winter Simulation Conference*, *2020-December*, 2293–2304. https://doi.org/10.1109/WSC48552.2020.9383858

Rosen, R. (1991). *Life itself : a comprehensive inquiry into the nature, origin, and fabrication of life*. Columbia University Press.

Salt, J. D. (2008). The seven habits of highly defective simulation projects. *Journal of Simulation 2008 2:3*, *2*(3), 155–161. https://doi.org/10.1057/JOS.2008.7

Sarjoughian, H. S. (2006). Model composability. *Proceedings - Winter Simulation Conference*, 149–158. https://doi.org/10.1109/WSC.2006.323047

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*(1), 3–55. https://doi.org/10.1145/584091.584093

Staake, T., Thiesse, F., & Fleisch, E. (2009). The emergence of counterfeit trade: A literature review. *European Journal of Marketing*, *43*(3-4), 320–349. https://doi.org/10.1108/03090560910935451

Statista. (2021). Global seaborne trade of crude oil 2020. https://www-statista-com.tudelft.idm.oclc.org/statistics/264013/transport-volume-of-crude-oil-in-seaborne-trade/

Tombido, L., Louw, L., & van Eeden, J. (2020). The Bullwhip Effect in Closed-Loop Supply Chains: A Comparison of Series and Divergent Networks. *Journal of Remanufacturing*, *10*(3), 207–238. https://doi.org/10.1007/S13243-020-00085-9

UNCTADstat. (n.d.). Container port throughput, annual. http://unctadstat.unctad.org/wds/TableViewer/tableView.aspx?ReportId=13321

Van Zelm, R., & Huijbregts, M. A. (2013). Quantifying the trade-off between parameter and model structure uncertainty in life cycle impact assessment. *Environmental Science and Technology*, *47*(16), 9274–9280. https://doi.org/10.1021/ES305107S

Vangheluwe, H. (2002). DEVS as a common denominator for multi-formalism hybrid systems modelling, 129–134. https://doi.org/10.1109/CACSD.2000.900199

Vautard, R., Gobiet, A., Jacob, D., Belda, M., Colette, A., Déqué, M., Fernández, J., García-Díez, M., Goergen, K., Güttler, I., Halenka, T., Karacostas, T., Katragkou, E., Keuler, K., Kotlarski, S., Mayer, S., van Meijgaard, E., Nikulin, G., Patarčić, M., … Yiou, P. (2013). The simulation of European heat waves from an ensemble of regional climate models within the EURO-CORDEX project. *Climate Dynamics*, *41*(9-10), 2555–2575. https://doi.org/10.1007/s00382-013-1714-z

Vilko, J., Ritala, P., & Edelmann, J. (2014). On uncertainty in supply chain risk management. *International Journal of Logistics Management*, *25*(1), 3–19. https://doi.org/10.1108/IJLM-10-2012-0126

Walker, W. E., Harremoës, P., Rotmans, J., Sluijs, J. v. d., Asselt, M. v., Janssen, P., & Krauss, M. K. v. (2010). Defining Uncertainty: A Conceptual Basis for Uncertainty Management in Model-Based Decision Support. *Integrated Assessment*, *4*(1), 5–17. https://doi.org/10.1076/IAIJ.4.1.5.16466

Walker, W. E., Marchau, V. A., & Kwakkel, J. H. (2013). Uncertainty in the Framework of Policy Analysis. *International Series in Operations Research and Management Science*, *179*, 215–261. https://doi.org/10.1007/978-1-4614-4602-6_9

Wang, J., Mo, H., Wang, F., & Jin, F. (2011). Exploring the network structure and nodal centrality of China's air transport network: A complex network approach. *Journal of Transport Geography*, *19*(4), 712–721. https://doi.org/10.1016/J.JTRANGEO.2010.08.012

Warmink, J. J., Janssen, J. A., Booij, M. J., & Krol, M. S. (2010). Identification and classification of uncertainties in the application of environmental models. *Environmental Modelling & Software*, *25*(12), 1518–1527. https://doi.org/10.1016/J.ENVSOFT.2010.04.011

Webster, M., Sokolov, A., & Joint, M. (1998). Quantifying the uncertainty in climate predictions. https://dspace.mit.edu/handle/1721.1/3610

Winsberg, E. (2010). *Models of Climate: Values and Uncertainties*. University of Chicago Press.

Yilmaz, L. (2019). Toward Self-Aware Models as Cognitive Adaptive Instruments for Social and Behavioral Modeling. In P. K. Davis, A. O'Mahony, & J. Pfautz (Eds.), *Social-behavioral modeling for complex systems* (pp. 569–586). John Wiley & Sons Inc. https://doi.org/10.1002/9781119485001

Zeigler, B. P., & Hammonds, P. E. (2007). Modeling & simulation-based data engineering : introducing pragmatics into ontologies for net-centric information exchange, 433.

Zeigler, B. P., Muzy, A., & Kofman, E. (2019a). Basic Formalisms: Coupled Multi-component Systems. *Theory of modeling and simulation: Discrete event & iterative system computational foundations* (3rd ed., pp. 167–194). Elsevier. https://doi.org/10.1016/C2016-0-03987-6

Zeigler, B. P., Muzy, A., & Kofman, E. (2019b). Introduction To Discrete Event System Specification (DEVS). *Theory of modeling and simulation: Discrete event & iterative system computational foundations* (3rd ed., pp. 93–125). Elsevier. https://doi.org/10.1016/C2016-0-03987-6

A

Constraints

This appendix lists all the constraints that were taken into account by the model composer. Table A.1 provides an overview of the constraints in the number of nodes (supply chain actors). Table A.2 shows the limitations on the indegree and outdegree of nodes. Table A.3 shows the predecessor and the succesors of the supply chain entities. Supply chain actors can only be succeeded by the successor, and preceded by their predecessor. Table A.4 shows the datasets used as constraints. The sources of these datasets are listed in table A.6. Table A.5 shows the buffer sizes used as constraints. Table A.1, table A.2, table A.3, table A.4, and table A.5 contain the base set of constraints. The datasets listed in table A.4 are visualized in figure A.1 to A.7.

Table A.7 shows the values that are different in the constraint sets used in the experiments. The changed datasets in these constraints sets are shown in figure A.8 and figure A.9. Figure A.8 shows the transit ports used in constraint set 2. This dataset was created by the author, based on discussions with experts and inspection of Google Maps. Figure A.9 shows the area in which retailers and wholesalers may be placed in constraint set 3.

Table A.1: Constraints in number of nodes

Constraint	Variable name	Value
Number of suppliers	n_suppliers	5
Number of manufacturers	n_manufacturers	5
Number of export ports	n_export_ports	2
Number of transit ports	n_transit_ports	2
Number of import ports	n_import_ports	2
Number of wholesalers	n_wholesalers	3
Number of retailers	n_retailers	10

Table A.2: Constraints in degree

Node type	Variable name	Value (minimum, maximum)
Suppliers	indegree range	(0,0)
Suppliers	outdegree range	(1,3)
Manufacturers	indegree range	(1,3)
Manufacturers	outdegree range	(1,3)
Export ports	indegree range	(1,4)
Export ports	outdegree range	(1,3)
Transit ports	indegree range	(1,-1)
Transit ports	outdegree range	(1,-1)
Import ports	indegree range	(1,-1)
Import ports	outdegree range	(1,-1)
Wholesalers	indegree range	(1,-1)
Wholesalers	outdegree range	(1,-1)
Retailers	indegree range	(1,3)
Retailers	outdegree range	(-1,-1)

Note. -1 means that there is no restriction.

Table A.3: Predecessor and successor constraints

Node type	Variable name	Value
Supplier	predecessor	None
Supplier	successor	Manufacturer
Manufacturer	predecessor	Supplier
Manufacturer	successor	Export port
Export port	predecessor	Manufacturer
Export port	successor	Transit port
Transit port	predecessor	Export port
Transit port	successor	Import port
Import port	predecessor	Transit port
Import port	successor	Wholesaler
Wholesaler	predecessor	Import port
Wholesaler	successor	Retailer
Retailer	predecessor	Wholesaler
Retailer	successor	None

Table A.4: Datasets required

Dataset	Description
Ports suppliers side	Dataset with ports on the supplier side of the supply chain
Transit ports	Dataset with transit ports (same as ports supplier side)
Land use supplier side	Dataset with the land use of the supplier side area
Borders supplier side	Dataset with the administrative borders of the supplier side
Ports receiver side	Dataset with ports on the receiving side of the supply chain
Land use receiver side	Dataset with the land use of the receiving side of the supply chain
Borders receiver side	Dataset with the administrative borders of the receiving area of the supply chain
Sea dataset	Dataset with a binary classification of the world, distinction between land and water.

Table A.5: Buffer ranges

Buffer	Value
Supplier	80.000
Manufacturer	80.000
Wholesaler	80.000
Retailer	80.000
Increment	10.000

Table A.6: Dataset sources

Data set	Source
Landuse data sets	Buchhorn et al. (2020)
Vietnam ports data set	Open development Mekong (https://data.opendevelopmentmekong.net/dataset/bn-cng)
Dutch ports	own interpretation
Alternative transit ports	own interpretation based on satellite imagery and port lists
Administrative regions Europe	Nuts (https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/nuts)
Administrative region Vietnam	Open data soft (https://public.opendatasoft.com/explore/dataset/world-administrative-boundaries/export/)
Sea dataset	Buchhorn et al. (2020)

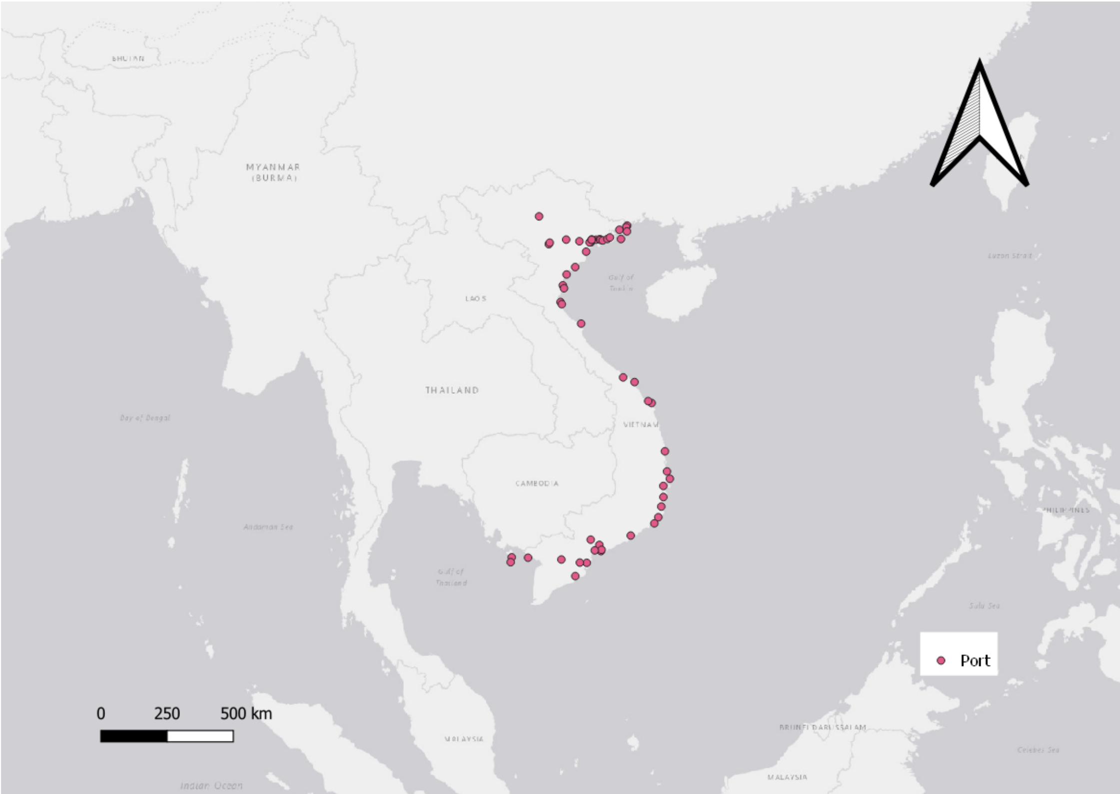

Figure A.1: Export and transit ports in the base set of constraints (Ports suppliers and Transit ports)

Figure A.2: Administrative boundaries used in Vietnam (Borders suppliers side)

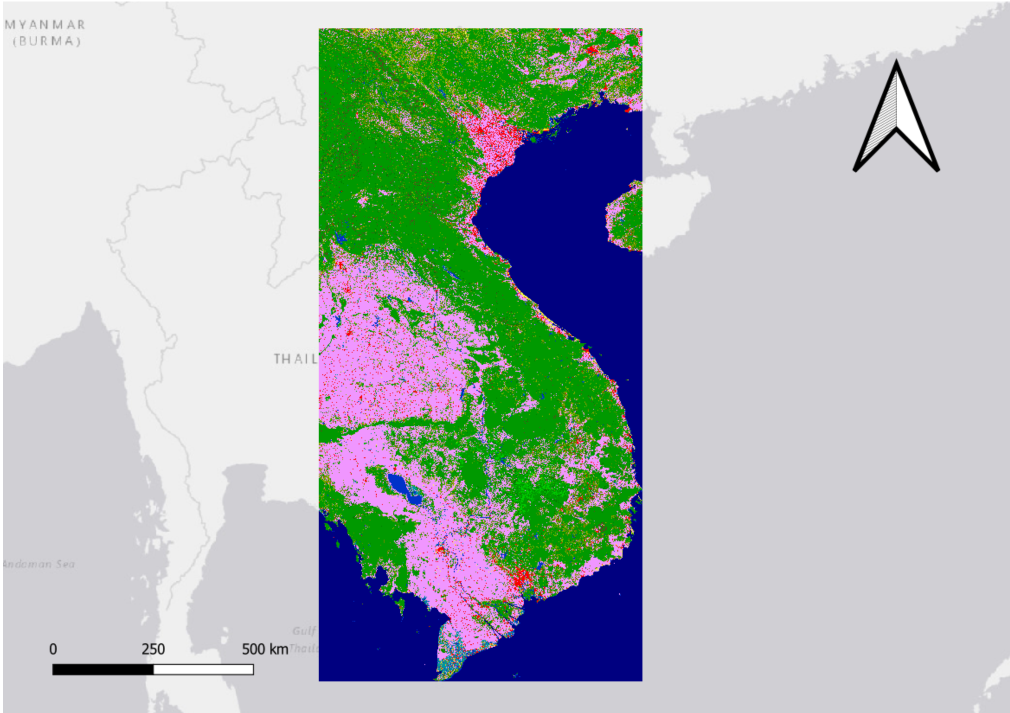

Figure A.3: Land-use Vietnam (Land use supplier side)

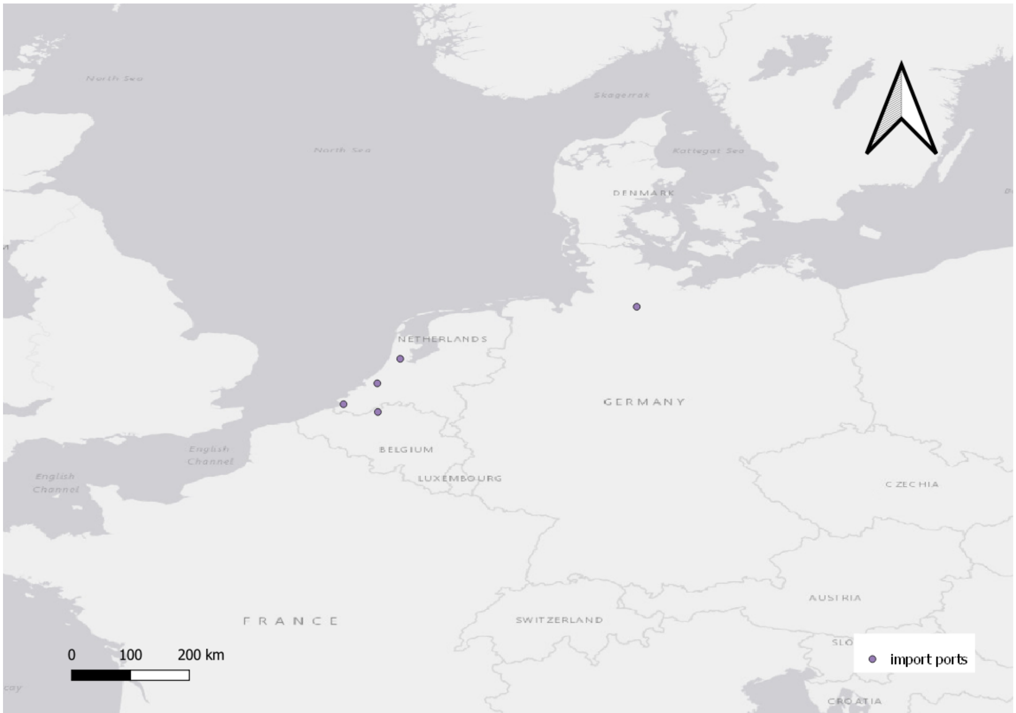

Figure A.4: Dutch ports of Vlissingen, Rotterdam and Amsterdam. Port of Hamburg (Germany) and Antwerpen (Belgium) (Ports receiver side)

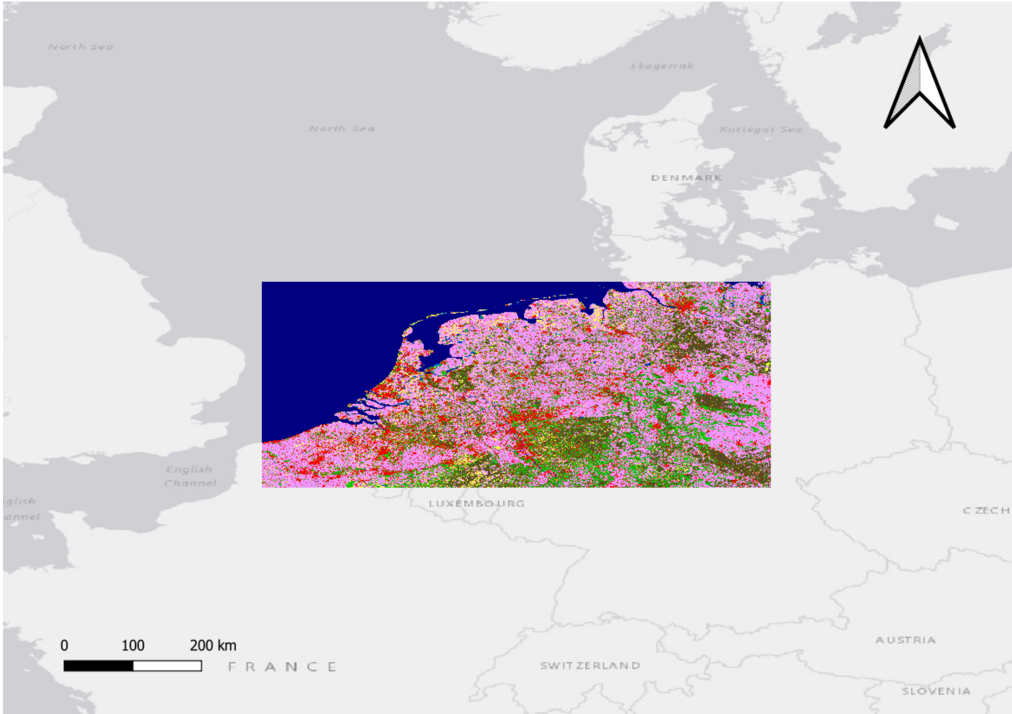

Figure A.5: Land use in the Benelux and Western Germany (Land use receiver side)

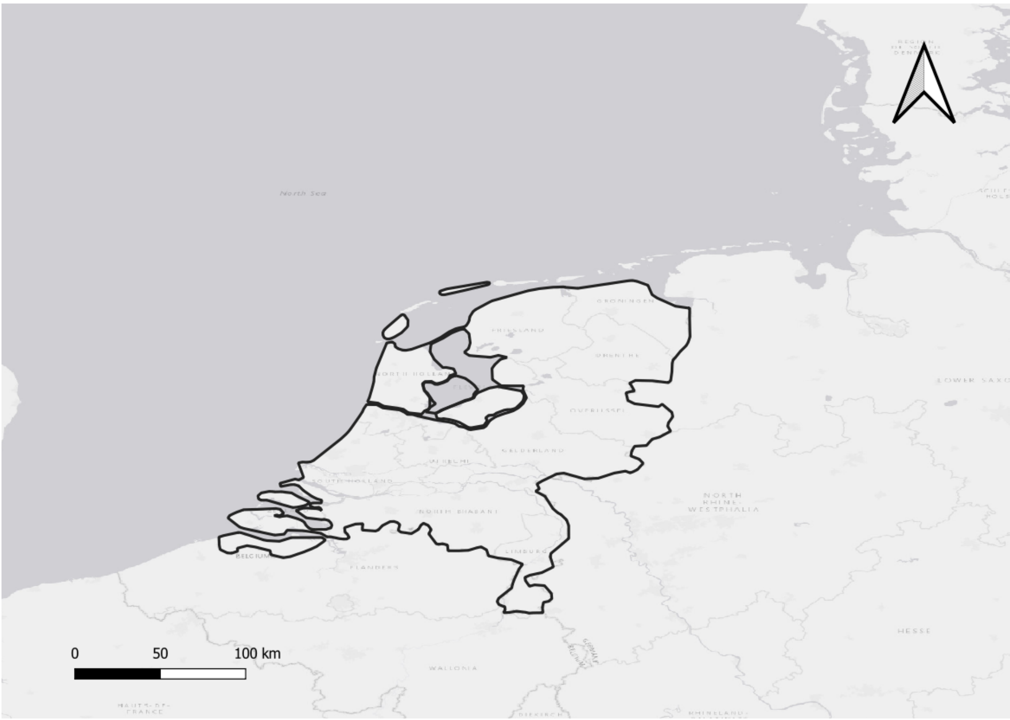

Figure A.6: Administrative boundaries of the Netherlands (Borders receiver side)

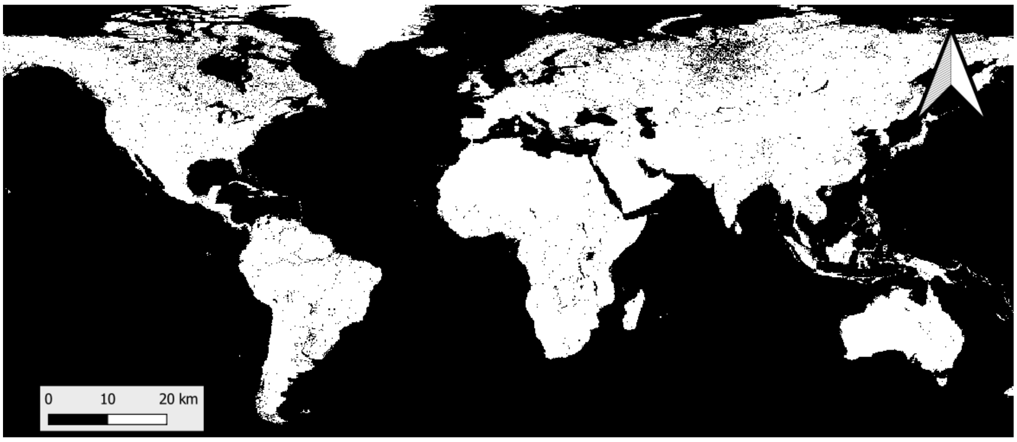

Figure A.7: Sea dataset (Sea data set)

Table A.7: Variation of constraint sets

Constraints set	Variable name	Value
Constraint set 1 (base)	no changes	-
Constraint set 2 (different transit ports)	Transit ports	Dataset with different ports
Constraint set 3 (larger retailer area)	Borders receiver side	Different administrative boundaries for locations wholesalers and retailers
Constraint set 3	buffer wholesaler	200.000
Constraint set 3	buffer retailer	200.000
Constraint set 4 (Larger retailer network)	n_wholesalers	3
Constraint set 4 (Larger retailer network)	n_retailers	20
Constraint set 5 (larger connected supplier network)	n_suppliers	15
Constraint set 5 (larger connected supplier network)	n_manufacturers	10

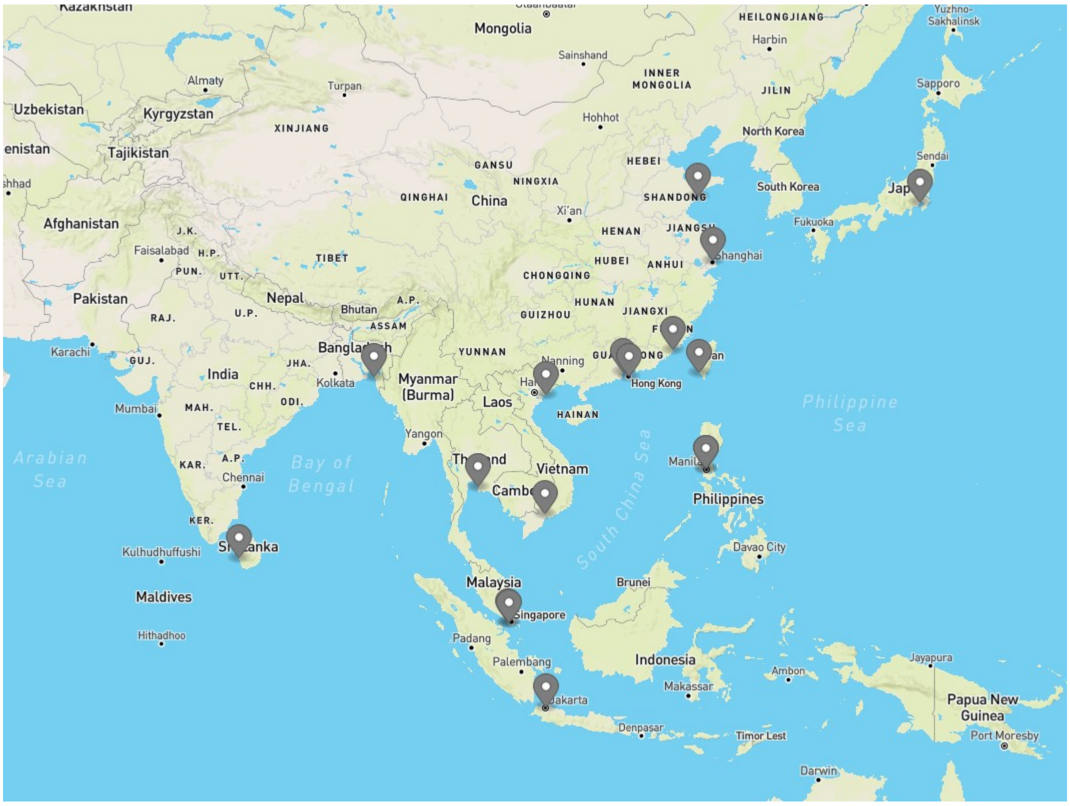

Figure A.8: Transit ports used in constraint set 2

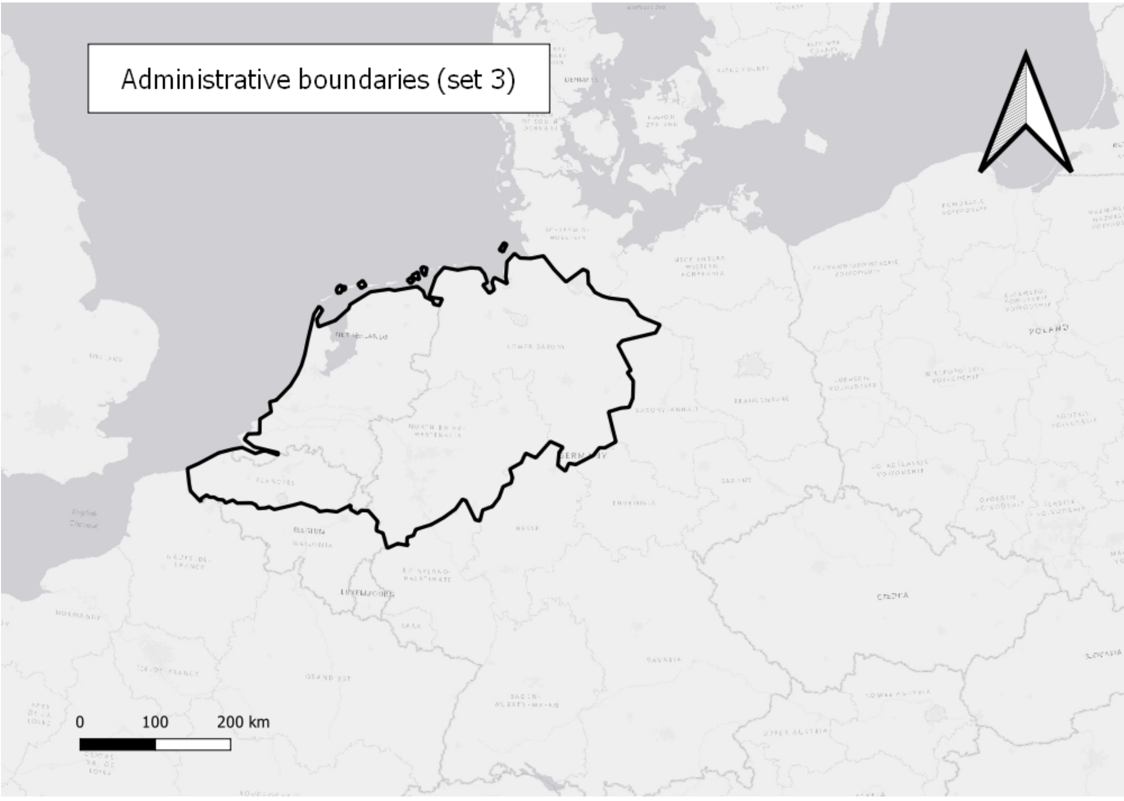

Figure A.9: Administrative boundaries used in constraint set 3 (Borders receiver side)

B

Unit tests: code validation

This appendix covers some more details on the python implementation of the model composer. Table B.1 shows all the composer tests that have been constructed to ensure that the code was implemented as intended. Table B.2 shows all the tests of the utilities used by the composer.

Table B.1: Composer tests

Test name	Function to test
test_create_instance	function that determines entities per entity type
test_validate_instance	function that checks if an instance (number of entities per type) is valid
test_validate_instance_search	function that searches for a valid combination of number of entities per type
test_create_ids	function that generates a unique id per entity
test_create_graph	function that creates a networkx graph from a combination of number of entities per type
test_create_graph2	function that test a specific case that was known to result in bugs
test_validate_graph	function that tests whether a graph complies to all in and out degree constraints
test_graph_set	function that generates a number of graphs
test_sea_distance	function that computes a sea route and its distance

Table B.2: Utility tests

Test name	Function to test
test_find_overlaps	function that checks if two ranges overlap or not. Used in validating in and out degrees.
test_generate_buffer_zone	function that generates a buffer zone
test_get_distance_points_ max	function that computes the distance between two points
test_get_distance_two_points_min	function that computes the distance between two points
test_get_distance_two_points_haversine	function that computes the distance between two point using the mathematical haversine approach.
test_get_urban_locations_tif	function that retrieves a location within a polygon classified as an urban area.

C
Pydsol visualization tool

This appendix is about the web app that is developed to visualize the pydsol simulation components. A web app is developed to assist in the development of the model composer. The web app helps to quickly visualize models. This helps to quickly visualize obvious mistakes. The web app shows the spatial configuration of a model. Figure C.1 shows the UI of the web application. The web app is developed in VUE.js 3 (released in February 2022), which is a code framework written in JavaScript. An experimental version of leaflet is used to visualize the model on a map.

The user can drag the map to a location of interest. Furthermore, the user can hover over a specific model component to view the components' id, name, and type. Besides, each type of model component has its own colour. Figure C.1 shows an example of a supply chain model generated using the model composer. In the figure, two blue circles indicate the transit ports in the model. The orange circles indicate the manufacturers in the model. The black circles indicate the export ports in this particular model.

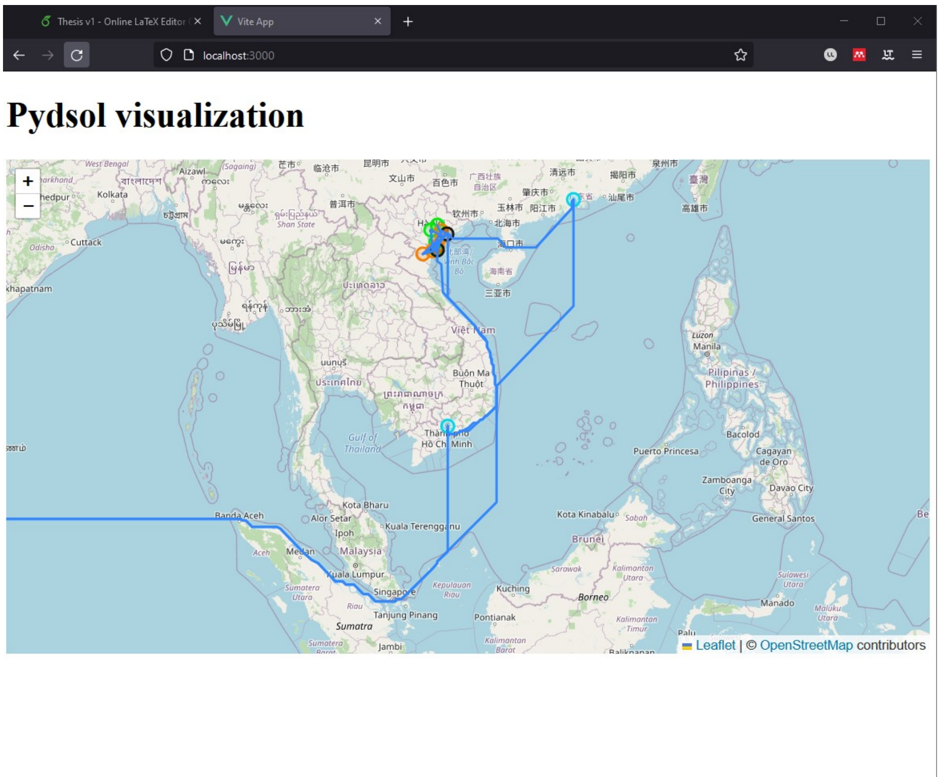

Figure C.1: Pydsol web app visualization utility

D

Regression analysis topological features

This appendix contains the outcomes of a regression analysis between the outcomes of the simulation models and the topological features of the models. Table D.1 shows the statistics of this regression model. Table D.2 presents the model coefficients and their associated p-values.

Table D.1: Time in system statistics

Dep. Variable:	Time_In_System	**R-squared:**	0.391
Model:	OLS	**Adj. R-squared:**	0.391
Method:	Least Squares	**F-statistic:**	3851.
Date:	None	**Prob (F-statistic):**	0.00
Time:	15:32:54	**Log-Likelihood:**	-35130.
No. Observations:	30000	**AIC:**	7.027e+04
Df Residuals:	29994	**BIC:**	7.032e+04
Df Model:	5		
Covariance Type:	nonrobust		

Table D.2: Time in system statistics

	coef	**std err**	**t**	**P$>$ |t|**	**[0.025**	**0.975]**
edges	0.2144	0.020	10.805	0.000	0.176	0.253
nodes	0.1460	0.029	4.980	0.000	0.089	0.204
betweenness	0.4950	0.009	54.982	0.000	0.477	0.513
degree_centrality	-1.6177	0.022	-74.124	0.000	-1.660	-1.575
closeness_centrality	1.1153	0.016	70.259	0.000	1.084	1.146
const	-8.275e-16	0.005	-1.84e-13	1.000	-0.009	0.009

E
Regression analysis model structural elements

This appendix contains all regression analysis of the constraints (model structural elements) and the simulation models. In these regression analyses, all simulation results from all constraint sets are included. In each regression model, the dependent variable is the outcome of the simulation model.

Table E.1 and E.2 present the (standardized) regression coefficients and statistics of a regression model with the time in system as dependent variable. Tables E.3 and E.4 show the coefficients and statistics of the production time. Tables E.5 and E.6 show the coefficients and statistics of the transfer time. Tables E.7 and E.8 present the outcomes of the regression between the model structural elements and the international transport time. Tables E.9 and E.10 show the results of a model with the wholesales time as dependent variable. Finally, table E.11 and E.12 show the coefficients and statistics of a regression model with the demand side time as dependent variable.

Table E.1: Time in system statistics

Dep. Variable:	Time_In_System	**R-squared:**	0.571
Model:	OLS	**Adj. R-squared:**	0.570
Method:	Least Squares	**F-statistic:**	2344.
Date:	None	**Prob (F-statistic):**	0.00
Time:	12:07:03	**Log-Likelihood:**	-29885.
No. Observations:	30000	**AIC:**	5.981e+04
Df Residuals:	29982	**BIC:**	5.996e+04
Df Model:	17		
Covariance Type:	nonrobust		

Table E.2: Time in system coefficients

	coef	std err	t	P> |t|	[0.025	0.975]
n_suppliers	0.4822	0.009	56.561	0.000	0.465	0.499
n_manufacturers	-0.4789	0.011	-41.827	0.000	-0.501	-0.456
n_export_ports	-0.2753	0.008	-32.979	0.000	-0.292	-0.259
n_transit_ports	-0.0510	0.010	-5.176	0.000	-0.070	-0.032
n_import_ports	-0.1212	0.008	-15.716	0.000	-0.136	-0.106
n_wholesalers	-0.0326	0.008	-4.176	0.000	-0.048	-0.017
n_retailers	-0.2408	0.016	-15.481	0.000	-0.271	-0.210
total_length_link	0.1078	0.006	18.332	0.000	0.096	0.119
total_length_link_sea	-0.0043	0.019	-0.222	0.824	-0.042	0.033
n_sealinks	0.1320	0.034	3.918	0.000	0.066	0.198
n_links	0.6935	0.035	19.844	0.000	0.625	0.762
supplier_outdegree	0.0870	0.009	9.726	0.000	0.069	0.105
manufacturer_outdegree	0.0490	0.007	6.982	0.000	0.035	0.063
manufacturer_indegree	0.0870	0.009	9.726	0.000	0.069	0.105
export_port_outdegree	-0.0716	0.009	-8.098	0.000	-0.089	-0.054
export_port_indegree	0.0490	0.007	6.982	0.000	0.035	0.063
transit_port_outdegree	-0.0623	0.008	-7.743	0.000	-0.078	-0.047
transit_port_indegree	-0.0716	0.009	-8.098	0.000	-0.089	-0.054
import_port_outdegree	-0.0290	0.004	-7.022	0.000	-0.037	-0.021
import_port_indegree	-0.0623	0.008	-7.743	0.000	-0.078	-0.047
wholesaler_outdegree	-0.1257	0.006	-21.184	0.000	-0.137	-0.114
wholesaler_indegree	-0.0290	0.004	-7.022	0.000	-0.037	-0.021
retailer_indegree	-0.1257	0.006	-21.184	0.000	-0.137	-0.114
const	3.886e-16	0.004	1.03e-13	1.000	-0.007	0.007

Table E.3: Production time statistics

Dep. Variable:	Production_Time	**R-squared:**	0.803
Model:	OLS	**Adj. R-squared:**	0.803
Method:	Least Squares	**F-statistic:**	7172.
Date:	None	**Prob (F-statistic):**	0.00
Time:	12:07:04	**Log-Likelihood:**	-18228.
No. Observations:	30000	**AIC:**	3.649e+04
Df Residuals:	29982	**BIC:**	3.664e+04
Df Model:	17		
Covariance Type:	nonrobust		

Table E.4: Production time coefficients

	coef	std err	t	P> |t|	[0.025	0.975]
n_suppliers	0.3298	0.006	57.064	0.000	0.318	0.341
n_manufacturers	-0.2794	0.008	-35.989	0.000	-0.295	-0.264
n_export_ports	-0.9577	0.006	-169.205	0.000	-0.969	-0.947
n_transit_ports	0.0053	0.007	0.793	0.428	-0.008	0.018
n_import_ports	-0.0028	0.005	-0.540	0.589	-0.013	0.007
n_wholesalers	-0.0179	0.005	-3.383	0.001	-0.028	-0.008
n_retailers	-0.0198	0.011	-1.882	0.060	-0.041	0.001
total_length_link	0.0463	0.004	11.618	0.000	0.038	0.054
total_length_link_sea	-0.0009	0.013	-0.070	0.944	-0.026	0.025
n_sealinks	0.3059	0.023	13.387	0.000	0.261	0.351
n_links	0.1178	0.024	4.973	0.000	0.071	0.164
supplier_outdegree	0.1275	0.006	21.009	0.000	0.116	0.139
manufacturer_outdegree	0.0649	0.005	13.639	0.000	0.056	0.074
manufacturer_indegree	0.1275	0.006	21.009	0.000	0.116	0.139
export_port_outdegree	-0.0481	0.006	-8.036	0.000	-0.060	-0.036
export_port_indegree	0.0649	0.005	13.639	0.000	0.056	0.074
transit_port_outdegree	-0.0478	0.005	-8.758	0.000	-0.058	-0.037
transit_port_indegree	-0.0481	0.006	-8.036	0.000	-0.060	-0.036
import_port_outdegree	0.0388	0.003	13.857	0.000	0.033	0.044
import_port_indegree	-0.0478	0.005	-8.758	0.000	-0.058	-0.037
wholesaler_outdegree	-0.0394	0.004	-9.801	0.000	-0.047	-0.032
wholesaler_indegree	0.0388	0.003	13.857	0.000	0.033	0.044
retailer_indegree	-0.0394	0.004	-9.801	0.000	-0.047	-0.032
const	2.359e-16	0.003	9.2e-14	1.000	-0.005	0.005

Table E.5: Transfer time statistics

Dep. Variable:	Transfer_Time	**R-squared:**	0.234
Model:	OLS	**Adj. R-squared:**	0.233
Method:	Least Squares	**F-statistic:**	537.6
Date:	None	**Prob (F-statistic):**	0.00
Time:	12:07:06	**Log-Likelihood:**	-38577.
No. Observations:	30000	**AIC:**	7.719e+04
Df Residuals:	29982	**BIC:**	7.734e+04
Df Model:	17		
Covariance Type:	nonrobust		

Table E.6: Transfer time coefficients

	coef	std err	t	P> |t|	[0.025	0.975]
n_suppliers	-0.0902	0.011	-7.916	0.000	-0.112	-0.068
n_manufacturers	-0.1133	0.015	-7.405	0.000	-0.143	-0.083
n_export_ports	0.2058	0.011	18.456	0.000	0.184	0.228
n_transit_ports	0.0633	0.013	4.811	0.000	0.038	0.089
n_import_ports	-0.2164	0.010	-21.007	0.000	-0.237	-0.196
n_wholesalers	-0.0093	0.010	-0.889	0.374	-0.030	0.011
n_retailers	-0.0454	0.021	-2.183	0.029	-0.086	-0.005
total_length_link	-0.1222	0.008	-15.560	0.000	-0.138	-0.107
total_length_link_sea	2.3747	0.026	92.380	0.000	2.324	2.425
n_sealinks	-2.1998	0.045	-48.850	0.000	-2.288	-2.112
n_links	0.2473	0.047	5.297	0.000	0.156	0.339
supplier_outdegree	-8.167e-05	0.012	-0.007	0.995	-0.024	0.023
manufacturer_outdegree	0.0281	0.009	2.999	0.003	0.010	0.047
manufacturer_indegree	-8.167e-05	0.012	-0.007	0.995	-0.024	0.023
export_port_outdegree	0.4191	0.012	35.496	0.000	0.396	0.442
export_port_indegree	0.0281	0.009	2.999	0.003	0.010	0.047
transit_port_outdegree	-0.3659	0.011	-34.062	0.000	-0.387	-0.345
transit_port_indegree	0.4191	0.012	35.496	0.000	0.396	0.442
import_port_outdegree	0.0176	0.006	3.189	0.001	0.007	0.028
import_port_indegree	-0.3659	0.011	-34.062	0.000	-0.387	-0.345
wholesaler_outdegree	-0.0607	0.008	-7.658	0.000	-0.076	-0.045
wholesaler_indegree	0.0176	0.006	3.189	0.001	0.007	0.028
retailer_indegree	-0.0607	0.008	-7.658	0.000	-0.076	-0.045
const	1.346e-15	0.005	2.66e-13	1.000	-0.010	0.010

Table E.7: International transport time statistics

Dep. Variable:	International_Transport_Time	**R-squared:**	0.847
Model:	OLS	**Adj. R-squared:**	0.847
Method:	Least Squares	**F-statistic:**	9795.
Date:	None	**Prob (F-statistic):**	0.00
Time:	12:07:07	**Log-Likelihood:**	-14367.
No. Observations:	30000	**AIC:**	2.877e+04
Df Residuals:	29982	**BIC:**	2.892e+04
Df Model:	17		
Covariance Type:	nonrobust		

Table E.8: International transport time coefficients

	coef	std err	t	P> |t|	[0.025	0.975]
n_suppliers	0.0340	0.005	6.688	0.000	0.024	0.044
n_manufacturers	-0.0920	0.007	-13.479	0.000	-0.105	-0.079
n_export_ports	0.9104	0.005	182.937	0.000	0.901	0.920
n_transit_ports	-0.0521	0.006	-8.879	0.000	-0.064	-0.041
n_import_ports	-0.1056	0.005	-22.971	0.000	-0.115	-0.097
n_wholesalers	-0.0075	0.005	-1.618	0.106	-0.017	0.002
n_retailers	-0.1920	0.009	-20.705	0.000	-0.210	-0.174
total_length_link	0.0406	0.004	11.588	0.000	0.034	0.047
total_length_link_sea	-0.0026	0.011	-0.224	0.823	-0.025	0.020
n_sealinks	-0.2514	0.020	-12.509	0.000	-0.291	-0.212
n_links	0.4799	0.021	23.032	0.000	0.439	0.521
supplier_outdegree	-0.0759	0.005	-14.229	0.000	-0.086	-0.065
manufacturer_outdegree	-0.0345	0.004	-8.243	0.000	-0.043	-0.026
manufacturer_indegree	-0.0759	0.005	-14.229	0.000	-0.086	-0.065
export_port_outdegree	-0.0060	0.005	-1.139	0.255	-0.016	0.004
export_port_indegree	-0.0345	0.004	-8.243	0.000	-0.043	-0.026
transit_port_outdegree	0.0018	0.005	0.367	0.714	-0.008	0.011
transit_port_indegree	-0.0060	0.005	-1.139	0.255	-0.016	0.004
import_port_outdegree	-0.0729	0.002	-29.611	0.000	-0.078	-0.068
import_port_indegree	0.0018	0.005	0.367	0.714	-0.008	0.011
wholesaler_outdegree	-0.0651	0.004	-18.411	0.000	-0.072	-0.058
wholesaler_indegree	-0.0729	0.002	-29.611	0.000	-0.078	-0.068
retailer_indegree	-0.0651	0.004	-18.411	0.000	-0.072	-0.058
const	-4.996e-16	0.002	-2.21e-13	1.000	-0.004	0.004

Table E.9: Wholesales time statistics

Dep. Variable:	Wholesales_Time	**R-squared:**	0.517
Model:	OLS	**Adj. R-squared:**	0.516
Method:	Least Squares	**F-statistic:**	1886.
Date:	None	**Prob (F-statistic):**	0.00
Time:	12:07:08	**Log-Likelihood:**	-31661.
No. Observations:	30000	**AIC:**	6.336e+04
Df Residuals:	29982	**BIC:**	6.351e+04
Df Model:	17		
Covariance Type:	nonrobust		

Table E.10: Wholesales time coefficients

	coef	std err	t	P> |t|	[0.025	0.975]
n_suppliers	0.0439	0.009	4.857	0.000	0.026	0.062
n_manufacturers	-0.0620	0.012	-5.100	0.000	-0.086	-0.038
n_export_ports	0.0476	0.009	5.371	0.000	0.030	0.065
n_transit_ports	-0.0248	0.010	-2.374	0.018	-0.045	-0.004
n_import_ports	0.4264	0.008	52.128	0.000	0.410	0.442
n_wholesalers	-0.4973	0.008	-59.961	0.000	-0.514	-0.481
n_retailers	-0.0740	0.017	-4.485	0.000	-0.106	-0.042
total_length_link	-0.0076	0.006	-1.217	0.224	-0.020	0.005
total_length_link_sea	0.0954	0.020	4.675	0.000	0.055	0.135
n_sealinks	0.0369	0.036	1.031	0.303	-0.033	0.107
n_links	0.1833	0.037	4.943	0.000	0.111	0.256
supplier_outdegree	-0.0307	0.009	-3.229	0.001	-0.049	-0.012
manufacturer_outdegree	0.0150	0.007	2.010	0.044	0.000	0.030
manufacturer_indegree	-0.0307	0.009	-3.229	0.001	-0.049	-0.012
export_port_outdegree	-0.0308	0.009	-3.284	0.001	-0.049	-0.012
export_port_indegree	0.0150	0.007	2.010	0.044	0.000	0.030
transit_port_outdegree	0.0244	0.009	2.865	0.004	0.008	0.041
transit_port_indegree	-0.0308	0.009	-3.284	0.001	-0.049	-0.012
import_port_outdegree	0.1149	0.004	26.221	0.000	0.106	0.123
import_port_indegree	0.0244	0.009	2.865	0.004	0.008	0.041
wholesaler_outdegree	-0.0277	0.006	-4.399	0.000	-0.040	-0.015
wholesaler_indegree	0.1149	0.004	26.221	0.000	0.106	0.123
retailer_indegree	-0.0277	0.006	-4.399	0.000	-0.040	-0.015
const	5.152e-15	0.004	1.28e-12	1.000	-0.008	0.008

Table E.11: Demand side time statistics

Dep. Variable:	Demand_Side_Time	**R-squared:**	0.269
Model:	OLS	**Adj. R-squared:**	0.269
Method:	Least Squares	**F-statistic:**	649.3
Date:	None	**Prob (F-statistic):**	0.00
Time:	12:07:09	**Log-Likelihood:**	-37866.
No. Observations:	30000	**AIC:**	7.577e+04
Df Residuals:	29982	**BIC:**	7.592e+04
Df Model:	17		
Covariance Type:	nonrobust		

Table E.12: Demand side time coefficients

	coef	std err	t	P> |t|	[0.025	0.975]
n_suppliers	0.2930	0.011	26.343	0.000	0.271	0.315
n_manufacturers	-0.1096	0.015	-7.334	0.000	-0.139	-0.080
n_export_ports	-0.1579	0.011	-14.494	0.000	-0.179	-0.137
n_transit_ports	-0.0813	0.013	-6.330	0.000	-0.107	-0.056
n_import_ports	0.3043	0.010	30.248	0.000	0.285	0.324
n_wholesalers	-0.1708	0.010	-16.743	0.000	-0.191	-0.151
n_retailers	-0.1823	0.020	-8.984	0.000	-0.222	-0.143
total_length_link	0.3621	0.008	47.202	0.000	0.347	0.377
total_length_link_sea	0.4018	0.025	16.007	0.000	0.353	0.451
n_sealinks	0.0151	0.044	0.343	0.732	-0.071	0.101
n_links	0.1175	0.046	2.576	0.010	0.028	0.207
supplier_outdegree	-0.1424	0.012	-12.198	0.000	-0.165	-0.120
manufacturer_outdegree	-0.0211	0.009	-2.309	0.021	-0.039	-0.003
manufacturer_indegree	-0.1424	0.012	-12.198	0.000	-0.165	-0.120
export_port_outdegree	-0.0058	0.012	-0.504	0.614	-0.028	0.017
export_port_indegree	-0.0211	0.009	-2.309	0.021	-0.039	-0.003
transit_port_outdegree	-0.1573	0.010	-14.992	0.000	-0.178	-0.137
transit_port_indegree	-0.0058	0.012	-0.504	0.614	-0.028	0.017
import_port_outdegree	0.0193	0.005	3.583	0.000	0.009	0.030
import_port_indegree	-0.1573	0.010	-14.992	0.000	-0.178	-0.137
wholesaler_outdegree	-0.0208	0.008	-2.689	0.007	-0.036	-0.006
wholesaler_indegree	0.0193	0.005	3.583	0.000	0.009	0.030
retailer_indegree	-0.0208	0.008	-2.689	0.007	-0.036	-0.006
const	-1.624e-15	0.005	-3.29e-13	1.000	-0.010	0.010