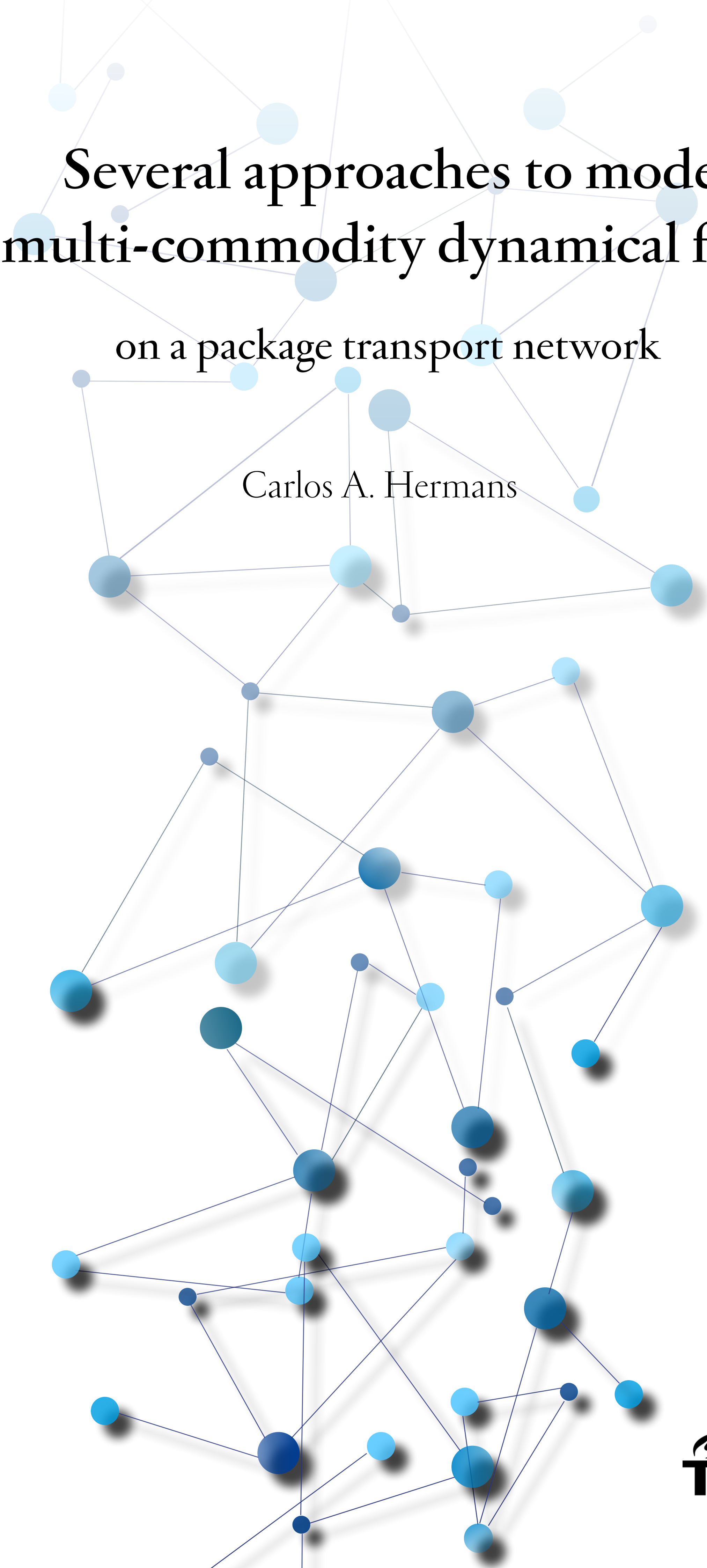
# Several approaches to model multi-commodity dynamical flow

## on a package transport network

Carlos A. Hermans



# Several approaches to model multi-commodity dynamical flow
## on package transport networks

by

## Carlos Hermans

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday July 12, 2019 at 14:00 PM.

|  |  |  |
|---|---|---|
| Student number: | 4384814 |  |
| Project duration: | December 1, 2018 – July 12, 2019 |  |
| Thesis committee: | Dr. J.L.A. Dubbeldam, | TU Delft, supervisor |
|  | Dr. ir. J. T. van Essen, | TU Delft |
|  | Dr. ir. H. M. Schuttelaars , | TU Delft |
|  | Dr. A. Stam, | Almende B.V., supervisor |

# Abstract

This thesis investigates several kinds of models of describing multi-commodity dynamical flow. Three models have been developed for a spatial logistic network, on which packages are delivered by vehicles and another model was developed to describe interactions between the customers, the stores and the deliverers, which does not explicitly take delivery time into account like before. The former three models consist of an agent-based model (ABM), and two different (differential) equation-based models (EBMs). One of the EBMs is built from the bottom up, and was created by averaging the randomness in the ABM. The other EBM was built top-down and involved optimizing a general model with respect to some parameters, such that its solution resembles the ABM solution as closely as possible.

The agent-based model is computationally intensive as it relies on a multitude of simulations to yield representative solutions. On the flipside, it is easy to use and naturally leads to variances in package amount and transportation time, which is not taken into account in the EBM, as it is by construction averaged out already. Contrary to the ABMs, though, it is hard to incorporate congestion into the EBM, but a steady state can be analytically described and it can be found numerically easily and quickly using Picard's fixed point method. An EBM with congestion is achievable, but integrating it numerically proves to be unstable, even for very small time steps. Moreover, the needed memory is in the order of 20 GB and the integration takes 27 hours to perform. Hence it can be concluded that the method, while demonstrating that creating a suitable EBM is possible, does not work well enough yet. The considered models go through a start-up process, after which a steady state is relatively quickly attained. It turns out that the EBMs do their job badly in this start-up phase, when applied on networks with very secluded nodes. The EBM with congestion displays this behaviour too, but provides accurate start-up processes for regular networks without secluded nodes, up to the steady state, after which it diverges. This is promising, as Picard's method gives an accurate steady state solution and the EBM gives an accurate start-up process. Together they thence cover the entire solution.

When omitting congestion, the results of both models match up to a discretization error, which can be remedied by refining the amount of time steps in both the ABM and EBM. This indicates that the models describe the same situation similarly and well. For larger networks, approximately beyond 40 nodes, the ABM simulations begin to require processing power in the order of days and time step refinement is not a feasible tool anymore, so the congestionless EBM becomes more desirable. Hence, the different models find their use in different cases and there is, except for exceedingly large networks, always an alternative to choose from. It is furthermore useful to know that both models give similar results in relatively simple situations, as the models might be adapted to work similarly in harder situations too.

Frustrated by the failure to produce a realistically functional EBM with congestion, the next attempt was to try and create the EBM with the optimization approach. This turned out to be a fruitful exercise, as the resulting EBM proved to be a reasonably good approximation. In the congestionless case, the start-up process in the ABM contained too many discontinuities in the derivative resulting in an abrupt change from start-up to steady state. This did not translate well to the EBM, but that is something that can possibly be improved upon by tinkering a bit at the chosen shapes of functions and their parameters. With congestion, however, the found EBM excels and almost completely overlaps with the ABM solution. This demonstrates that the chosen

approach has a lot of potential and could possibly be used to create a better understanding of EBMs that describe ABMs and how to make them.

In the second part of the thesis, the focus shifted to a generalized model (GM) approach of underlying interactions on multi-commodity dynamical flow networks; especially the dynamics describing order placement, scheduling and delivery. Such a network is cyclic and contains a feedback loop, where customers are less likely to order more products if they were recently delivered. This can range from someone not needing to purchase a car if they just bought a brand new one to one not needing to order dinner when they just did so. Such networks are in general described by four kinds of elasticities in this thesis, namely the elasticities to stock, inventory level, saturation, and co-production. These influence different parts in the network. For example, the negative elasticities to inventory level model that high inventory levels inhibit more inventory production, to prevent build-up, and low inventory levels allow for larger inventory production, to prevent drainage. The other elasticities fulfil similar roles.

It turns out that, when only looking at a single motive, with one customer, one store, and one deliverer, the network behaves predictably according to the explanations of the elasticities. In addition, it turns out that increasing the elasticity to saturation brings a lot of instability, while making it strongly negative also destabilizes the network. Increasing the elasticity to stock and decreasing the elasticity to inventory level both stabilize the network, and decreasing the elasticity to co-production destabilizes the network in general. This is intuitive, and has been verified using the generalized models.

Using a bifurcation analysis enables the creation of precise stability regions in simplified circumstances, by taking all elasticities of the same kind uniform, which still give a somewhat representative idea of the network. On the other hand, the networks have been investigated using the complementary statistical ensemble method. This yields similar results to the bifurcation analysis, but allows for all parameters to be varied and can hence lead to identification of the most sensitive and influential parts. These sensitivities and influences are quantified and displayed, which portrays the relative importance of parts in the network.

Our performed analyses yield physically interesting results. For instance, by starting with a single motive and building up to a larger network, it becomes clear that the overall shape of most bifurcation surfaces does not change, only their size; adding more interactions leads to more bifurcation surfaces. This suggests that behaviour for smaller networks is inherited by larger networks with the smaller network as a sub-motive. At the same time, an individual ensemble can sometimes yield significantly different results from the bifurcation analysis.

Another interesting result tells that blindly discriminating between customers in a network doubles the instability in a network with one store and two customers. Another discussed example implies that having one entity perform the same role in its multiple connected sub-networks destabilizes the network slightly. When complicating the networks more, it turns out that adding more customers and, thereby, introducing the elasticity to co-production, harshly destabilizes the network. This occurs even more so when stock is not assumed infinite. On the other hand, when considering two stores, this effect does not come up, as the co-production takes place on other parts in the network, which are not directly affected by stock, and hence the simultaneous introduction of both co-production and finite stock can be identified as strongly destabilizing.

Of course, these notions can be translated to other settings in other problems as well, and this is the beauty of the generalized models. Without knowing any interactions specifically, it is possible to make non-trivial useful statements about the stability of the network and how to improve it. This is then another complementary way to analyze networks, next to ABMs and EBMs, adopted in this case to capture the complex network dynamics in supply and transport systems.

# Contents

**Acknowledgements** 1

**Introduction** 3

**1 Theory** 5

1.1 Notation & Graphs . . . 5

1.2 Generalized models . . . 5

1.2.1 Bifurcation Analysis . . . 7

1.2.2 Statistical Ensemble . . . 9

**2 The Agent-based model** 11

2.1 Choices and simplifications . . . 11

2.2 Remarks . . . 13

**3 EBM description of the ABM** 15

3.1 First Approach . . . 15

3.2 Second Approach . . . 15

3.3 Without congestion . . . 16

3.4 With congestion . . . 16

**4 A simple example** 19

4.1 Without congestion . . . 19

4.2 With congestion . . . 20

**5 Larger networks** 25

5.1 The custom-made network . . . 26

5.2 The large map-like network . . . 29

5.3 Other networks . . . 31

**6 An optimization approach** 37

6.1 The optimization approach . . . 38

**7 The Generalized model approach** 43

7.1 One store and two customers . . . 43

7.1.1 Two Motives . . . 54

7.1.2 Equal $D$ nodes . . . 58

7.1.3 Extending the model; adding elasticity to co-production, $f_C$ . . . 63

7.1.4 Addition of a reservoir at $S$ . . . 66

7.2 Competition: Two stores and one customer . . . 69

7.2.1 Only one deliverer . . . 72

7.2.2 Addition of reservoirs at the $S$ nodes . . . . . . . . . . 75
7.3 Summary of the results for the GMs . . . . . . . . . . 77

**8 Conclusions & Future research** **79**
8.1 The spatial models; ABMs and EBMs . . . . . . . . . . 79
8.2 The generalized model . . . . . . . . . . 80

**A Results for the first EBM attempt** **83**

**B Derivation of the general solution to $N$-dimensional differential equations** **87**

**C Derivation of the matrix exponential derivative** **89**

**D Sensitivity and Influence** **91**

**References** **95**

# Acknowledgements

*Amicus fidelis protectio fortis; qui autem invenit illum, invenit thesaurum.*
*"A faithful friend is a sturty shelter; whoever finds one, finds a treasure."*

One obtains their Master's degree themselves, but that does not entail doing it alone. Without my friends and family, I wouldn't have been able to achieve what I have. Therefore, I would like to thank some people:

—Special thanks to Johan and Andries, my supervisors, who not only guided me through my master's thesis, but also advised me on a multitude of occasions this year.

—The entirety of Almende, as everyone who worked around me was such a positive influence and helped me enjoy the project even more.

—Huiswerk enzo, who made my university track and career that much more enjoyable. Thanks Pieter & Darcy, Rick, Massimo, Roel, Marco & Mike.

—My friends, who supported me on a multitude of occasions during the past 5 years. Thank you, Sybren, Ellis, Odo, Andy, Robbert, Ella, Victoria & Chanel, and many more.

—My loving family; my mother and father, without whom I would not have gone to university as I wouldn't be born anyway, and of course Alice, George and Bob.

Thank you for your never ending support. ♡

# Introduction

Multi-commodity dynamical flow networks span a lot of ground, especially nowadays with the abundance of network structures in everyday lives. In general, such networks can be employed in a variety of situations, with a variety of interactions on those networks. Multi-commodity flow phenomena are omnipresent in virtually everyone's everyday life, ranging from for instance package delivery services to internet traffic [7, 8] and signals or particles sent between cells in any living being. All these examples can, to some extent, be modelled like multi-commodity dynamical flow networks. In this thesis, two situations in particular will be under consideration. 1. Package flow on a logistic network, using agent-based and equation-based models, referred to as ABMs and EBMs respectively. 2. Package delivery with explicitly taking order placement and order processing into account using so-called generalized models. We next briefly discuss both models.

The first considered problem is package flow in a logistic network. Conventionally, there are two ways to model these phenomena. Agent-based models (ABMs), in which so-called agents act on predetermined decision rules, are being employed increasingly over the years, as computing power is more readily available. On the other hand, equation-based models (EBMs), which refers to the collective of transport differential equations, have been studied extensively as they yield a deeper understanding of underlying interactions in such a model. These dynamical flow problems have been treated before in for instance [9, 10, 11], however using different models. In this thesis, these two modelling approaches are investigated and compared specifically with a package delivery context in mind, and hence the equations are tailored to satisfy such a setting.

In the ABM, every package delivery truck is a so-called agent, which may drive around the network picking up and delivering packages, based on a certain set of decision rules. These decision rules basically constitute the management policy, or scheduling entity of the delivery company. The ABM is then solved using simulations, and the results are presented as averages over those simulations. Running sufficiently many simulations, often takes longer than numerically solving differential equations.
On the flipside, the EBM is an attempt at creating a solution that looks most like the ABM, without the need for simulations. This is tackled by averaging over the randomness in the ABM and deriving a differential equation based on the decision rules.
In these models, each node generally generates its own commodities, which are demanded by the other nodes. Thence these models constitute multi-commodity flow. For concepts such as package build-up and vehicle congestion, the specific commodities and vehicles do not matter. Therefore, most of the analysis does not explicitly distinguish between the different commodities, while they need to be present to satisfy heterogeneous demands. It is then of interest to investigate the accuracy of both models, their similarities and their runtimes to see how well both models do and whether they can be employed at all. Such comparisons have been done before [12, 13], a typical example of which is the SIR epidemic model treated in chapter 5 of [14]. However, it has mostly been done for simple ABMs, whereas treating more complex ABMs and comparing those to their EBM counterparts could possibly give some insight in whether this can readily be done for general ABMs.

After looking at the previous problem, the interest shifted to a different kind of modelling altogether; Generalized modelling (GM). This method of modelling is suitable for networks on which the shape of the interactions is unknown. This can be used to incorporate an analogon of the decision rules in the ABM. However,

in the ABMs considered in this thesis, these rules are generally not very complicated and depend linearly on the demand and the available supply.
Because of that, and inspired by [1, 2], the package delivery problem is expanded such that it includes the social interactions leading to order placement, the handling of these orders by a store entity, and moreover, customers and delivery companies are split into separate nodes this time around, which takes away from the abstraction in the ABM, but adds meaningful complexity to the GM. The nodes are thence no longer considered as actual physical locations, and this yields entirely different network dynamics. This constitutes the second considered problem. The extended problem can then be analyzed by bifurcation analysis and the statistical ensemble method. Gross et al. (2) discussed what happens for line and triadic motives. The network under consideration in this thesis gives rise to a cyclic triangle motive, which to my knowledge has not yet been considered in the literature. This makes the topic novel and opens up new kinds of interactions to be investigated.

The main use of GMs is however not to yield an explicit solution, but rather to investigate *stability properties of the system*, which is often of paramount importance in dynamical flow networks. As the precise shape of interactions in the system is unknown, this is usually done for systems with a sizeable set of parameters. Think, for instance, of Black Friday or Christmas sales and the strain these days put on stock and delivery in package flow networks. It is hard to predict how much strain will be exercised on both of these entities, but one can be sure that this strain is present and increased, when compared to the remainder of the year. GMs hence yield results that complement ABMs, and are a counterpart to EBMs, as the interactions are necessarily known to be able to solve ABMs and EBMs.

The main objective of this work is to compare these different models, namely agent-based, equation-based, and generalized models, for transport on networks.
The sections in this thesis are organized as follows: chapter 1 formally introduces the notations and less known methods, like the GM analysis, used in this thesis. Thereafter, chapters 2 to 5 closely follow the structure of my submitted paper [23], with more examples and discussion. Chapter 2 explains the ABM for the logistic package traffic network, along with a general algorithm and more specific choices and simplifications made in this specific situation. In chapter 3, the equations describing the EBM counterpart of the ABM are developed, with some earlier attempts presented in appendix A. Congestion is treated separately, as it introduces a plethora of problems that have to be dealt with and which increase the complexity a lot. Having established this basis, the applicability of these models is demonstrated on a line network in section 4, in which the solutions can still be determined by hand. Consecutively, these models are also applied to larger networks, which allow for a more meaningful assessment of their effectiveness. Chapter 6 concludes the EBM by attempting to reverse engineer the equations starting from the ABM solutions, using optimization packages to solve for the parameters in a chosen general model. Lastly, chapter 7 constitutes an extensive investigation of package delivery networks, by parametrizing the entire system. The equations are then analysed in two ways: with a bifurcation analysis to determine their stability regions and with a statistical ensemble method to determine the overall stability without simplifying the parameter sets. The results of these models are concluded in chapter 8.

# Chapter 1: Theory

Let us start off by establishing some basic notation and some central non-trivial concepts used in this thesis, the first of which being some necessary notation on graphs, which will mostly be used in chapter 3. The second section contains a little bit of background on generalized models, which will be used in chapter 7.

## 1.1 Notation & Graphs

In general, we will be looking at package transport on (logistic) networks, which are represented by directed graphs. In general, each node will produce its own commodity or commodities, which are demanded by other nodes. Let us denote $\mathscr{V}$ as the set of nodes, labelled from $0$ to $N-1$, where $N$ is the size of the network. Without loss of generality, these indices both denote the location of the respective node and the commodities originating from this node (although a node can produce multiple commodities, but that extension is straight-forward). Furthermore, $\mathscr{E}$ denotes the set of directed edges in the network graph, containing all $e = (i, j)$, with $i$ the outgoing node and $j$ the ingoing node. Let $\mathscr{N}_i$ be the set of neighbouring nodes of $i$: $\mathscr{N}_i = \{j : (i, j) \in \mathscr{E}\} \subset \mathscr{E}$, and $\overline{\mathscr{N}}_i = \mathscr{N}_i \cup \{i\}$.

$\mathscr{P}$ denotes the set of available paths in the graph, with $\mathscr{P}_{ij} \subset \mathscr{P}$ being the subset of available paths from node $i$ to node $j$, which is assumed to be non-empty, i.e. the graph is connected. Each such path is denoted by $p_{ijk}$, where $i, j \in \mathscr{V}$ and the subscript $k$ labels the chosen path between nodes $i$ and $j$. $p_{ijk}$ consists of an ordered set of edges. $\ell(i, j)$ denotes the weight of, or needed time to travel over, edge $(i, j)$. Moreover, $\ell(p_{ijk})$ denotes the length of path $p_{ijk}$; it is the sum of all the edges contained in $p_{ijk}$. Lastly, $x_{ij}$ is the amount of packages originating from node $j$, on node $i$, while $x_i$ is the amount of packages, summed over all commodities, i.e. $x_i = \sum_{j \in \mathscr{V}} x_{ij}$, and $F_{ij}$ is the outgoing flow from node $i$ to node $j$.

## 1.2 Generalized models

The discussion of generalized models (GMs) will be strongly based on a paper by Demirel et al. [1], and a paper by Gross and Feudel [1, 3], whence the inspiration came to incorporate GMs in this thesis. The explanation of GMs mostly follows the supplement of that paper, in order to illustrate how its analyses are performed.

Conventionally, dynamical flow systems, and, more precisely, supply systems, are modelled by agent-based models (ABMs), control theory and discrete-event simulation. Borne out of ecology, however, GMs have recently been adopted in supply networks to determine the stability of the network. The use of GMs brings the advantage of not needing a lot of information on the relations in the network, and how variables depend on each other. Hence, these models can be employed by only knowing the network structure itself. Additionally, these models are relatively efficient and light-weight, such that they can be feasibly run on larger networks.

A dynamical flow network can be modelled with $N$ differential equations, such that we write

$$\dot{x} = G(x) - L(x), \quad J_{ij}(x) = \frac{\partial}{\partial x_j}(G_i(x) - L_i(x)) \tag{1}$$

$x$ being the vector of variables. $G(x)$ is some gain function and $L(x)$ is some loss function, which corresponds with production/delivery and consumption/outflow of parts (partial products). A steady state of this system occurs at $G(x) = L(x)$. Conventionally, analysis around a steady state yields the stability of that steady state. If the leading (largest) eigenvalue of the Jacobian of the system is negative, when evaluated at the steady state, the system is stable. Otherwise, the system is unstable, with exception of 0 being the largest eigenvalue, which is a special case, and in general requires further analysis.

As we generally do not explicitly know $G(x)$ and $L(x)$, we cannot explicitly construct its Jacobian matrix, $J$. However, one might for instance know that higher supply leads to higher flow, and hence ranges for the non-zero matrix entries of $J$ can be inferred, by general knowledge of the system dynamics. This yields a qualitative description of the partial derivatives in equation 1. This can in general be done for all variables, $x_j$, and all gain and loss functions. One can take samples from these ranges and create a mesh, on each point of which one can test for stability. These can then be averaged out to find an average stabilility for different values of each variable. This allows one to infer stability properties, but also sensitivity and influence properties of nodes and variables in a network.

Concretely, in the context of supply networks, the variables in this thesis will be referred to as $P$, which denote the parts that traverse the network. Additionally, $G$ and $L$ denote the flow of parts, which constitutes the transformation into other parts. Let therefore, $P$, $G$ and $L$ be scaled as follows;

$$p = \frac{P}{P^*}, \quad g(p) = \frac{G(P)}{G^*}, \quad l(p) = \frac{L(P)}{L^*}. \tag{2}$$

with $P^*$, $G^*$, and $L^*$ the steady state values of $P$, $G$, and $L$ respectively. This gives rise to the entries of the Jacobian, around the stady state;

$$J_{ij} = \alpha(g_i^{pj} - l_i^{pj}); \quad g_i^{pj} = \frac{\partial g_i}{\partial p_j}, \quad l_i^{pj} = \frac{\partial l_i}{\partial p_j} \tag{3}$$

in which $\alpha = \frac{G^*}{P^*} = \frac{L^*}{P^*}$, which necessarily have to be equal in steady state.

In the remainder of this thesis, $G_i$ and $L_i$ will simply be denoted by $F_j$ for some indexing $j$. The stability of the system, around a steady state $p = 1$, can hence be completely described by the elasticity parameters $f_i^{pj}$. Moreover, using this framework, a concept of sensitivity and influence can be introduced. That is, if a perturbation of a certain variable $P_i$ results in a large disruption in the remainder of the network, this variable is said to be influent. Similarly, if a variable is susceptible to perturbations in other variables, this variable is said to be sensitive. Let $v$ be the set of right eigenvectors of $J$, and $w$ be the set of left eigenvectors of $J$. Then the influences and sensitivities are defined by

$$In_i = \ln\left(\sum_k \frac{|w_i^{(k)}|}{Re(\lambda_k)}\right), \quad Se_i = \ln\left(\sum_k \frac{|v_i^{(k)}|}{Re(\lambda_k)}\right), \tag{4}$$

as given in the supplement of Demirel et al. [1], which is a slight variation of the definition in Aufderheide et al [4]. $w^{(k)}$ and $v^{(k)}$ here denote the eigenvectors pertaining to eigenvalue $\lambda_k$. These definitions are only applicable just next to a bifurcation surface (i.e. with leading eigenvalue $0 < Re(\lambda) < 0.01$, for instance), to ensure that the ln won't have a negative argument. However, there could in rare cases be an additional negative eigenvalue such that $-0.01 < Re(\lambda) < 0$, which might still make the ln negative. These have to be discarded, as the ln yields complex values otherwise. Regardless, the resulting influence and sensitivity should still be representative.

In the paper by Aufderheide [4], it is argued that the right eigenvectors correspond to parts that react to a given perturbation, and how strongly they do, whereas left eigenvectors correspond to the kind of perturbation that can lead to a given response. The respective eigenvalues $\lambda_k$ then correspond with the strength of the reaction to the perturbations. Therefore, the impact of a perturbation is introduced by

$$I = J^{-1}K = \sum_k \frac{v^{(k)}w^{(k)} \cdot K}{\lambda_k}, \tag{5}$$

in which $K$ is some perturbation vector. This sum is basically an expansion of the Jacobian into its eigenvectors. Taking an average over possible perturbations $K$ yields equation 4, after taking the logarithm to keep numbers manageable.

Using this framework, there are two possible directions in which to perform an analysis; a bifurcation analysis, or an analysis of the system using a statistical ensemble, both of which are elaborated upon in the following sections.

### 1.2.1 Bifurcation Analysis

This route is concerned with exploring the bifurcation surfaces in the parameter space. That is, the parameter space is divided into regions where the behaviour of the investigated steady state is either stable or unstable. These regions are separated by bifurcation surfaces, on which the sign of an eigenvalue changes, which in turn may change the stability of the system. Assuming a steady state at $p = 1$, the Jacobian of the system is determined by the so-called elasticities, which are defined as the $g_i^{p_j}$ and $l_i^{p_j}$ from equation 3. However, as an $N$-dimensional system may contain $N^2$ elasticities in the worst case, such a system cannot meaningfully be represented in a two or three dimensional space, as the dimensions have to be reduced a lot.

These elasticities can generally however be grouped together and still give meaningful insight into the problem. For instance, in the example of a supply network, when the arrival of incoming parts increases, the production may increase. Conversely, when the demand of parts increases, the production is allowed to increase too, as the parts can be conveyed. If such interactions happen often in a network, it is plausible to assume that similar management policies, which are the physical equivalent of said elasticities, are employed at different entities in such a network.

Bifurcations arise in two different ways; either the leading eigenvalue becomes zero, or a pair of complex conjugate eigenvalues pass through the imaginary axis. In the former case, a saddle-node type bifurcation occurs (i.e. a saddle-node, transcritical or pitchfork bifurcation). In the latter case, a Hopf bifurcation occurs, which leads to local periodic solutions around the steady state.

Saddle-node type bifurcations can be found by solving $Det(J) = 0$, as the determinant of $J$ is only zero when an eigenvalue is zero. Hopf bifurcations can be found by solving $Res(J) = 0$, which denotes that the so-called resultant of a matrix needs to be zero. The resultant of the Jacobian matrix is created by assuming that there exists an eigenvalue pair $\lambda_- = -\lambda_+$. In the case of a Hopf bifurcation, these $\lambda$ are purely imaginary. Consequently, the characteristic polynomial of $J$ should be determined, and in general it equals

$$\mathscr{P}(\lambda) = \sum_{n=0}^{N} c_n \lambda^n, \tag{6}$$

in which $N$ is the size of the Jacobian matrix, or the system.
Evaluating $\mathscr{P}$ at both $\lambda_+$ and $\lambda_-$ gives two equations, in which the even powered terms are the same, but the odd powered terms differ by a $-$-sign:

$$\sum_{n=0}^{N} c_n \lambda_+^n = 0, \quad \sum_{n=0}^{N} c_n (-1)^n \lambda_+^n = 0, \tag{7}$$

adding and subtracting both of which yields

$$\sum_{0\leq 2m\leq N} c_{2m}\lambda_+^{2m} = 0, \qquad \sum_{0\leq 2m+1\leq N} c_{2m+1}\lambda_+^{2m+1} = 0. \tag{8}$$

Assuming $\lambda_+ \neq 0$, the latter can be divided by $\lambda_+$ and $\lambda_+^2$ can be set to $\chi$, which is the definition of the Hopf number. Two symmetrical eigenvalues hence exist if

$$\sum_{0\leq 2m\leq N} c_{2m}\chi^{m} = 0, \qquad \sum_{0\leq 2m+1\leq N} c_{2m+1}\chi^{m} = 0, \tag{9}$$

where $\chi < 0$ if both eigenvalues are imaginary and, by extension, a Hopf bifurcation occurs. If $\chi > 0$, a real Hopf situation occurs, in which a homoclinic orbit is created, as there is a negative eigenvalue that induces an attraction that is balanced by the repulsion from its symmetric positive eigenvalue. Lastly, when $\chi = 0$, a Bogdanov-Takens bifurcation occurs, and when $\chi$ is undefined, the situation should be treated with more care, as a more exotic bifurcation takes place.

Hence, an order $N$ polynomial equation has been reduced to two order $\frac{N}{2}$ polynomial equations. Equations (9) can further be reformulated to finally create the resultant, in the following way; Let us denote the odd equation by $\mathscr{Q}(\chi)$ and the even equation by $\mathscr{S}(\chi)$. For some $\chi_0$ satisfying $\mathscr{Q}(\chi_0) = \mathscr{S}(\chi_0) = 0$, it holds that

$$\mathscr{Q}(\chi) = (\chi - \chi_0)\mathscr{q}(\chi), \quad \mathscr{S}(\chi) = (\chi - \chi_0)\mathscr{s}(\chi), \tag{10}$$

with the degree of $\mathscr{q}$ being less than the degree $\mathscr{Q}$ and the degree of $\mathscr{s}$ being less than the degree of $\mathscr{S}$. This can be written as

$$\mathscr{Q}(\chi)\mathscr{s}(\chi) - \mathscr{S}(\chi)\mathscr{q}(\chi) = 0. \tag{11}$$

This equation only has non-trivial solutions for specific $\mathscr{q}$ and $\mathscr{s}$. Rewriting $\mathscr{q}$ and $\mathscr{s}$ into the polynomial basis changes 11 to a matrix equation, which can be formulated as follows:

$$\mathscr{R}z = \begin{pmatrix} c_1 & c_0 & 0 & \dots & 0 \\ c_3 & c_2 & c_1 & \dots 0 & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_N & c_{N-1} & c_{N-2} & \dots & c_0 \\ 0 & 0 & c_N & \dots & c_2 \\ 0 & 0 & 0 & \dots & c_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & c_{N-1} \end{pmatrix} z = 0, \tag{12}$$

which only has non-trivial solutions if $Res(J) \equiv Det(\mathscr{R}) = 0$. This resultant is zero, when both polynomials have an equal root. $z$ is then a representation of $\mathscr{q}$ and $\mathscr{s}$ in the polynomial basis.

This matrix $\mathscr{R}$ is constructed by setting the $\mathscr{R}_{11} = c_1$. Let in general $\mathscr{R}_{i,j} = c_k$, then $\mathscr{R}_{i+1,j} = c_{k+2}$ and $\mathscr{R}_{i,j+1} = c_{k-1}$. The elements with an invalid index of $c$ (i.e. larger than $N$ or smaller than $0$), are simply set to zero.

There is a simple algorithm for generating $\chi$. Let $A$ be the matrix created when removing the last two columns and the last row from $\mathscr{R}$. Let $B$ be the matrix created by removing the first row from $A$ and $C$ be be the matrix created by removing the second row from $A$. Then

$$\chi = -\frac{|B|}{|C|}. \tag{13}$$

In fact, $A$ may created by removing any row $j$ and $C$ by removing any row $j + 1$. In practice, though, it is often easier to just determine the nature of the bifurcation surfaces by testing the eigenvalues there, or by using numerical continuation software like Matcont [18] or AUTO [19].

To recap, saddle-node type bifurcation surfaces can be found by solving $Det(J) = 0$ and Hopf-type bifurcations can be found by solving $Res(J) = 0$. However, it is often not necessary to find all bifurcation surfaces, as different bifurcation surfaces generally pertain to different eigenvalues traversing zero. As a result, a lot of network information is contained in the stability regions alone. These can be simultaneously determined with symbolic computation software, such as Maple [20], by solving the following equation

$$\max_{k} Re(\lambda_k) = 0. \tag{14}$$

### 1.2.2 Statistical Ensemble

Sometimes, a bifurcation analysis is not necessary, too hard or one just wants a more concrete measure for stability in the network. In such a scenario, the statistical ensemble (SE) method is probably the right way to go. Following this method, the parameter space is sampled in some way. The network stability can be determined for each such sample, by determining the eigenvalues of the Jacobian.
Subsequently, when varying over the values for a single parameter, their results can be averaged over the remainder of the parameter space to yield an average stability as a function of that single parameter. Accordingly, when only considering the samples with a small positive leading eigenvalue, an average influence and sensitivity can also be calculated.

The boon of this method is that it is relatively lightweight and easy to use, compared to generating 3D bifurcation surfaces. It is also often more readily intelligible, when the bifurcation surfaces can be quite complicated, making interpretation not always straightforward. However, for a large array of parameters, the SE method might prove to lack a proper spread of samples and is hence not very accurate. The accuracy gain due to a larger sample size is directly limited by computing power, and might hence not be feasible for a network with many parameters, such as the elasticities mentioned above.

# Chapter 2: The Agent-based model

The contents of this chapter present the agent-based model (ABM) used to describe package flow on a logistic network. Opposed to equation-based models (EBMs), ABMs do not explicitly describe a system with equations. Rather the model manifests itself in so-called agents, which make decisions and perform actions based on certain (decision) rules. The only way to realize these models is by simulation, as decisions generally do not translate well to equations.

In the researched context of package transport in logistic networks, both customers (which coincide with nodes in this case, and should hence rather be considered as a collective of customers) and trucks act as agents, although the latter is a more obvious choice and more explicitly follows decision rules. The rules governing the vehicles determine how the vehicle drives through the network and how the path is chosen. The rules governing the customers determine the generation of demand to be satisfied by delivered packages. The latter of these is considered to be simple in this thesis, and hence does not express decision-like properties, but the model presented in this section could handily be adapted to include more sophisticated customer behaviour. In the general case where each node produces some commodity, and supplies that commodity according to demand of that commodity on other nodes, the ABM is formulated as follows:

1. At every time step, some random amount of demand for each commodity is generated on each node. 2. Each node has its own (possibly infinite) amount of trucks. A truck with a package on its home node will depart to its destination using some path through the graph, to satisfy the positive demand. [1] 3. Once a truck arrives at its destination, it can again eventually detruck, after which it is sent back to its origin node. 4. When the truck is back at its origin, it is free to accept subsequent cargo again. 5. All trucks that have not arrived at their origin or destination, will move one unit forward over their current edge, or onto the next edge, in case they arrived on an intermediate node.
6. At the end of every time step (after all truck movements have been performed) all probabilities of choosing paths need to be updated according to the change in the amount of trucks on edges. 7. The resulting effective edge length will be updated as well. This allows for the modelling of congestion.
Taking $T$ to be the amount of time steps over which the simulations take place, and $N$ to be the size of the network, algorithm 1 gives a general pseudo-code of an ABM simulation. Of course, somewhere along the way, one has to keep track of the amount of available trucks and packages pertaining to nodes.

## 2.1 Choices and simplifications

Many choices are necessary in order to fully establish this model. As mentioned before, each node generally produces its own commodity, which is generally desired by all the other nodes, and hence each node has its own supply and demand. For simplicity, supply is assumed such that the demand is exactly balanced at all times. This can practically be understood as if items are in stock. The demand is assumed to be Poissonian distributed, with a mean that is constant in time. This corresponds with the simple customer decision rule.

[1] Before such a truck actually leaves its home node, it possibly still needs to load the cargo for some time steps, such that it not yet driving, but is occupied.

**Algorithm 1:** The algorithm for a general multi-commodity package delivery ABM simulation.

```
for $t \leq T$ do
    for all nodes do
        Generate supply & demand
    for all nodes do
        while trucks and supply are available do
            Choose destination for truck
            Start loading the truck
        for all non-idle trucks do
            if truck is done loading then
                Send the truck on the path to its destination
            else if truck is back at its origin then
                Retire the truck
        else if truck is at its destination then
            De-truck
        else if truck is done detrucking then
            Send the truck back to its origin
        else if truck is at its next node then
            Send the truck on the way to its subsequent node
        Count the package and truck transport
    for all origin-destination pairs do
        Update choosing probabilities and path lengths, based on amount of trucks on the edges
```

To ensure that the simulations do not last unnecessarily long, the available paths for every pair of nodes $i$ and $j$ have been determined using Yen's k shortest path algorithm [6]. Here the amount of considered paths, $k$, has been chosen relatively small to ensure that unnecessary detours will not be considered. However, it should still be chosen sufficiently large to make sure that relevant paths are not omitted. The probability of a path being chosen is assumed to read

$$P_{ijk} = P(p = p_{ijk}) = \frac{e^{-\ell(p_{ijk})/\omega_{ij}}}{\sum_k e^{-\ell(p_{ijk})/\omega_{ij}}}, \tag{15}$$

in which $\omega_{ij}$ is a weight which determines how likely longer paths are to be chosen. In particular, as $\omega_{ij} \to 0$, only the shortest path will be considered. Similarly, as $\omega_{ij} \to \infty$, all paths will be chosen with equal probability. In this thesis, $\omega_{ij}$ is constant over all pairs $i, j$.

Nodes having more unanswered demand will have a higher priority of supply being sent to them, when choosing the destination for a truck. However, if available trucks are abundant, this should not be relevant. The capacity of a truck has been arbitrarily set to 1 for computational convenience. Consequently, supply is only generated in integers and hence trucks are always either full or empty.

All trucks have identical properties: they have the same maximum velocity, capacity, load times and detruck times. When trucks are deployed, they stay some time at the origin to load the packages and when they arrive at their destination, they remain there to detruck the packages; for instance, the packages might have to be delivered at the door, which takes some time. A package is assumed to be delivered at the beginning of the detrucking process, and that is the moment of consumption. Trucks always have to move back to their origin and finish their shift there. When returning, the truck chooses a new return path, which may differ from the initial path, as traffic flow may have changed at a later time, or the driver simply feels like taking another road.

If a package has to be delivered at the same node as its origin, the delivery time has been chosen to equal

loading time.
Every truck on a road adds an increment to the roads respective congestion, which is linearly dependent on the amount of trucks on that road. In general, these could be chosen individually over all edges, making the amount of congestion parameters 26. This could represent different amounts of lanes on roads. In this thesis, they are chosen constant over all edges.
As packages move between nodes, they are assigned to both nodes, proportional to how close they are to those nodes, i.e. when a package is midway between two nodes, it belongs half to both. This is updated each time step for every truck.

## 2.2 Remarks

The chosen congestion model is admittedly not realistic in general. This choice has purely been made for computational convenience and verifiability. It does this job well, as it illustrates the influence of congestion on flow in general, while it is easy to implement and verify. A more realistic approach might be $v \propto \frac{1}{\rho}$, but makes the system a bit harder to analyze.

Setting the capacity of a truck to 1, gives freedom for choosing its unit. One can then artificially tune the flow based on how large such a unit is supposed to be. By the same token, load and detruck times are arbitrary concepts. When thinking of these, it is important to keep the above mentioned definition of delivery, which has been chosen to take place at the beginning of the detrucking process, in mind. Likewise, the way to define flow is not straight forward. One could argue that packages are only transferred, when they arrive at their destination, as opposed to considering a constant continuous flow, which is weighed by the effective edge length. In any way, the flow should average to the same value. Worst case, the amount of packages on a node would be shifted slightly in time, as arrival times would not coincide with the definitions here any more, but otherwise this choice is arbitrary and does not impact the solution of the ABM substantially.

It is useful to notice how the definition of package unit and truck amount are linked. Effectively, $n$ trucks, containing 1 unit each, bear the same physical meaning as one truck containing $n$ units (apart from the truck amount, of course), with a congestion weight $n$ times as high. A poor man's way to scale up simulations, can then be achieved by letting trucks artificially contain more cargo this way. However, of course it will take more simulations to average this out well, so in the end this does not make much of a change.
Necessarily, the required amount of available trucks, which is a valid variable in real life logistic networks, is bounded by the outgoing path lengths for a node and is hence in general dependent on the node and the supply rate. A necessary amount of trucks hence becomes a node-dependent network property.

This concludes the exposition of the considered ABM. In the following chapter, EBMs are developed to model the same situation as the ABMs from this chapter.

# Chapter 3: EBM description of the ABM

At the start, the primary focus of this thesis was to investigate whether this logistic flow problem can be modelled in more effective ways, whether similar results can be achieved. Later on though, this research question evolved into investigating which kind of model would yield which kind of results; which model could capture which kind of behaviour and which model would be useful to employ in which kind of situation. The main goal of this chapter is to state a basis for the comparison between agent-based and equation-based model solutions, and hence to explain how to develop EBMs for this cause.

## 3.1 First Approach

As a first attempt, the assumed differential equation model was chosen as

$$\dot{x}_i = u_i + \sum_j (F_{ji} - F_{ij}) - w_i, \tag{16}$$

with the assumption $F_{ij} = \alpha_{ij} x_i$, which is at first not a obvious choice, as there is no reason to believe flows to be linearly dependent on package amounts. However, this system is linear and by extension easy to analyze. As a result, succesfully being able to cast the model in this form would be favourable.

However, there is no natural way to cleverly determine these $\alpha_{ij}$ beforehand. As a result, they have to be determined by comparing to ABM simulations. Effectively, the task is then to find a matrix $A$, containing all $\alpha_{ij}$ as elements, such that

$$\dot{x} = (A^T - diag(A \cdot 1))x + u - w, \tag{17}$$

where 1 here denotes the all one vector and $diag(z)$ is the matrix with the vector elements of $z$ on its diagonal.

This set of equations led to the analysis presented in appendix A, and it did not turn out to work as desired. It was hence necessary to tackle the problem in another way.

## 3.2 Second Approach

The equations are henceforth constructed by averaging over the randomness in the ABM, and are hence expected to yield the same results. However, by nature, differential equations are continuous in time, contrary to the ABM, which clearly requires time discretization to trigger events. A numerical approximation of the differential equation, with appropriate time steps, could be considered as an analogon to the average realization of the ABM, and the differential equation should hence be thought of as the continuum limit of said ABM.

After averaging over the random demand in the ABM, the EBM simply reads:

$$\dot{x}_i = s_i(t) - c_i(t) + \sum_{j \in \overline{\mathcal{N}_i}} (F_{ji}(t) - F_{ij}(t)). \tag{18}$$

Here, the dot denotes the time derivative as usual, $x_i$ denotes the (summed amount of) packages on node $i$ and $s_i$, $c_i$ respectively denote the generated supply and consumption rates.
One should read $F_{ij}(t)$ as "the flow, going from node $i$ to node $j$, that (temporarily) arrives in node $j$ at time $t$". Keep in mind that $j$ can equal $i$, although that is strictly speaking not a flow. These flows $F_{ij}$ are in turn determined by iterating over all possible origin-destination pairs and taking the relevant paths (with probabilities) that pass through $(i, j)$.

Two cases are distinguished in this thesis. The first of which does not take congestion into account; the phenomenon by which cars slow down on a road due to a higher density of cars, eventually resulting in traffic jams. The second case does include congestion, which makes the situation quite a bit more complicated, as now path lengths are not constant. Let us first consider a congestionless model, as some notions carry over to the model with congestion.

## 3.3 Without congestion

In the absence of congestion, hence for fixed distances between the nodes, explicit expressions for $c_i$ and $F_{ij}$ can easily be found.

$$c_i = \sum_{j=0}^{N} s_{ji} \sum_{k} P_{jik} u_{(\ell(p_{jik})+t_l)}(t), \tag{19}$$

where $u_\tau(t)$ denotes the Heaviside step-function with shift $\tau$ and $t_l$ denotes the loading time of the truck. $s_{ij}$ is the supply balanced by the demand on node $j$ of commodity $i$. $s_{ij}$ is then the average generated supply, and hence the parameter used in the chosen Poisson distribution when performing the ABM simulations, for each origin-destination pair. For convenience, $s_{ij}$ have been assumed to be equal over all $i$ and $j$. Equation 18 is then derived by using that the average of the package generation is constant and that packages have a certain constant arrival time. In general, each of the different commodities can arrive at $i$, using different paths with probability $P_{jik}$. In each of these paths a package sent at $t = 0$ fully belongs to its destination node at $t = \ell(p_{jik}) + t_l$ and is consumed at that time. Hence, this is the time shift in the Heaviside step-function. The resulting $c_i$ is thence a summation over all possible origins and the possible paths taken from those origins to node $i$, of the individual consumptions of the packages that took these paths.
By considering the arrivals at previous and subsequent nodes, $F_{ij}$ can be written as

$$F_{ij}(t) = \sum_{l,m\in\mathscr{V}^2} s_{lm} \sum_{(i,j)\in p_{lmk}} P_{ijk} u_{(\tau_{lmk,i}+t_l)}(t), \tag{20}$$

where $\tau_{lmk,i}$ denotes the amount of time needed to arrive at node $i$, when taking path $k$ between nodes $l$ and $m$. A concrete example will be given in the next chapter. Knowing the $c_i$ and $F_{ij}$ completely determines $x$, as the equation can now be simply integrated; note that $\dot{x}$ does not depend on $x$.

As all path properties remain constant in this situation, the equation is quickly made. However, this proves to be more challenging once congestion is introduced.

## 3.4 With congestion

When adding congestion, the equations become much harder to solve, as the travel time becomes dependent on the amount of trucks that linger on the roads. One has to bear in mind that, without congestion, the steady state amount of trucks on an edge is (on average) twice the amount of packages, as trucks have to travel back to their origin in the used model. A steady state can then be established with Picard's fixed point method [22], by

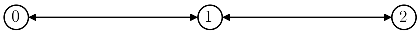


**Figure 3.1:** The extended line network used as an illustration for the definition of $\tau$, and $\theta$.

iterating as follows: starting from a congestionless steady state, travel times and probabilities $P_{ijk}$ are updated by recursively solving the congestionless equation, until convergence to the congested steady state is reached. This approach is simple, yet relatively efficient. Its implementation will be demonstrated in the next two chapters.

It is also possible to develop equation 18, when congestion is included. It is however a tougher problem to solve, as path lengths are no longer constant. As a result, $c_i$ and $F_{ij}$ are more complicated. $s_i$ is still assumed constant, though. Let us first and foremost establish some notation, to facilitate the explanation of this equation; in a similar fashion to before, $\tau_{ji}(t)$ denotes the time at which a package would arrive at node $i$, coming from node $j$, i.e.:

$$t_0 + \ell((i,j), t_0) = t_1 \implies \tau_{ij}(t_1) = t_0, \tag{21}$$

where $\ell(\cdot, t)$ is now a function of time as well. However, the time-dependence is implicit, and $\ell((i,j))$ really is a function of the amount of trucks on edge $(i,j)$. For instance, in the assumed simple case of linear congestion, $\ell(\cdot)$ would read

$$\ell((i,j), t) = \ell((i,j), 0) + w_c T_{ij}(t), \tag{22}$$

in which $T_{ij}(t)$ denotes the amount of trucks on edge $(i,j)$ at time $t$ and $w_c$ denotes the congestion weight.

Consequently, we necessarily require two more definitions, which deal with whether a package has been able to reach a certain node in its path. The first of the two is $\theta_{ijk,l}(t)$, which denotes the time required to reach node $l$, somewhere in path $p_{ijk}$. This $\theta_{ijk,l}(t)$ is built from all respective $\tau_{mn}$, where edges $(m,n)$ lead up to node $l$, with accordingly shifted times. As a simple illustrative example, let us consider $i = 0, j = 2$ and $l = 2$, with $p_{ijk} = \{(0,1),(1,2)\}$; a path between nodes 0 and 2, with one intermediate node 1, as shown in figure 3.1. To compute $\theta_{02k,2}(t)$ in this case, a delivery truck necessarily has to pass $(1,2)$ before arriving at node 2, such that the truck passed node 1 at $\tau_{12}(t)$. Subsequently, that delivery truck necessarily has passed $(0,1)$ just before, such that the truck left node 0 at $t = \tau_{01}(\tau_{12}(t))$. This demonstrates that this process is recursive. In total, we find here that $\theta_{02k,2}(t) = t - \tau_{01}(\tau_{12}(t))$, which is the total length necessary to traverse the path. When considering larger networks in general, such sums become increasingly difficult to compute and explicitly obtain.

The second definition is merely an extension of the previous one. By construction of the model, trucks have to return to their origin, and the EBM somehow has to keep track of that too. Let therefore $\vartheta_{ijk,l,m}(t)$ be the necessary time to reach node $l$ in path $p_{ijk}$ on the way back. There are a finite amount of initial paths $p_{jim}$ and in general all have different lengths. This $\vartheta_{ijk,l,m}$ is constructed very much the same as $\theta_{ijk,l}$, with the addition of also containing the length of the initial path $p_{jim}$, constructed in the same way, with the $\tau$s, and also containing the detruck time, $t_d$.

One has to keep in mind, though, that as one introduces discontinuities when numerically integrating this, $\tau_{ij}(t)$ is not guaranteed to exist any more, because $\ell((i,j),t)$ makes jumps at every time step. An edge length can increase just enough to create gaps in the continuity of truck flow. This effect is relatively small. Hence a valid option is to omit this effect, and assume no gaps. A more thorough approach involves keeping track of the gaps and not letting trucks arrive at nodes when $\tau_{ij}(t)$ does not exist, which has been opted for here. This makes the functions less pretty to integrate, as they contain more step functions, but in the continuum limit they should yield the same result.

Taking $s_i$ constant and hence known, these $\theta$ and $\vartheta$ cover how to find $c_i(t)$ and $F_{ij}(t)$ in the following way:

$$c_i(t) = \sum_{j\in\mathscr{V}} s_{ji} \sum_{k} P_{jik}(t - \theta_{jik,i}(t))u_{(\theta_{jik,i}(t)+t_l)}(t), \tag{23}$$

where $P_{ijk}$ has explicitly been made time dependent, and $u_x(t)$ is the Heaviside step-function with shift $x$.

This equation is formed by summing over all paths that reach node $i$, weighed with $P_{jik}$, evaluated on $\theta_{jik,i}(t)$, as as that is the time at which the paths were chosen, with the probability distribution at that time. The Heaviside step-function ensures that these paths are counted when they actually have had time to reach node $i$.

$$\dot{F}_{ij}(t) = \frac{\sum_{l,m\in\mathscr{V}^2} s_{lm} \sum_{p_{lmk}\ni(i,j)} P_{lmk}(t-\theta_{lmk,i}(t))u_{(\theta_{lmk,i}(t)+t_l)}(t)}{\ell((i,j),t)} - \dot{F}_{ij}(\tau_{ij}(t)), \tag{24}$$

Equation 24 consists of two terms. Similar to equation 23, the numerator of the first term is simply a sum over all paths that pass through $(i,j)$. The denominator, $\ell((i,j),t)$ exists, as the flow continuously adds to the next node, over the course of the entire edge. The second term, $\dot{F}_{ij}(\tau_{ij}(t))$ takes the flow into account that has already reached node $j$ by now, and hence does not contribute to the transport on edge $(i,j)$ anymore.

As functions for $c_i(t)$ and $F_{ij}(t)$ have been created, equation 18 is integrable. However, one should bear in mind how neither $c_i(t)$, nor $F_{ij}(t)$ are easily determined beforehand, and require knowledge of the evolutions of the $P_{ijk}(t)$, $\ell((i,j),t)$ and $T_{ij}(t)$. It is however known that $P_{ijk}(t)$ is a predetermined function of $\ell((i,j),t)$, which in turn is a predetermined function of $T_{ij}(t)$. This $T_{ij}(t)$ is thence the last variable to require an equation, in order for the problem to be solvable. It should be stressed that the only difference with the congestionless equation is that $\ell((i,j),t)$ now depends on $T_{ij}(t)$, which makes the equation increasingly complex, as a lot more time steps have to be taken into consideration.

Using the definitions of $\theta_{ijk,l}(t)$ and $\vartheta_{ijk,l,m}$, $T_{ij}(t)$ can be described as well:

$$\begin{aligned}\dot{T}_{ij}(t) = &\sum_{l,m\in\mathscr{V}^2} s_{lm} \sum_{p_{lmk}\ni(i,j)} P_{lmk}(t-\theta_{lmk,i}(t))u_{(\theta_{lmk,i}(t)+t_l)}(t)\\ + &\sum_{l,m\in\mathscr{V}^2} s_{lm} \sum_{p_{lmk}\ni(i,j)} P_{lmk}(t-\theta_{lmk,i}(t)) \sum_{n} P_{mln}(t-\vartheta_{mln,i,n}(t))u_{(\vartheta_{lmk,i,n}(t)+t_l)}(t) - \dot{T}_{ij}(\tau_{ij}(t))\end{aligned} \tag{25}$$

The first term here is simply a sum over all initial paths going through $(i,j)$. The second term is a sum over all initial paths going to node $l$ and returning from node $l$ back to node $m$, while passing through edge $(i,j)$. This is where $\vartheta$ comes in, which keeps track of the return paths. Lastly, the final term removes the trucks that have arrived at the next node and hence no longer contribute to the amount of trucks on the edge.

This chapter has extensively covered how to build the EBM counterpart of the originally considered ABM, along with a brief mention of a first failed attempt to do so. The next step is to investigate how this solution holds up in both accuracy and efficiency to the ABM, which will be done in the following chapters.

# Chapter 4: A simple example

Let us first discuss a simple example to illustrate the different concepts and demonstrate the discussed models. This serves as a way to get acquainted with the models and to check whether they work well, as this example is sufficiently simple to be computed manually.

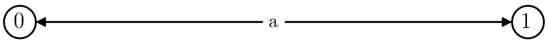


**Figure 4.1:** A simple line network, with edge length $a$.

The simple network consists of two nodes labelled 0 and 1, connected by a bidirectional edge, with length $a$, see figure 4.1. Let us furthermore also assume that both nodes possess sufficiently many trucks and that they generate demand for both commodities at an average rate of $\frac{1}{2}$ per time step, such that the total demand on each node is 1 on each time step.

## 4.1 Without congestion

In the situation without congestion,

$$s_i = 1, \quad c_i = \frac{1}{2}\left(u_{(a+t_l)}(t) + u_{t_l}(t)\right). \tag{26}$$

Both $s_i$ equal 1, as each node supplies enough to satisfy demand on both nodes, which equal $\frac{1}{2}$. As supply is generated, trucks will begin to fill and leave their origin node. After the loading time, they deliver to their origin node, whence the $u_{t_l}(t)$ comes from, and hence those packages can be consumed. The packages that are meant for the other node, first also need to traverse the edge. They are then ready to be consumed after $t_l + a$ time steps, whence the other Heaviside step-function originates. Furthermore, for completeness:

$$\sum_{j\in\overline{\mathcal{N}_i}} F_{ij} = \sum_{j\in\overline{\mathcal{N}_i}} F_{ji} = \begin{cases} 0, & \text{for } t < t_l \\ \frac{t-t_l}{2a} + \frac{1}{2}, & \text{for } t_l \leq t \leq a + t_l \\ 1, & \text{for } t > a + t_l \end{cases}. \tag{27}$$

By symmetry of the network, it is evident that $\sum F_{ij}$ and $\sum F_{ji}$ cancel at any time here. They also do not depend on $i$, as the network is symmetrical and very simple. In general, neither of these statements hold, though, until a steady state is reached, where the flows have to cancel of course. The differential equation in this case then simply reads

$$\dot{x}_i = 1 - \frac{1}{2}\left(u_{(a+t_l)}(t) + u_{t_l}(t)\right) \tag{28}$$

Hence, a steady state should be expected for $t > t_l + a$, after which $\dot{x}_i = 0$. Both $x_i$ then equal $\frac{a}{2} + t_l$. Note that these derivations only work when the amount of available trucks is abundant. Otherwise, surely the solution will diverge, but this behaviour is harder to accomplish using an EBM. Integrating equation 28 yields:

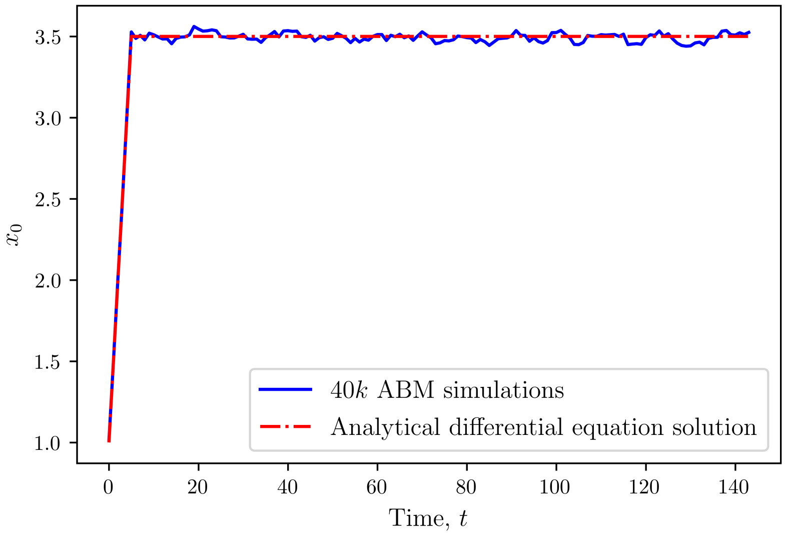


**Figure 4.2:** Time evolution of $x_0$, without congestion, for the line network. There is an abundance of (10000 available) trucks per node. The edge length $a$ is 5 units and the amount of time steps is 144. The load and detruck times are 1 timestep. The demand generation for both commodities on both nodes is $\frac{1}{2}$ per timestep. The first time step lies at $t = 1$, and hence $x$ does not start at 0.

$$x_i = \begin{cases} t, & \text{for } t \leq t_l \\ t_l + \frac{1}{2}(t - t_l), & \text{for } t_l \leq t \leq t_l + a \\ t_l + \frac{a}{2}, & \text{for } t \geq t_l + a \end{cases} \tag{29}$$

The EBM for this line network without congestion is not very complicated, and the solution is readily available. However, to demonstrate how easily this equation integrates numerically, it has been done so using Euler forward.

Figure 4.2 shows 40000 ABM simulations in blue and the numerically integrated EBM solution in dashed red, which leaves the impression that the ABM and EBM give similar results. Whether this generally holds for more complex networks is hard to accurately predict. However, this is at worst an indication that ABMs and EBMs can sometimes yield similar and complementary solutions. Choosing $t_l$ and $a$ integer, allows the ABM to be implemented more easily. Coincidentally, if the time steps for numerical integration divide these ABM time steps, the shift in the Heaviside functions are also divided perfectly, which results in an equal numerical and analytical solution reading

$$x_i = \begin{cases} t, & \text{for } t \leq 1 \\ 1 + \frac{1}{2}(t - 1), & \text{for } 1 \leq t \leq 6 \\ \frac{7}{2}, & \text{for } t \geq 6 \end{cases} \tag{30}$$

This simple equation for $x_i$ constitutes the solution to the congestionless equation.

## 4.2 With congestion

Next, the equations from section 3.4 are used to elicit the solution to the differential equation with added linear congestion. Again, $s_i = 1$, but $c_i$ and the $F_{ij}$ differ from equations 26 and 27. By symmetry, the flows still cancel though.

Following equation 23, we have:

$$c_i(t) = \sum_{j \in \mathscr{V}} \frac{1}{2} \cdot u_{\theta_{ji1,i}(t)+t_l}(t), \tag{31}$$

where the summation over the paths has been omitted, as there only exists one path (labeled 1), which is chosen with probability 1. For this network, $c_i(t)$ is independent of $i$, like in the congestionless case, and as such, the analysis can be restricted to node 0:

$$c_0(t) = \frac{1}{2} \cdot \left(u_{t_l}(t) + u_{\theta_{101,0}(t)}(t)\right) \tag{32}$$

Furthermore, when restricting to edge 01, equations 24 and 25 become:

$$\dot{F}_{01}(t) = \frac{\frac{1}{2} \cdot u_{tl}(t)}{\ell((0,1),t)} - \dot{F}_{01}(\tau_{01}(t)) = \frac{1}{2} \frac{u_{tl}(t)}{\ell((0,1),0) + w_c T_{01}(t)} - \dot{F}_{01}(\tau_{01}(t)) \tag{33}$$

$$\dot{T}_{01} = \frac{1}{2}\left(u_{t_l}(t) + u_{\vartheta_{011,0,1}(t)+t_l}(t)\right) - \dot{T}_{01}(\tau_{01}(t)), \tag{34}$$

which, like equation 32, has already been summed over all possible paths $k$ and $n$ (again labeled with 1, as there is only one). By construction of $\tau$, $\theta$ and $\vartheta$, the following relations also hold: $\theta_{101,0}(t) = \tau_{10}(t)$ and $\vartheta_{011,0,1}(t) = \theta_{101,0}(t - t_d) + t_d$. The problem hence reduces to finding $\theta_{101,0}(t)$, after which the equations can readily be integrated.

In this simple example, this can be achieved by looking at the steady state amount of packages. This method is however not easily extendible to general networks, as it assumes all packages will be self-consumed or transported to the only other node, and hence the considered node has degree 1. A general network can clearly not satisfy this.

Let $x_{01}$ denote the amount of packages on edge $(0,1)$. This edge can, again, be considered alone, as the problem is symmetrical. The following equality holds:

$$x_{01} = \frac{x_0 + x_1}{2} - 2st_l = s \cdot \left(a + 2w_c\left(\frac{x_0 + x_1}{2} - 2st_l\right)\right), \tag{35}$$

where $s = s_{01} = s_{10} = s_{00} = s_{11}$, which are all assumed equal here, and $w_c$ is again the congestion weight. The first equality holds, by how the flow is constructed; packages closer to a node belong more to the closer node. On average, some random package on edge $(0,1)$ will hence belong half to node 0 and half to node 1. Similarly, half of the packages, that are not loading anymore, belonging to node 0 will be on edge $(0,1)$ and half will be on edge $(1,0)$. As $x_0 = x_1$, $\frac{x_0+x_1}{2}$ can simply be rewritten to $x_0$. Each time step, $s_{00} + s_{01} = 2s$ packages are generated on node 0, which results in $2st_l$ packages residing in node 0 until they have been consumed, or have left. Hence, the amount of packages on edge $(0,1)$ equals half the amount of packages belonging to both nodes, minus the packages that are actually still located on the node; the first equality.

The second equality requires a bit of prior knowledge; in steady state, the entire edge will be filled with flow. The average base amount of packages on that edge equals $sa$, because the base edge length is $a$ and the amount of package flow on an edge is dictated by the supply, $s$. The edge increases in length by $w_c$ times the amount of trucks on a node. This amount of trucks, necessarily has to be twice the amount of packages, which has just been determined in the previous paragraph, since trucks also have to return to their origin node, and returning

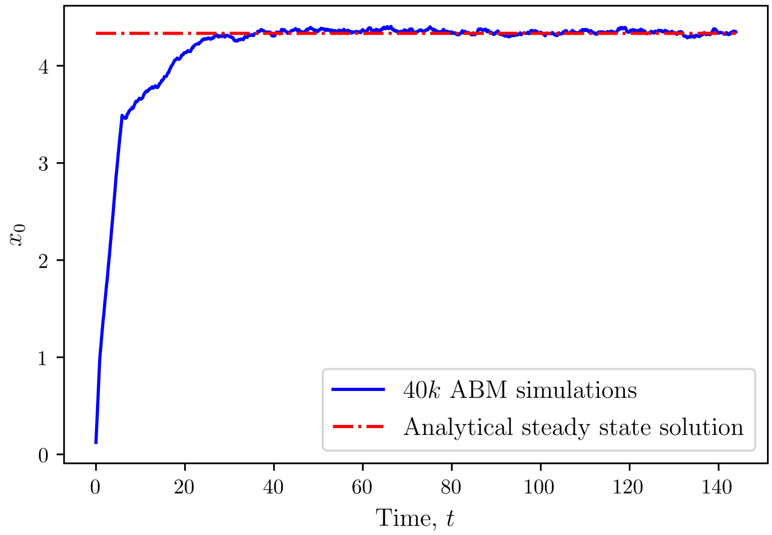


**Figure 4.3:** Time evolution of $x_0$ with congestion, for the line network. The congestion weight is $\frac{1}{4}$ per truck. The other parameters are the same as in Figure 4.2, refining by a factor 8, and hence some parameters are scaled accordingly.

trucks too count towards congestion. This second term hence deals with the congestion on edges, whereas the first equality is only concerned with the amount of packages on nodes. Solving equation 35 yields:

$$x_0 = \frac{sa}{1 - 2sw_c} + 2st_l = \frac{13}{3}, \tag{36}$$

when taking $s$ and $a$ as in figure 4.2, and $w_c = \frac{1}{4}$, as used in figure 4.3.

Figure 4.3 again depicts 40000 ABM simulations in blue, whereas the dashed red line shows Picard's fixed point method, used on equations 32 to 34, which coincides perfectly with equation 36. In the first iteration of Picard's method, congestion was assumed zero, which yields some steady state. In the second iteration, congestion is added, based on said steady state, and the edge lengths are updated, after which the steady state is again sought without congestion. This is done until convergence to the desired edge lengths. In this case, the edge lengths become $\frac{20}{3}$, namely $5 + \frac{2}{4} \cdot (\frac{13}{3} - 1)$, which is stems from the second equality in equation 35.

To somewhat diminish the discretization errors in the ABM solution, the amount of time steps has been refined 8 times. Of course, as it only determines the steady state, Picard's method does not convey the start-up process, which the ABM displays.

The EBM can be integrated too, but now instead of only having to consider the $x_i$ and $F_{ij}$, the $T_{ij}$, $\tau_{ij}$ and $\theta_{ijk,l}$ have to be considered as well. This makes the equation much slower to integrate, which however does not yet really come through in this simple example.

Figure 4.4 shows the difference between the same 40000 ABM simulations as in figure 4.3 in blue, and the numerically integrated EBM solution in dashed red. The EBM here also includes the start-up process, where the Heaviside functions reflect their behaviour, as $x_0$ grows from approximately 3.5 to the previously calculated $\frac{13}{3}$. The first bend happens due to the packages arriving at the nodes. After that, the function does not reach a steady state immediately, but the trucks necessarily have to return to their origin, which slowly adds congestion to the edges, which in turn increases $x_0$ until a steady state has finally been reached. It should however be noted that there is a small error between both solutions at steady state.

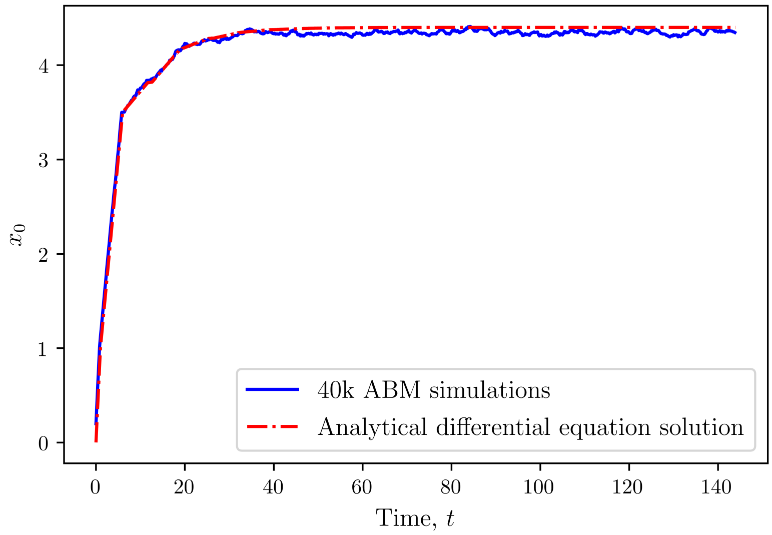


**Figure 4.4:** Time evolution of $x_0$ with congestion. All parameters are the same as in figure 4.3. The ABMs have been run with time step refinement 5, while the EBMs have been numerically integrated with refinement 1000.

Besides finding the steady states and the analytical solution, it is also interesting to look at behaviour when changing parameters in the model. For instance, if $w_c$ and $s$ grow, there will again be a point when congestion grows out of control. Clearly, by equation 36, $2sw_c$ cannot reach 1, as by then on each time step trucks increase the edge length by one. Taking $s = \frac{1}{2}$ and varying with respect to $w_c$ gives figure 4.5. This has been generated by repeatedly performing the Picard iterations for different values of $w_c$ and finding their respective steady state $x$. The same procedure can be used to find a bifurcation point, for which the system commences to diverge.

This figure gives the expected behaviour; for $w_c = 0$, steady state $x_0$ equals $\frac{7}{2}$, which is in accordance with the results in section 4.1, while on the other hand, $x_0$ diverges for $w_c \to 1$. $2sw_c = w_c = 1$ then constitutes the bifurcation point, at which the system starts to diverge. It consequently also turns out that for increasing $w_c$, the results contain a larger error (not shown), which happens due to the edge lengths receiving larger jolts of length increase each time a truck arrives on the edges. The size of those jolts directly cause a larger variance and hence error.

This chapter has covered all possible analysis on the simple line network, which consists of solving the equations and running ABM simulations with and without congestion, and determining for which congestion constant the system inevitably diverges. This analysis indicates that the models work well and complement each other and that the results are conceivable. The next chapter performs the same analyses for larger networks, which more clearly shows how useful both models are in a more general situation.

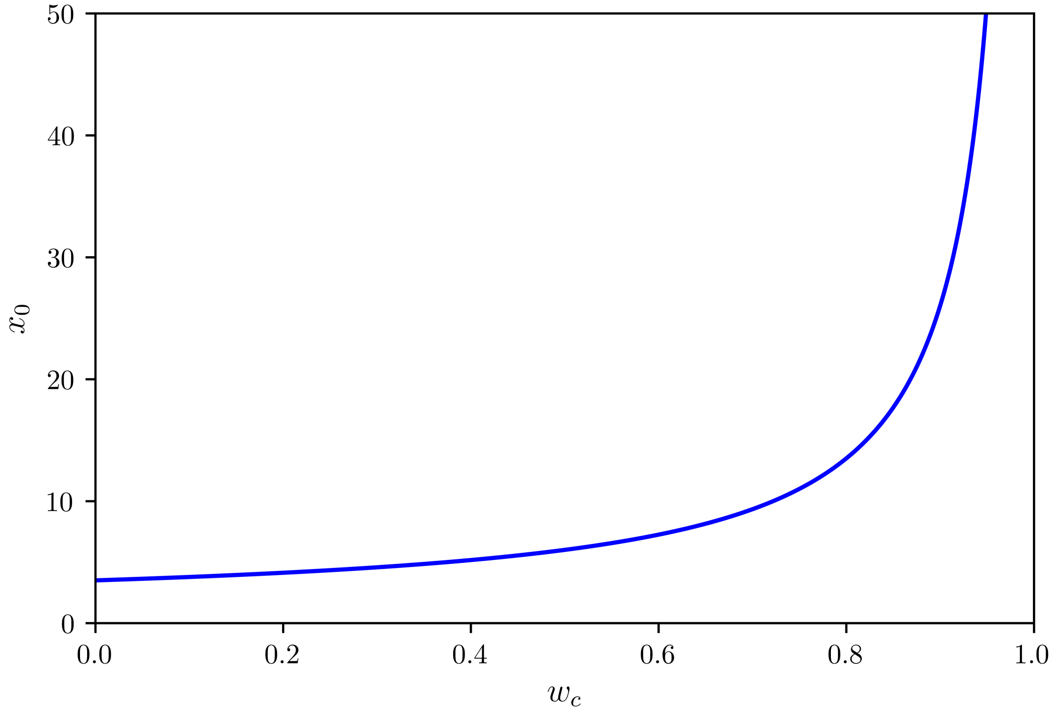
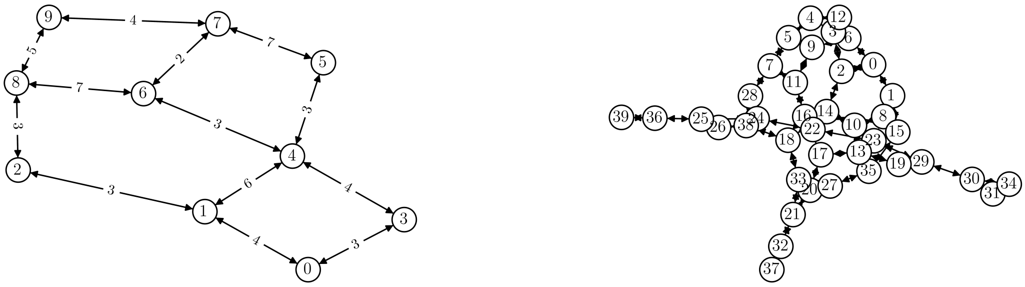


**Figure 4.5:** Steady state $x_0$ for different congestion weights. $s$ has been chosen $\frac{1}{2}$.

# Chapter 5: Larger networks

While it is illustrative to see how the EBM and ABM give similar results for a simple network, where everything is calculable anyway, the real usefulness comes through in larger networks. Hence, to verify whether the models are extendible and scalable for more complex situations, both have been ran for networks ranging in size from 10 to 80 nodes. As the network size increases, it becomes increasingly hard to find the functions in the EBM, and hence also its explicit solution.

In particular, the main focus has been on a custom-made 10 node graph (figure 5.1a), and on a map-like 40 node graph, where each node has maximum degree 3 (figure 5.1b), meaning each node connects to at most three other nodes. The 10 node graph has the benefit of being tailored such that it has one rather decentralized node, namely node 5, while it also contains two hubs, nodes 4 and 6. By being not a very large network, individual node effects can be ruled out more clearly. On the other hand, the larger network allows to explore whether the models are able to perform well for larger networks. In addition, this particular network brings a stronger contrast between centralized and decentralized nodes, such as the ones on the very left and on the very right.



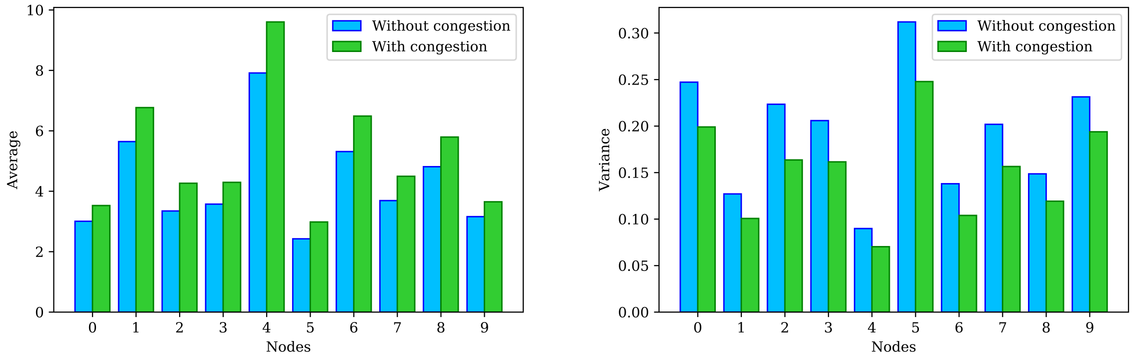
**(a)** The custom-made network.

**(b)** The map-like network.

**Figure 5.1:** The mentioned tested networks: 5.1a shows the custom-made 10 node graph, with edge lengths added in the middle of the edges. 5.1b depicts the 40 node graph, where nodes have maximum degree 3. Travel times, or equivalently, the length of the links between two nodes are Poisson distributed with average 7 on top of a fixed travel time of 3 units. Hence, the total travel time is on average 10. One has to bear in mind that direction and scale do not have any meaning in these images.

In addition to this, some other graphs were regarded, albeit not to the same extent. Their respective structures and results are presented in section 5.3.

Of course, as a network does not generally wield a symmetric or simple structure, all nodes behave differently now. Centralities and degrees of nodes differ and become an important factor that determine the node dynamics, along with the outgoing and ingoing edge lengths. More central nodes will in general experience more traffic, and hence contain more packages and send more outgoing flow. This will be touched upon in the coming sections.



**(a)** The package average.

**(b)** The percentual variance per simulation.

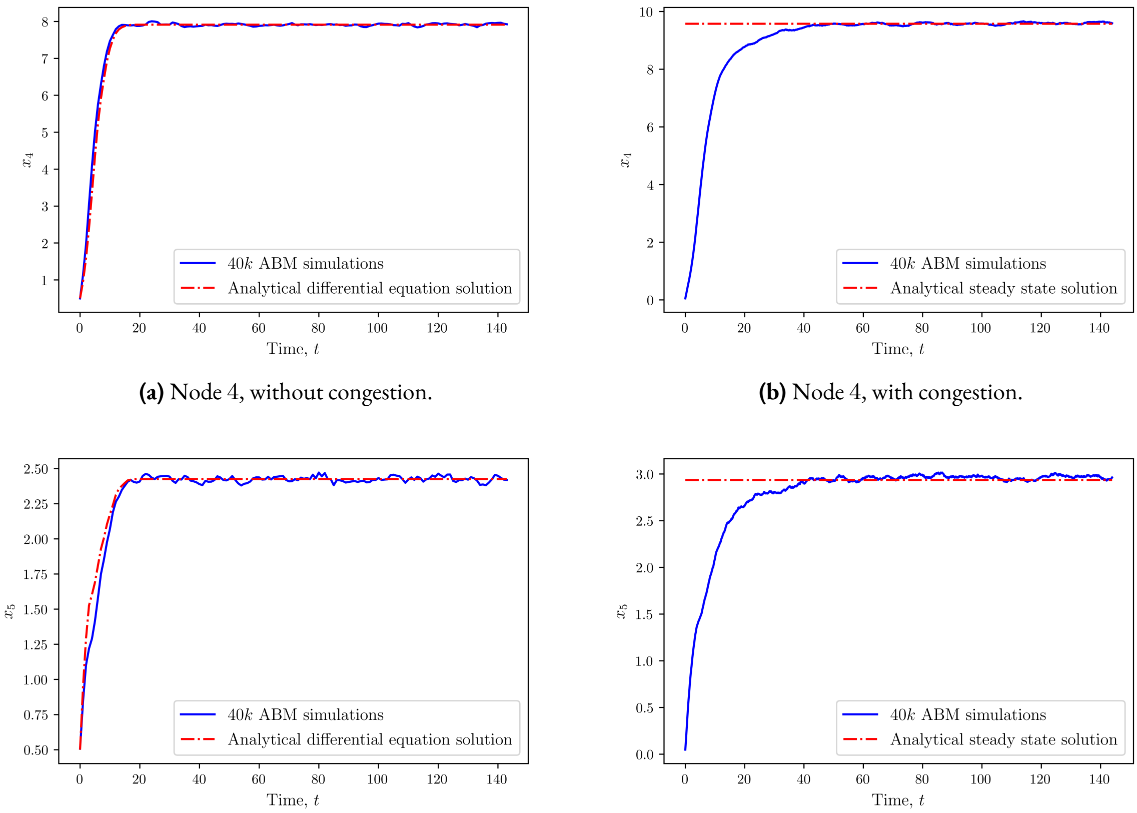
**Figure 5.2:** The package average and percentual variance per simulation for all nodes, with and without congestion, for the custom-made 10 node network. The variance is the squared error of a single ABM simulation relative to the analytical steady state. The simulations were run for 144 timesteps, refined five times in the simulations with congestion. The steady states have been chosen as $t > 20$ and $t > 250$ time steps respectively, as by then no significant grow seemed to occur. There is an abundant amount of trucks ($10000$) and there is on average $0.05$ demand for every commodity on every node on every time step. $40000$ simulations were run, with $t_l = t_d = 1$. In the simulations with congestion, the congestion weight was chosen to be $\frac{1}{4}$.

## 5.1 The custom-made network

Let us first turn attention to the custom-made network depicted in figure 5.1a. Unlike with the line network, a plot of the amount of package on a single node will not contain enough information anymore. However, showcasing the package amounts for all the nodes is not illustrative either, as it either contains multiple figures, which are tedious to process, or too much happens in a single visualisation. Rather, the behaviour will be depicted by showing the average and variance in steady state for all nodes and for all time steps, when compared to the results obtained with Picard's iteration method. The average and variance are only taken over the steady state as the Picard iterations only yields the steady state solution. To keep things consistent between the results with and without congestion, the analysis will be done based on the steady state for both. The comparison to the EBM is performed later in this section.

Let us turn to the distribution of packages among the different nodes. Figures 5.2a and 5.2b depict the steady state average and variance between the models, for the custom-made network. One can see that node $5$ has the highest variance, followed by node $0$, $2$, $3$, $7$ and $9$, while nodes $1$, $4$, $6$ and $8$ have the lowest variance. Conversely, the opposite is true for the averages on those nodes. The nodes with the highest percentual variance contain the lowest amount of packages. This is partly due to their centrality in the network; as there are more paths going through the central nodes, more trucks pass through those nodes and hence the amount of trucks averages out better. Moreover, it seems like, besides closeness and betweenness centralities, the degree and non-shortest outgoing path lengths contribute to the differences in the variances. One should bear in mind that, as a rule of thumb, when nodes are less central, they will contain fewer trucks and less traffic. As a result, the percentual variance on them will become larger.

We also see in Figure 5.2 that adding congestion basically scales the averages and variances. This can be explained by realizing that the amount of trucks in transit will increase once congestion is added. If the congestion is sufficiently large, enhanced spreading of the packages will occur, and simple scaling is no longer valid.



**(a)** Node 4, without congestion.

**(b)** Node 4, with congestion.

**(c)** Node 5, without congestion.

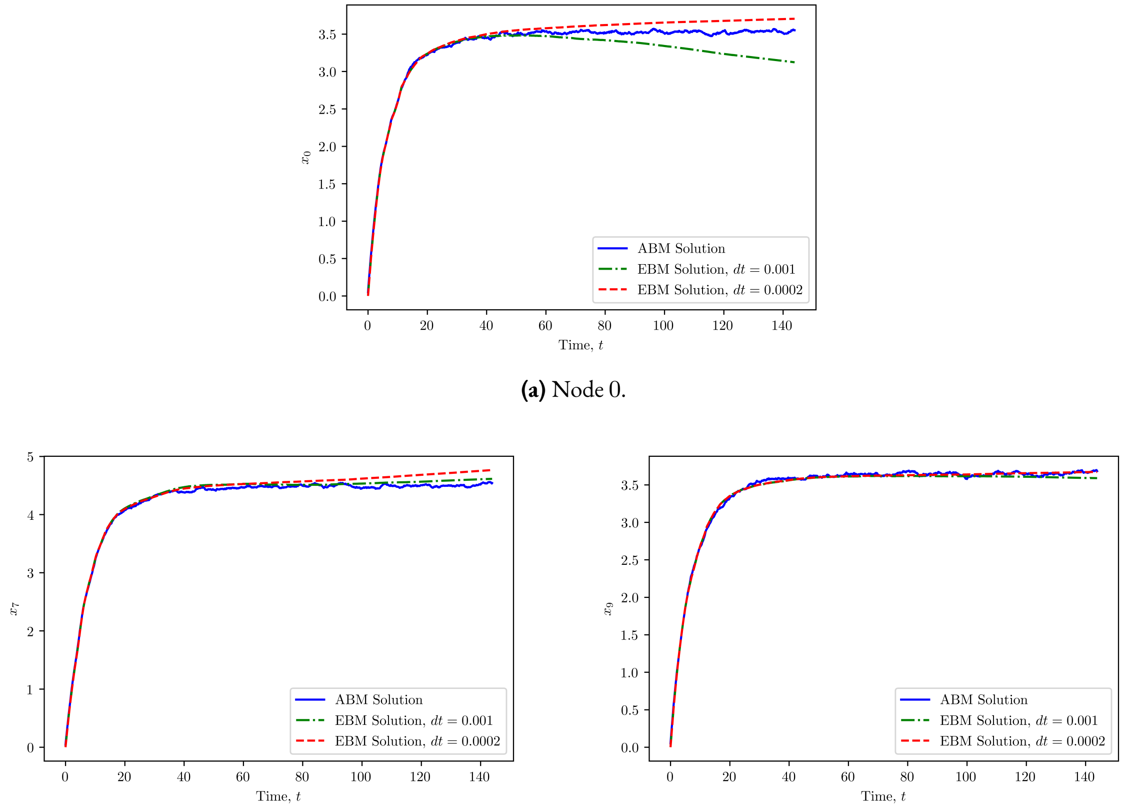
**(d)** Node 5, with congestion.

**Figure 5.3:** A visualization of the ABM and EBM solutions without congestion for nodes 4 and 5 in the custom-made network, and a visualization of the ABM solution, for the models with congestion, with its respective steady state Picard method. The simulations with congestion were run with a time step refinement factor 10.

The differential equation approach closely follows the simulations from the initial value to the steady state. The difference between the two results is small for the most connected nodes, whereas there is, more so at the beginning, some discrepancy for the less connected nodes. Figure 5.3 shows that the ABM simulations without congestion agree with the EBM and that the simulations with congestion give the same steady state as with Picard iteration, used on the analytical equations; these show examples of individual nodes and mirror what can be seen in figure 5.2 in a different way. This gives a strong indication that this ABM can be modelled well with the EBM without congestion and that the steady state solution can also be found rather accurately.

Figure 5.3 identifies a difference in behaviour between the ABM and EBM on different nodes. For node 4, the ABM and EBM solutions practically overlap, whereas the Picard iteration method also pretty much overlaps with the steady state of the congestion-included ABM. Node 5 has a much higher variance overall, which is in accordance with the results from figure 5.2 and the start-up does not completely overlap in the congestionless case.

The customized network runs its simulations for 144 time steps with and without congestion in about 13.5 and 56 (roughly four times as much) minutes respectively. Above treated simulations with congestion and 5 times refinement took about 3 hours (rougly three times as much time). These have all been run on an AWS server, using all 8 2.3 GHz cores to run parallel simulations. The congestionless EBM and Picard's method for the EBM with congestion take seconds to solve instead.



**(a)** Node 0.

**(b)** Node 7.

**(c)** Node 9.

**Figure 5.4:** A comparison of an average over 40000 ABM simulations and numerically integrated EBM solutions with congestion, performed for several nodes. To demonstrate the influence of the time step refinement, the EBM solutions have been presented for $dt = 10^{-3}$ (dash-dotted green) and $dt = 2 \cdot 10^{-4}$ (dashed red).

Furthermore, when actually integrating the EBM with congestion, it is evident that they do not entirely overlap, as the EBM solution diverges, as can be seen in figure 5.4. There might be a couple of reasons as to why this instability occurs. The culprit might be the non-existense of the $\tau$, $\theta$ and $\vartheta$ for some times, or the many included Heaviside functions, with shifts that aren't divided by the integration time steps. It could also be a lacklustre implementation, although it has been checked multiple times, and I found no mistakes. The error usually appears to reduce for smaller time steps, though. This time, the numerical integration takes about 27 hours for 144 time steps, which has been refined to $dt = 0.0002$ time steps. The required memory is enormous, in the order of 20 GB, as so many things have to be kept track of, the worst of which is the large list of possible paths between nodes and their probabilities, for all time steps. While there is plenty of room for optimizations in the implementation, it will probably not provide an efficient means to achieve the desired solution, more so as it currently does not properly converge for this 10-node network. It will thenceforth become increasingly challenging to find converging solutions for larger networks. For this reasfioon, and because it takes a lot of resources, this equation won't be solved numerically for other networks in later sections.

Figure 5.4 depicts a comparison of the ABM simulation results and the EBM integration results for different nodes in the network. They have been elaborately chosen such that they show different behaviours; for $x_0$, the EBM solution becomes better as $dt$ is lowered. For $x_7$, the numerical approximation becomes a bit worse as $dt$ is lowered, and for $x_9$ both solutions are already pretty good, but lowering $dt$ improves the accuracy a towards the later time steps. All three nodes have one thing in common, though: the solutions for earlier time steps, where the network is still in its start-up process, pretty much overlap. The solutions for other nodes, with exception

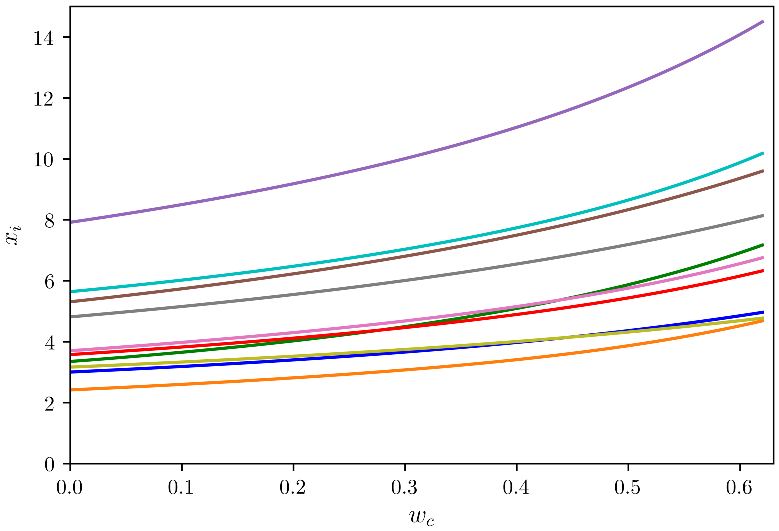


**Figure 5.5:** The different steady state $x_i$ for all nodes in the custom-made network, as a function of $w_c$. The figure stops at $w_c = 0.6299$, as just beyond that point the system diverges.

of node 5, show similar behaviour as node $x_0$. For node 5, the behaviour does not improve upon decreasing $dt$.

Discretization errors should be resolved with smaller time steps. Reducing the size of the time steps here, usually yields significantly better results, which indicates no error in the implementation. In each case, as mentioned, the numerical solution, closely follows the ABM results up to the initial bend, which constitutes the start-up process. A low effort option to improve the accuracy could then be to consider two time scales; the start-up time scale and the steady state time scale. At some point, the steady state solution is reached after the start-up. The steady state Picard solution could then be glued to the EBM-obtained start-up solution, which makes the EBM implementation cheaper, as much less time steps have to be considered, while preserving accuracy on the steady state, on which the solution loses accuracy.

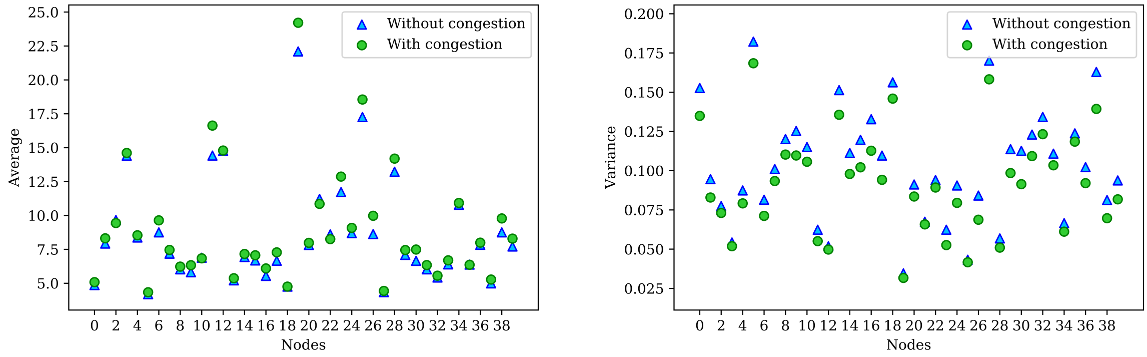
Lastly, just like in figure 4.5 of the previous chapter, a bifurcation $w_c$ can be found with the same procedure. This yields figure 5.5.

In contrast to the diagram for the line network, the individual $x_i$ curves do not diverge here, but reach some threshold value. Because there are multiple paths available to be chosen, and their probabilities of being chosen depend on the lengths of other paths, they create more of a buffer, that can accommodate the increase in packages on nodes. As a result, the increase of steady state package amount does not gradually grow until infinity, but rather reaches some value for each node, after which it abruptly jumps to infinity, such that the system suddenly becomes unstable. This value seems to lie between $0.62993485 \leq w_c \leq 0.62993535$.

The custom-made network has been analyzed, and the analysis shows that the EBM is insufficiently accurate, when congestion is added to the problem. The other results seem consistent with expectations, though. In the next section, the analysis is extended to a larger network, to see how the models scale.

## 5.2 The large map-like network

An obvious way to pursue the comparison between ABMs and EBMs is to extend the research to larger networks. Hence, let us move on to the map-like 40 node network and investigate the performance of both models here.



**(a)** The package average.

**(b)** The percentual variance per simulation.

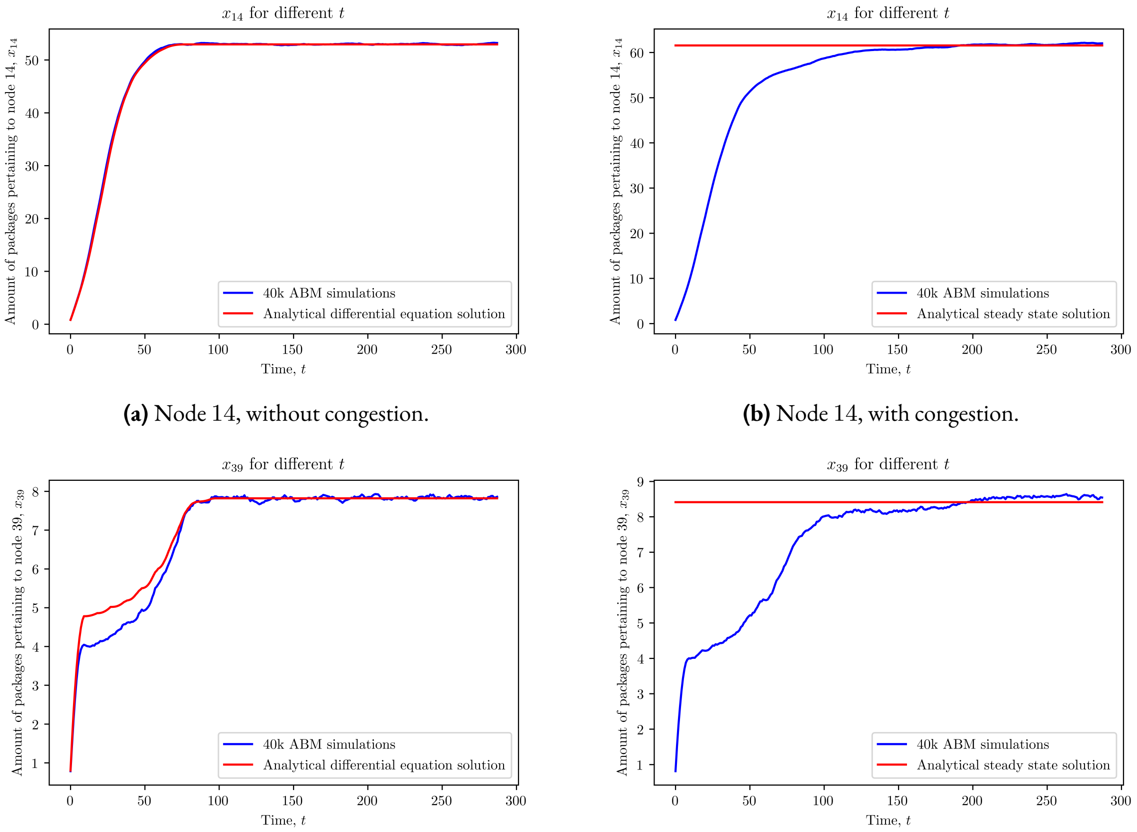
**Figure 5.6:** The package average (5.6a) and percentual variance (5.6b) per simulation for all nodes, with and without congestion, for all the nodes in the network from Figure 5.1b. The simulations were run for 288 timesteps and were depicted for respectively $t > 100$ and $t > 200$ time steps. There is again an abundance of trucks (10000) and there is on average 0.02 demand for every commodity on every node on every time step. This time, only 20000 simulations were run, with $t_l = t_d = 1$. In the simulations with congestion, the congestion weight was chosen to be 0.05.

It is clear here, when comparing Figure 5.6 to Figure 5.1b that both with and without congestion, the less central nodes have a remarkable amount of variance in the packages. Also, the variance grows significantly, when adding congestion. Mostly, this is in accordance with the smaller network. This implies that the simulations work well for different kinds of networks and larger networks, which means that the earlier discussed ideas might scale up to larger networks.

It has to be mentioned, though, that some nodes present slightly different behaviour from the conclusion in section 5.1; it is not true that congestion always increases the package average here. It however always decreases the percentual variance. This might be evidence of paths being rerouted by ongoing traffic, which leads to less usage of certain edges, which in turn doesn't increase the package average on the connected nodes.

A strange thing stands out when looking at solutions for $x$, for different nodes, as given in figure 5.7. Without congestion, the solutions for node 14 overlap well. However, for node 39 the ABM solution starts off significantly lower than the EBM solution. What's weird, though, is that the solutions both end up at the same steady state. One can also immediately see that the ABM simulations for node 39 are way more jagged than the solutions for node 14. That is no coincidence, as they have been chosen to represent central and secluded nodes. Node 39 is only connected to its sole neighbour 36, which is already at the outside of the network. Node 14 is, of course, only connected to three other nodes, but those are all also connected to three other nodes, which makes node 14 much more central than node 39. The discrepancy present for node 39 implies that the EBM lacks certain interaction, which affects the start-up process for the least central nodes. This probably also explains the behaviour of node 5 in the custom-made network, presented in the previous section. However, the error does not persist in the steady state, such that the model somehow corrects itself along the way.

The larger network simulations, with 40 nodes, took about 6 hours and 21 hours, with and without congestion, respectively. This means that as the amount of nodes increases past 40, the runtime of the simulations dwindles until this model eventually is not feasible to run anymore, until some optimizations are added. As the network size increases, the ABM simulations will take much more time to run than solving the congestionless differential equations. There's also a big difference in time between simulations with and without congestion, as updating the travel times takes a lot of processing power. However, the benefit of simulations is that they give



**(a)** Node 14, without congestion.

**(b)** Node 14, with congestion.

**(c)** Node 39, without congestion.

**(d)** Node 39, with congestion.

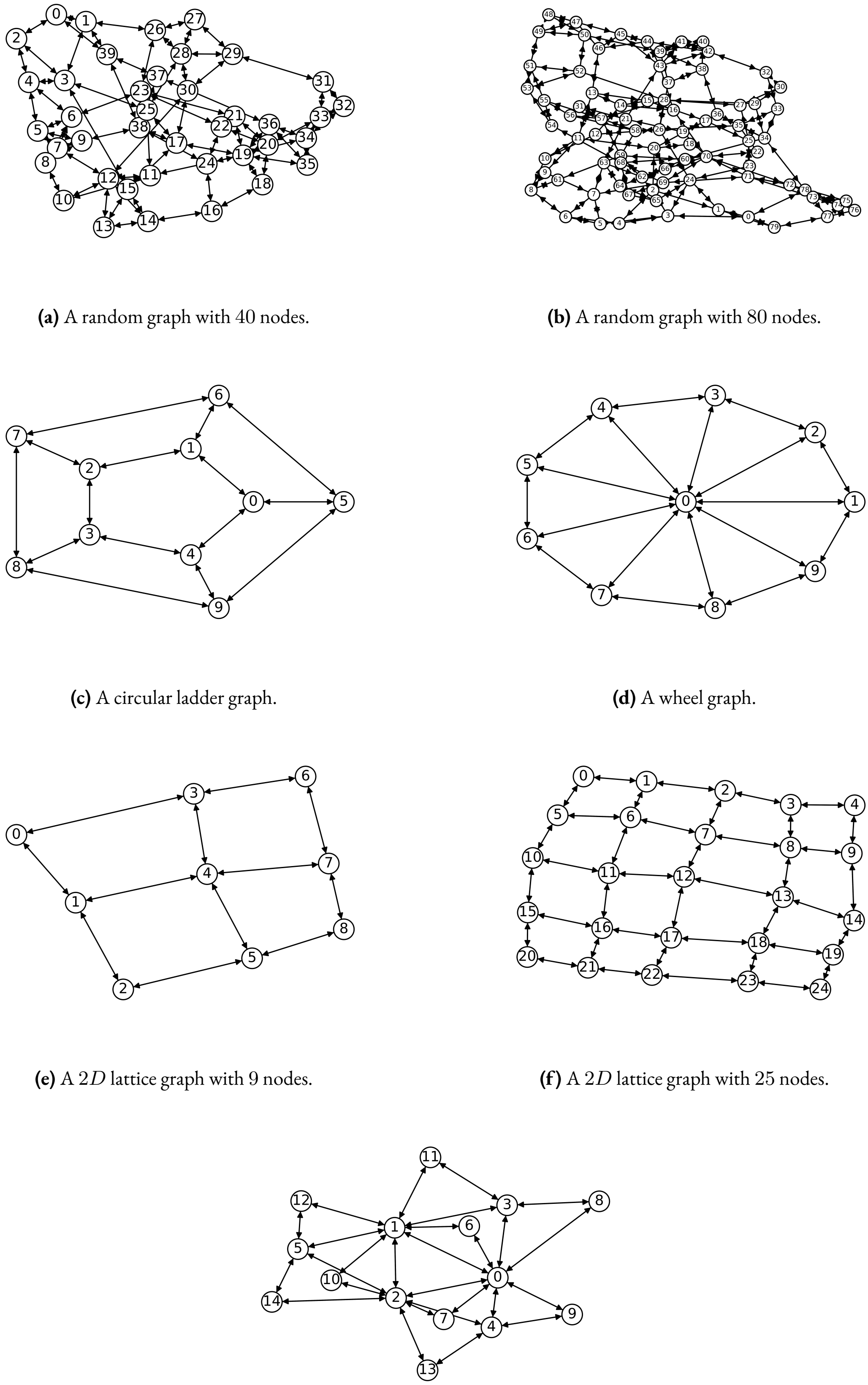
**Figure 5.7:** ABM and EBM solutions for nodes 14 and 39 in the larger map-like network, having used the same parameters as in figure 5.6.

insight on the individual interactions and generate a variance for the amount of packages, like demonstrated, given that the parameters are chosen realistically. Evidently, the EBM with added congestion is left out, as it did not perform well enough in the 10 node network, and surely won't perform better for 40 nodes.

## 5.3 Other networks

In addition to the custom-made and maximum degree 3 networks, several random graphs have been regarded, along with smaller networks such as a circular ladder graph, a wheel graph, lattice graphs and a particular kind of random graph; the Dorogovtsev-Goltsev-Mendes (or DGM) graph, as depicted in figure 5.8. Most of these networks have been considered as they represent part of some real world network structure, with exception of the random networks, which have been taken to test whether the equations work for arbitrary large networks. For instance, the wheel graph models a giant hub, while the lattice graph is a good approximation of cities such as new york. Many paths in such a graph are favourable and the middle nodes are also very central. The circular ladder graph represents a situation where another parallel path to the destination exists. The DGM graph is a kind of small world network, where the node degrees follow a power law. Each network thus captures some specific behaviour and allows for study of this behaviour alone.

Of these, the wheel, $2D$ lattice and DGM graphs are the most interesting to treat separately. The circular ladder graph does not give any striking results, and the larger $2D$ lattice graph is just a more elaborate version of



**(a)** A random graph with 40 nodes.

**(b)** A random graph with 80 nodes.

**(c)** A circular ladder graph.

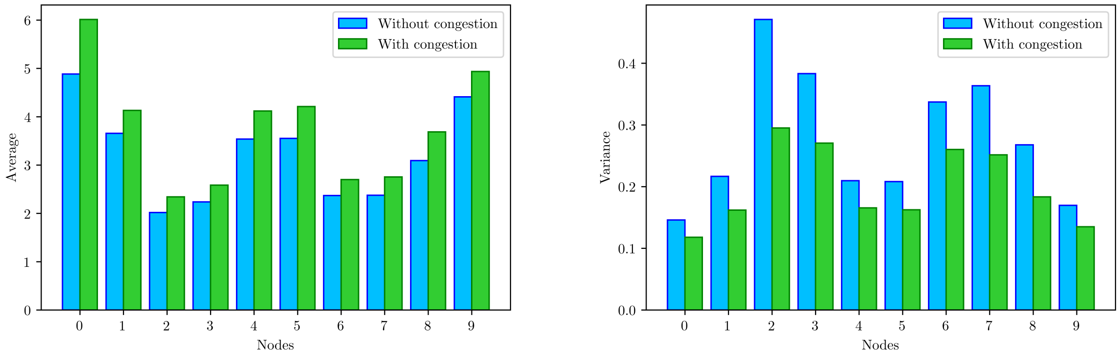
**(d)** A wheel graph.

**(e)** A $2D$ lattice graph with 9 nodes.

**(f)** A $2D$ lattice graph with 25 nodes.

**(g)** A DGM graph with 15 nodes.

**Figure 5.8:** Graphs for the additional networks under consideration, but for which no extensive analysis was performed.



**(a)** The package average.

**(b)** The percentual variance per simulation.

**Figure 5.9:** The package average and percentual variance per simulation for all nodes, with and without congestion, for the wheel network. The simulations were run for 144 timesteps, refined ten times in the simulations with congestion. The steady states have been chosen as $t > 20$ and $t > 40$ time steps respectivelyr. There is an abundant amount of trucks (10000) and there is on average 0.05 demand for every commodity on every node on every time step. 40000 simulations were run, with $t_l = t_d = 1$. In the simulations with congestion, the congestion weight was chosen to be $\frac{1}{4}$.

the smaller one, but just a bit more chaotic due to the additional 16 nodes. These will thenceforth be omitted. The remainder of this section will mostly be structured in a way such that the interesting results from each treated network are presented and afterwards briefly discussed.

### The Wheel network

The first network under consideration is the wheel network. In this particular case, the edge lengths of the network have been generated with length $1+$ Poisson$(5)$ for the spokes, and $1+$ Poisson$(1)$ for the periphery. In this case, the package averages and variances are given in figure 5.9.

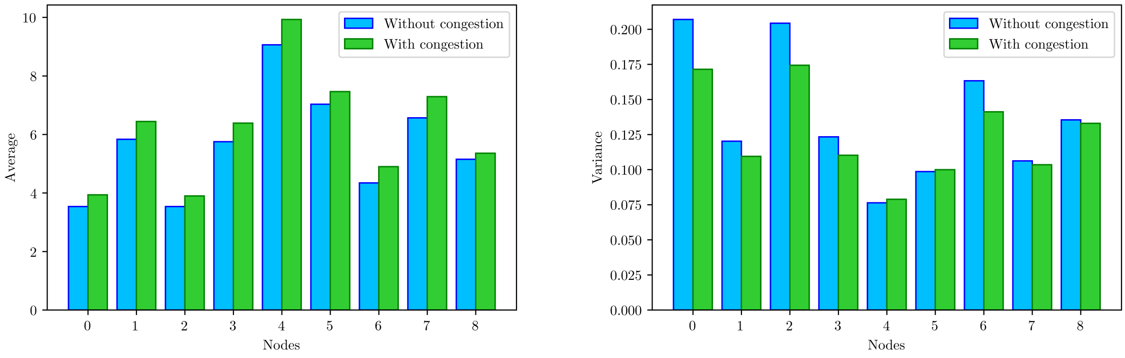
This figure shows that node 0, which is the center of the wheel, always contains the most packages, albeit not by a lot. As the spokes are relatively long, that is not strange. For smaller spokes, this average is predicted to increase. When congestion is present, though, the amount of traffic through the hub increases, as the edge lengths on the periphery proportionally increase a lot. One should expect that when the spokes are long, they become relatively more favourable when congestion is added, but when the spokes are short, they become relatively less favourable when congestion is added. As seen in the previous networks, the variance increases as package average decreases.

It however appears that no node in this network presents particularly interesting behaviour, except nodes 3, 6 and 7 exhibiting a slight discrepancy between the congestionless solutions, just like in figure 5.3c. Moreover, the variance in node 2 is disproportionally high, but that probably has to do with a lacklustre refinement. However, these two mentions do not need a figure on their own, as their results are nothing special.

### The $2D$ Lattice network

Up next is the $2D$ lattice network. Figure 5.10 shows the package averages and variances, as usual, and figure 5.11 shows the behaviours of different nodes in the network. This time around, all edge lengths have been generated $1+$ Poisson$(5)$.

The familiar result that central nodes have a high average and lower percentual variances is yet again verified here. Of course, there is often little reason for packages to pass through the corner nodes 0, 2, 6, and 9, while



**(a)** The package average. **(b)** The percentual variance per simulation.

**Figure 5.10:** The package average and percentual variance per simulation for all nodes, with and without congestion, for the $2D$ lattice network. The simulations were run for $144$ timesteps, refined ten times in the simulations with congestion. The steady states have been chosen as $t > 30$ and $t > 80$ time steps respectivelyr. There is an abundant amount of trucks ($10000$) and there is on average $0.05$ demand for every commodity on every node on every time step. $40000$ simulations were run, with $t_l = t_d = 1$. In the simulations with congestion, the congestion weight was chosen to be $\frac{1}{4}$.

packages often have more reason to pass through the periphery of the wheel graph for instance.

For nodes $0$ and $4$, we observe a small difference in the start-up process. The time evolution of node $4$ has an almost continuous line, with very little variance, whereas for node $0$ there is a small dent just before reaching steady state, occurring for most non-central nodes in some way. This dent originates from the network topology, in which edges traversed in a path is often constant for a origin-destination pair. For instance, when going from $8$ to $0$, one has to move both left and up twice. However, these may be arranged in any way, but they necessarily pass four edges (unless the edge lengths are very weirdly generated).

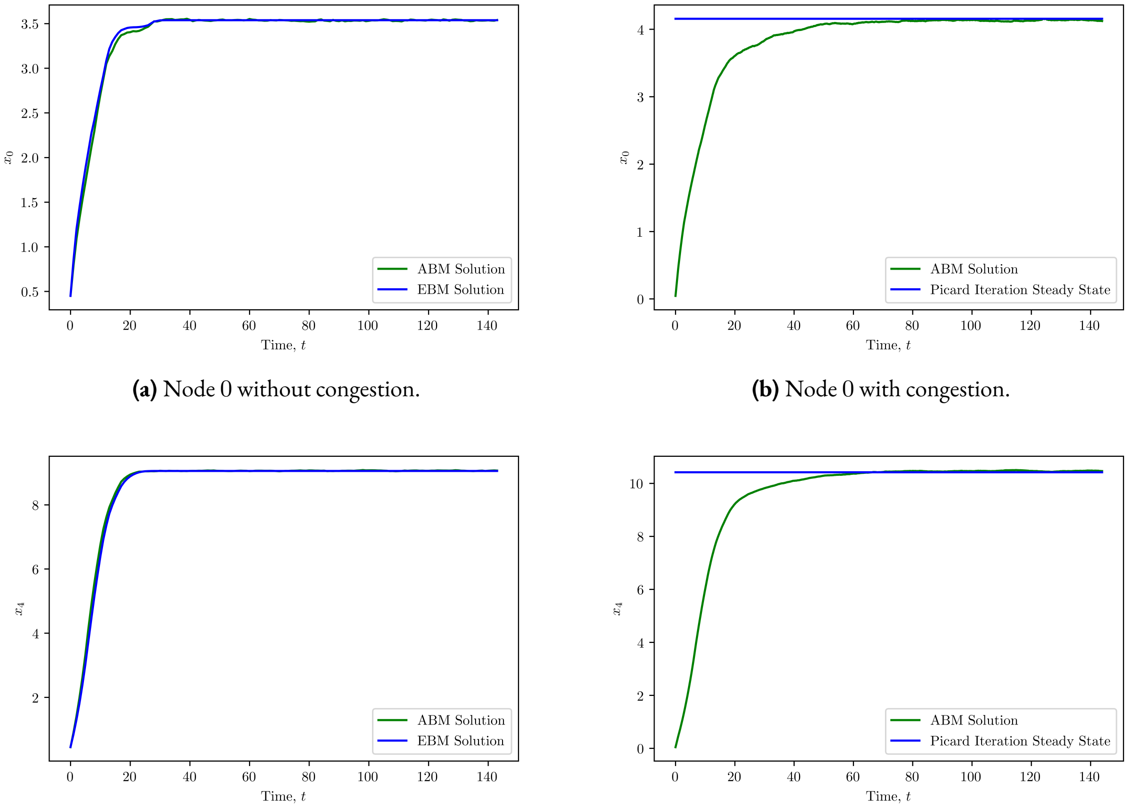
In general, this behaviour can be extrapolated to larger lattice networks. When considering the network from figure 5.8f, this dented behaviour can be expected for nodes $0$, $4$, $20$, and $24$ and to a lesser extent for the remaining boundary nodes. The centre of the graph, consisting of node $12$, but also its surrounding nodes, should also generally experience more traffic than the outside. One should, just as visible in figure 5.10a, expect that the package average decreases as the amount of edges needed to reach the centre increases.

### The DGM network

Last up is the DGM network, which is some way of generating a small world network. This is a network that often describes social networks, in which some influential hubs are much more connected than the other nodes, but this can be extended to, for instance, traffic on websites as well. Figure 5.12 shows the usual package averages and variances for the nodes in the DGM network, in which the edge lengths have been generated $1+$ Poisson$(5)$.

The way a DGM graph is generated, is by connecting new nodes to two existing nodes, with higher probability of them being connected to a more central node. As a result, the newest nodes (which are the ones with largest index) are the least central, and have degree $2$. This holds for nodes $6$ all the way through $14$. This is also clear when looking at figure 5.12a, in which this distinction can be inferred by looking at the package average. The most central nodes $0$ to $2$ have a much larger average than nodes $3$ to $5$, which also have a significantly larger average than the remaining nodes. Node $8$ is an outlier, but that happens due to an unfavourable generation of its outgoing edge lengths, and by extension its unpopularity.

Again, time evolution of the amounts of package at the nodes is not very interesting, except that nodes $6$



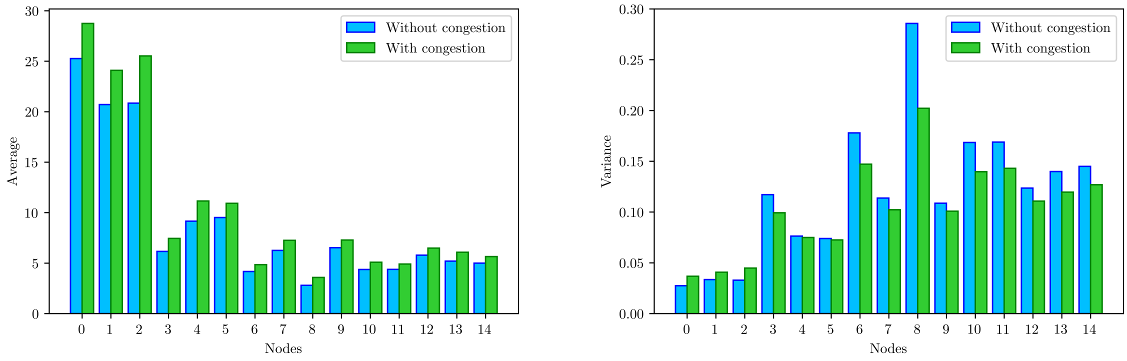
**(a)** Node 0 without congestion.

**(b)** Node 0 with congestion.

**(c)** Node 4 without congestion.

**(d)** Node 4 with congestion.

**Figure 5.11:** The time evolution of nodes 0 and 4 in the $2D$ lattice network.



**(a)** The package average.

**(b)** The percentual variance per simulation.

**Figure 5.12:** The package average and percentual variance per simulation for all nodes, with and without congestion, for the DGM network. The simulations were run for 144 timesteps, refined ten times in the simulations with congestion. The steady states have been chosen as $t > 30$ and $t > 70$ time steps respectively. There is an abundant amount of trucks (10000) and there is on average 0.05 demand for every commodity on every node on every time step. 40000 simulations were run, with $t_l = t_d = 1$. In the simulations with congestion, the congestion weight was chosen to be $\frac{1}{4}$.

through 14 show the familiar discrepancy in the start-up process between the congestionless ABM and EBM solutions. Beyond that, the time evolutions only mirror what can already be seen from figure 5.12.

# Chapter 6: An optimization approach

In the preceding chapters, the agent-based model and differential equation model were both developed and tested. The congestionless EBM described the ABM well. The performance of the EBM with congestion was lacklustre, though. At the very beginning of the development of the EBM, equations 16 and 17 did not seem to yield the desired results on their own. After developing the EBM and finding out that creating the EBM from the bottom up cost way too much performance, there were two options. Either, the EBM was to be repaired, which is a hard thing to do when the error is not obvious, or at least an equation can be created, which is made to approximately yield the averaged ABM solution as its solution. The paper by Cencetti et al. [15] sparked this idea, as it considered equations with a very similar form as equation 16 in a similar setting, and hence maintains the desired simplicity of the equations.

In said paper, the authors entertain the situation where commodity is placed on a network and attracted in some way to sinks, in which they will be trapped forever. Their commodity is denoted by $p$. In one time step, these commodities can hop to a neighbouring node, the odds of which are encompassed by matrix $M^{[i,j]}$, where $i$ and $j$ are the trapping sinks. Taking the continuum limit, this gives a continuous equation. In their notation, said equation generally reads:

$$\dot{p}_k(t) = \sum_l M_{kl}^{[i,j]} p_l(t) - \sum_l M_{lk}^{[i,j]} p_k(t) = \sum_l L_{kl}^{[i,j]} p_l(t), \tag{37}$$

where $L_{kl}^{[i,j]} = M_{kl}^{[i,j]} - \delta_{kl}$. This is true, as it is assumed that $\sum_l M_{lk}^{[i,j]} = 1$, which means that all the available commodity moves, and nothing remains on the current node. However, even if a proportion of some commodity $l$ remains on the node, this could be incorporated by letting $M_{ll}^{[i,j]} \neq 0$.

This model is very similar to the model in section 3.1. Mainly, the $F_{ij}$ terms there, with the assumption $F_{ij}(t) = \alpha_{ij} x_i(t)$, correspond perfectly well to the $\sum_l M_{lk}^{[i,j]} p_k(t)$ term. There are two differences though; for instance, there are no clear sinks in the model presented in chapter 3.1. This is not necessary for the analysis presented in the paper to work either, and as such it can in general be disregarded. Moreover, the paper does not consider supply and demand terms. The supply is generated beforehand, and is included in the initial condition; $p_k(0) = p_0$. The demand is undefined as it does not especially matter how many packages reach the sinks, as long as everything is drained from the remainder of the network in the end.

In the treatment of this problem in this current chapter, sink nodes are omitted, while they are surely still allowed to exist (think of a package distribution centre in a supply network, for example). Supply and consumption terms are added instead. The difference is that when a commodity is consumed, it is not necessarily trapped at the node.

The following equation is thence obtained:

$$\dot{p}_i = \sum_j M_{ji} p_j - \sum_j M_{ij} p_i + s_i - c_i, \tag{38}$$

where $M_{ij}$ is equivalent to the former $\alpha_{ij}$ and denotes the fraction of packages flowing out of node $i$ to node $j$. $s_i$ and $c_i$ are the supply and consumption terms on nodes. In the case a sink exists on node $i$, $M_{ij} = 0 \ \forall j \neq i$, which ensures that no packages leave the sink. If a source exists on node $i$, $M_{ji} = 0 \ \forall j \neq i$, which ensures that no packages are transported into the source.

The only constraint on this equation reads

$$\sum_i (s_i - c_i) \overset{t \to \infty}{\longrightarrow} 0, \tag{39}$$

which makes sure that there is no divergent behaviour.

This generalization of equation 37 does not allow for it to be rewritten to a simple $\dot{p}_i = \sum_j L_{ij} p_j$, as the inhomogeneous terms, $s_i$ and $c_i$, do not permit that. 38 can be written in a tidier form:

$$\dot{p} = (M - I)p + s - c = Ap + s - c \tag{40}$$

The homogeneous part of this equation has the solution

$$p_h(t) = e^{At} p_0, \tag{41}$$

which can further be rewritten as

$$p_h(t) = \sum_n \sum_k^{m(\lambda_n)} \mathscr{A}_{nk} t^{m(\lambda_n)} p_0 e^{\lambda_n t}, \tag{42}$$

in which $\lambda_n$ are the eigenvalues of $A$ and $m(\lambda_n)$ is the multiplicity of said eigenvalue. $\mathscr{A}_{nk}$ are matrices, determined by the system of equations

$$A^m = \frac{d^m}{dt^m} \left[ \sum_n \sum_{k=0}^{m(\lambda_n)} \mathscr{A}_{nk} t^k e^{\lambda_n t} \right] \Bigg|_{t=0}, \quad 0 \leq m \leq N - 1, \tag{43}$$

in which $N$ is the size of $A$.

However, equation 38 is inhomogeneous. If $A$ is constant, its solution is given by

$$p(t) = e^{At} \left( p_0 + \int e^{-At} b(t) dt \right), \tag{44}$$

which can numerically be integrated using

$$p(t + \delta t) = e^{A\delta t} p(t) + \delta t \frac{b(t + \delta t) + b(t))}{2}. \tag{45}$$

The derivations are given in appendix B.

## 6.1 The optimization approach

Let in general $\dot{p} = A(t)p + b(t)$. The question this time around is how to choose $A$ and $b$ such that $p$ resembles the ABM simulations, or the EBM, in order to generate an easier solvable EBM.

Let to that end $b_i(t) = \beta_i f_i(t, \omega_i)$ and $A_{ij} = \alpha_{ij} - \delta_{ij}$, in which the $\delta_{ij}$ exists to make sure that all the packages are transported away from a node, and potentially retained by the $\alpha_{ij}$. Hence, the $\alpha_{ij}$, $\beta_i$ and $\omega_i$ are in general the parameters over which the model is to be optimized. Let $\not{p}$ be the amount of packages obtained with the ABM, or whichever comparison model, the goal is then to minimize the following objective function:

$$\Sigma_p = \sum_{t=0}^{T} \left(\not{p}(t) - p(t)\right)^2 \tag{46}$$

constrained by

$$\sum_j \alpha_{ij} = 1,\ \forall i\ \alpha_{ij} \geq 0,\ \forall (i,j) \in \mathscr{E},\ \alpha_{ij} = 0,\ \forall (i,j) \notin \mathscr{E}, \tag{47}$$

and, additionally, if we assume $f_i(t, \omega_i) = e^{-t/\omega_i}$, $\omega_i > 0\ \forall i$, which corresponds with a situation similar to the assumptions made in the ABMs. $f_i(t, \omega_i)$ hence starts off at 1, which means that only supply is present. It decreases to 0, which means that the consumption caught up with the provided supply. Of course, it will not reach zero in finite time, but that is no problem as it will get close enough at around $t = 5\omega_i$.

Applying this optimization problem to the custom-made network from figure 5.1a, there are 36 $\alpha_{ij}$ and 10 $\omega_i$, which is already quite a considerable amount of parameters. The supply is assumed to start off at some $\beta_i$, which we take $\frac{1}{2}$ for all $i$. This is in accordance with the choice made in section 5.1, which will be the reference material for this section. Choosing the $\beta_i$ beforehand is done to reduce the amount of necessary parameters by 10. Eventually, one could even assume $A_{ij} = \alpha_{ij} g_{ij}(t, \chi_{ij})$, which generally increases the amount of parameters.

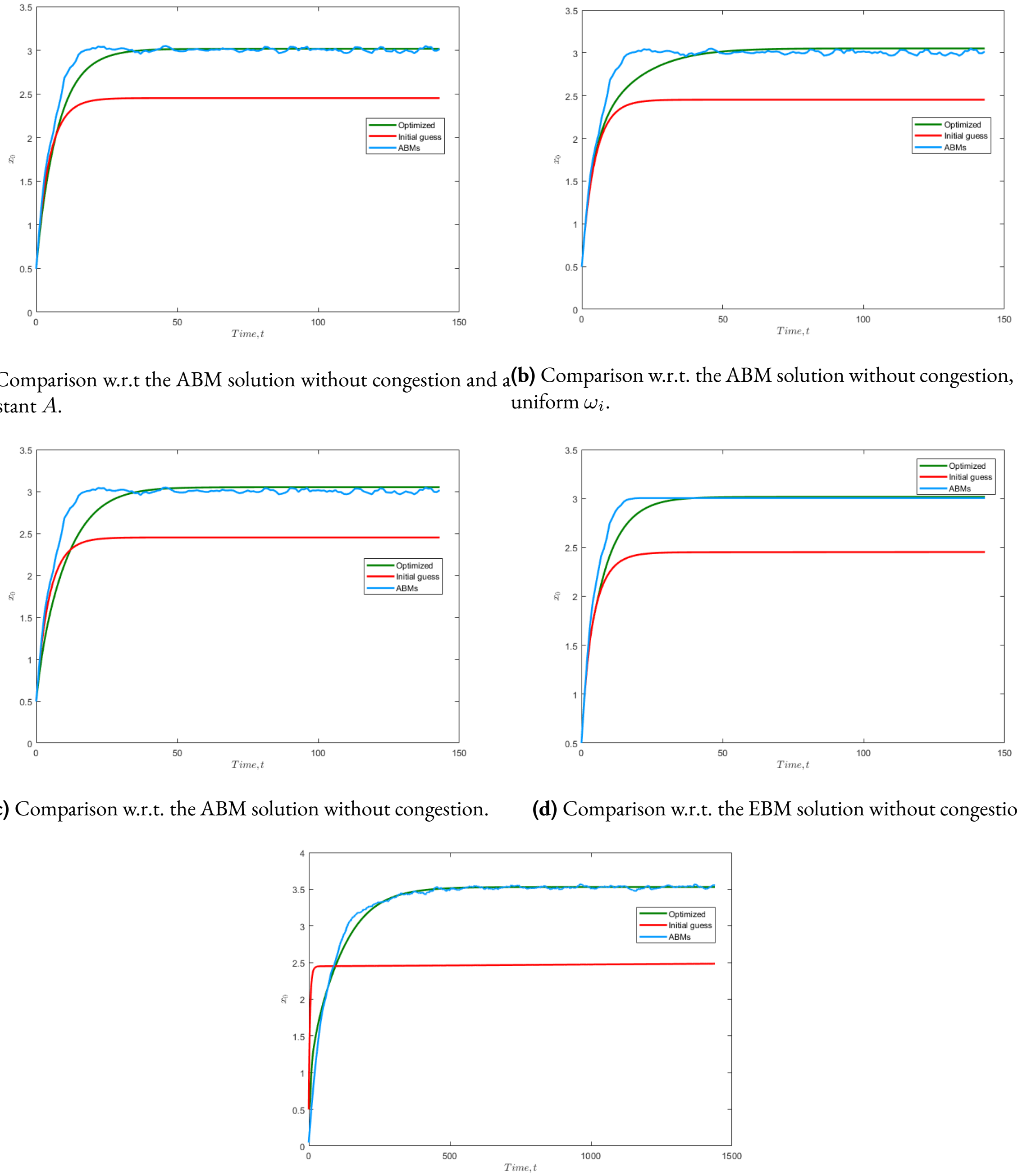
While performing a couple of time optimizations of the implementation, a compromise had to be made. The analytical result from equation 65 could be taken, which is expensive to find due to the re-evaluations of $e^{At}$ on each time step, and in which the integral would internally be numerically integrated by Matlab [21]anyway. The other option is to use equation 66, which iteratively determines the solution on the next time step, based on the solution on the previous time step. This method has the boon of not having to determine matrix exponentials each time step, and is hence much more efficient. Even though the relative numerical accuracy is only $10^{-3}$, this is sufficient as the ABM data is noisy anyway. This method is not as easily usable for time dependent matrices $A(t)$, though, in which case a simple Euler forward implementation works anyway. Higher order numerical methods are not necessary, as Euler forward turns out to be sufficiently accurate and still quick to implement.

Using a constant $A$ gave surprisingly accurate results, an example of which can be seen in figure 6.1a. This being already so accurate, and frankly more accurate than hoped or expected, prompted the motivation to make $A$ time-dependent in hopes of creating an even more accurate result, instead of improving the current result with a constant $A$. The chosen way to build up the matrix $A$ is

$$a_{ij} = (\alpha_{ij} - \delta_{ij}) \left(1 - e^{-\frac{t}{\omega_{ij}}}\right), \quad b_i(t) = s_0 e^{-\frac{t}{\omega_i}}, \tag{48}$$

in which $\delta_{ij}$ is the Kronecker delta. Figure 6.1b shows an optimization attempt in which the $\omega_{ij}$ are all taken equal to the $\omega_i$ used in the $b_i$, to reduce the amount of parameters considerably, and because the amount of consumption should approximately follow the same shape as the amount of leaving packages. Figure 6.1c shows an attempt in which all $\omega_{ij}$ are equal when varying $j$, but they are not equal to the $\omega_i$ used in the $b_i$. This increases the required amount of parameters by 10. One could go further and allow for all $\omega_{ij}$ do be different, but that changes the amount of $\omega_{ij}$ from 10 to 36, and as the results are already pretty good, this has been skipped.

Figure 6.1d shows the same procedure as figure 6.1c, except that the comparison has been performed with respect to the EBM solution instead of the ABM solution. As the start-up process ends abruptly, the exponentials cannot describe that well, but it is possible to choose other functions $g_{ij}(t, \chi_{ij})$ and $f_i(t, \omega_i)$ with more



**(a)** Comparison w.r.t the ABM solution without congestion and a constant $A$.

**(b)** Comparison w.r.t. the ABM solution without congestion, with uniform $\omega_i$.

**(c)** Comparison w.r.t. the ABM solution without congestion.

**(d)** Comparison w.r.t. the EBM solution without congestion.

**(e)** Comparison w.r.t. the ABM solution with congestion.

**Figure 6.1:** The result for $x_0$ of the optimization procedure for the custom-made network, using a constant matrix $A$ (6.1a) and a time-dependent matrix $A$ in the other subfigures, with parameters $\alpha_{ij}$, $\omega_i$ and $\omega_{ij}$. The optimization was performed with respect to the results from section 5.1.

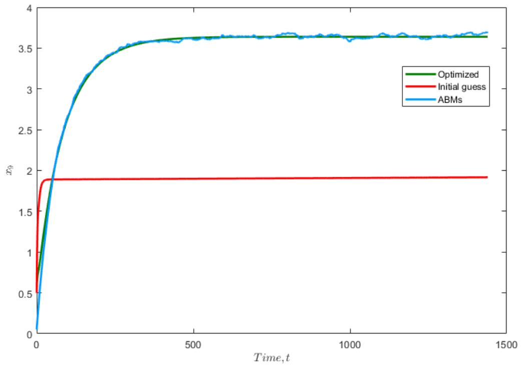


**Figure 6.2:** The result for $x_9$ of the optimization procedure for the custom-made network, using a time-dependent matrix $A$, when comparing to the ABM with congestion.

discrete and jumpy natures. It is notable that the steady state is very well approximated, but it should be noted that, as a long section of the solution is steady state, it is bound to weigh heavier in the optimization process.

Last, but not least, figure 6.1e shows a realization of the optimization approach used on the ABM with congestion. In this attempt, the obtained EBM solution overlaps pretty well with the ABM solution, but in the start-up process and in the steady state. This is due to the start-up being more stretched out by the influence of congestion and making more of a curve than a bend in the solution. Figure 6.2 shows an even better example of this for node 9 in the custom-made network, which pretty much completely overlaps.

One thing that should be clarified is the worsening of the optimized EBM solution between figures 6.1a and 6.1b. This happens because the EBM is optimized for all 10 nodes at once and it just so happens that a worse result for node 0 is favourable for the other nodes on average.

The same has also been done for the random 40 node network, to investigate whether optimization iterations for a larger network would converge much slower and whether they would be as accurate. We remark that the 40 node network leads to an optimization over 280 parameters. The numerical solution of this problem proved to be about as fast as running the ABM simulations, and is certainly faster than generating the EBM with congestion. More derivatives have to be determined due to the amount of parameters, while due to the amount of solutions to fit, the optimum is harder to reach; even moreso when comparing to the ABM results, as opposed to comparing to the EBM results. For multiple reasons, iterations take longer to process, but they still only take approximately a day or so, which is by far better than the EBM, as that took one day to solve with significant error for the 10 node network. The optimization attempts for the custom-made network took about 7 hours for the most intensive one, and about 2 to 3 for the one with a constant matrix $A$. This could be further reduced by reducing the $\omega_{ij}$ back and reducing the amount of time steps taken into account, as well as sacrificing accuracy for a larger time step used in the numerical integration of the EBM solution.

We conclude from this section that finding an EBM with optimization solvers is surprisingly well doable. The start-up process in the congestionless equations is not approximated very well, but for non-secluded nodes, the EBM from chapter 3 does its job well and quickly anyway. The steady states are pretty much always followed pretty closely, as they are taken into account for more time steps in the least squares fit. The real power of this approach is finding an equation that approximates the ABM solution with congestion reasonably well in a lot less time than the previously explored EBM approach.

# Chapter 7: The Generalized model approach

After attempting to optimize the model parameters, it is clear that choosing the shapes of $A(t)$ and $b(t)$ rather became a task of arbitrary guessing, as the way to elicit them is not obvious.

Let us thence take a look at a different way of modelling altogether; Generalized Models (GMs) is an evident way to remedy this flaw. Let us consider, for continuity sake, the custom-made network from figure 5.1a. This network has always been thought of as a network of cities containing distribution centres. As an extension of this, we might consider customers, or rather postal zones, ordering products from some (online) store, in which turn a delivery service delivers those orders, scheduled by the store. Rather than the previously studied physical package transport, a more abstracted supply network containing the entire package delivery network is considered. This time, the interactions between nodes will be regarded as management policies and social influences. These then correspond to decision rules in ABMs. However, contrary to ABMs, they are not explicitly defined.

Let $P$ be the vector of parts, i.e. orders, which are assumed to exist on edges in the network, tailored such that $P_i > 0$ for all $i$. From customer to store, this means a placed order, while from store to delivery centre, this means a scheduled order. From delivery centre to customer, this denotes an actual order being delivered. Additionally, two extra parts can be defined as entering and leaving the network; ”supplied” and ”consumed” orders, or rather the social influence which leads to orders being posted and the consumption of said orders. These are represented by the customer being connected to a reservoir, and the interactions taking place between the customer and the reservoir. Moreover, let $F$ be the vector of flows, or productions, in the network, where, again, $F_i > 0$ for all $i$. Contrary to intuition, these take place at the nodes and at the reservoir; at the source they might denote some kind of social influence, while at the customer this translates to an actual order placement. At the store, this denotes the act of scheduling an order, and at the delivery centre this denotes that an order is loaded into a truck. Furthermore, let $P^*$ and $F^*$ denote a steady state of $P$ and $F$, assuming at least one exists.

Let the customers be denoted by $C_i$, let the stores be denoted by $S_i$ and let the delivery centre nodes be denoted by $D_i$. Let lastly the external reservoir be denoted by $R$; different reservoirs are not labelled differently.

## 7.1 One store and two customers

Let us here look at a fairly simple, non-trivial example in which two customer-delivery centre pairs are present, and only one store. In general, though, one delivery centre might be connected to two customer zones, and multiple stores might deliver through the same delivery service. The focus here is on how to develop the equations, though, such that an extension including said behaviours would be clear.

Several expansions and variations of the same problem are covered in the current and following sections. An extensive bifurcation analysis is conducted in each case, followed by an comprehensive comparison with the statistical ensemble method. Additionally, for each section after the current one, there is a table in appendix D, representing the part sensitivities and influences. The most important results are recapped at the end of the chapter.

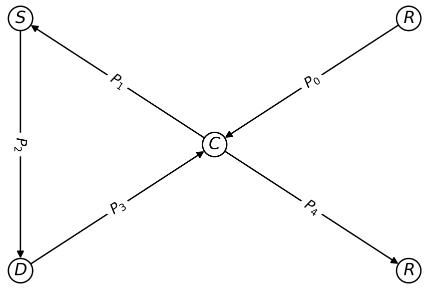


**Figure 7.1:** A simple example graph to illustrate the structure of the network. The single motive has been given with the respective parts on the edges, along with the reservoir they, if applicable, flow into or out from. This network consists of a cyclic triangle motive, with a customer, a store, a delivery company, and an external reservoir.

Figure 7.1 shows the initial toy graph, considered in this section. In this case there are five $P$'s. The equations read:

$$\begin{cases} \dot{P}_0 = F_0(P_0) - F_1(P_0, P_1, P_3) \\ \dot{P}_1 = F_1(P_0, P_1, P_3) - F_2(P_1, P_2) \\ \dot{P}_2 = F_2(P_1, P_2) - F_3(P_2, P_3) \\ \dot{P}_3 = F_3(P_2, P_3) - F_4(P_3, P_4) \\ \dot{P}_4 = F_4(P_3, P_4) - F_5(P_4). \end{cases} \tag{49}$$

In this equation, each $F$ corresponds to some production or transformation of parts in the network. For instance, $F_0$ describes the creation of parts and $F_5$ describes the consumption of parts. Likewise, $F_1$, $F_2$ and $F_3$ describe the transformation of parts into other parts. In general, the transformations of $P_1$ into $P_2$, could depend on the strain of the remainder of the network on the store and the transformation of $P_2$ into $P_3$ could depend on the strain on the delivery centres. As we only consider one motive here however, this is not relevant yet.

Moreover, the external production of $P_0$ and the consumption of $P_4$ could in general depend on more than just themselves. However, these effects are not included here.

These equations need to be scaled with respect to their steady state, such that $p = \frac{P}{P^*}$ and $f = \frac{F}{F^*}$ and for instance:

$$\dot{p}_0 = \frac{F_0^*}{P_0^*} f_0(p_0) - \frac{F_1^*}{P_0^*} f_1(p_0, p_1, p_3). \tag{50}$$

Doing this for all $p$, the Jacobian can be created as follows:

$$J = \Lambda \Theta_f^p, \quad \Lambda_{ij} = \frac{F_j^*}{P_i^*} N_{ij}, \tag{51}$$

in which $N$ is a matrix indicating which flows take part in which equation. $\Theta_f^p$ is constructed by simply taking the derivatives of each flow term, with respect to each $p$.

$$N = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}, \quad \Theta_f^p = \begin{pmatrix} f_0^{p0} & 0 & 0 & 0 & 0 \\ f_1^{p0} & f_1^{p1} & 0 & f_1^{p3} & 0 \\ 0 & f_2^{p1} & f_2^{p2} & 0 & 0 \\ 0 & 0 & f_3^{p2} & f_3^{p3} & 0 \\ 0 & 0 & 0 & f_4^{p3} & f_4^{p4} \\ 0 & 0 & 0 & 0 & f_5^{p4} \end{pmatrix} \qquad (52)$$

These $f_i^{p_j}$ are the mentioned elasticities, that parametrize the generalized model. They are meant to denote the derivative of the scaled $f_i$ with respect to scaled part $p_j$ at the steady state. These don't necessarily need to be known for the entire range of the $f_i$, because only the stability of $p = 1$ is of interest here.
Moreover, as in steady state the $\dot{P_i}$ are necessarily zero, it must hold that $F_i^* = F^*,\ \forall i$, for some constant $F^*$, which simplifies the equations further.

Next, the different elasticities need to be categorized, in order to sufficiently simplify the model for analysis. Let us therefore, like in [1, 2], distinguish three different elasticities; elasticity to supply, elasticity to inventory level and elasticity to co-production, denoted by $f_S$, $f_I$ and $f_C$ respectively.

As there are no multiple parts produced on one node, because only one structure is considered here, there exists no elasticity to co-production. If, however, multiple cyclical structures were to intersect at some point, either in the $S$ node and/or some $D$ nodes, this elasticity to co-production could be included.

Most elasticities in $\Theta_f^p$ can therefore be categorized into either $f_S$ or $f_I$. The elasticities to inventory level are $f_0^{p0}$, $f_1^{p1}$, $f_2^{p2}$, $f_3^{p3}$ and $f_4^{p4}$. The elasticity to supply parameters are $f_1^{p0}$, $f_2^{p1}$, $f_3^{p2}$, $f_4^{p3}$ and $f_5^{p4}$.

Lastly, $f_1^{p3}$ remains. This one can not easily be captured by the above-mentioned elasticity descriptions. Rather, it consists of the degree of saturation or novelty of the products. That is, if a new product enters the market, a lot of people might want to order it, as it is novel. Later on, when everyone possesses said product, less orders will come in. On the other hand, someone might not order food multiple consecutive days; such a scenario could also be described by this saturation. The so called elasticity to saturation is hence $f_1^{p3}$, which is denoted by $f_\sigma$. This should not be confused with $f_1^{p1}$, which means that less orders will be placed, when orders are recently placed, although they are similar. For this initial analysis, these conclude all the considered parameters and interactions in this section.

With these three elasticities, the Jacobian can be evaluated, along with its bifurcation points. Let us for simplicity assume that all $f_S$ and $f_I$ are identical, along with all $P^*$. Physically, the elasticities should satisfy $f_I \leq 0$, $f_S \geq 0$ and $f_\sigma$ can bear any sign. More specifically, they are generally varied in the ranges $0 \leq f_S \leq 1$, $-1 \leq f_I \leq 0$, and $0 \leq f_\sigma \leq 2$ in this thesis. $f_I$ should be negative, corresponding to less orders being placed if a lot of orders were already placed and less orders being able to be delivered, if a lot of delivery trucks are already on their way. $f_S$ should be positive, which means that social influence leads to orders being placed, more orders being delivered if more orders are handled by the store, and more products being consumed if more products are delivered. A positive $f_\sigma$ indicates that receiving a product increases its demand. This is for instance the case when customers exhibit compulsive behaviour, or are addicted to a product. If a customer node is interpreted as a postal area, a positive $f_\sigma$ could be considered as a new product entering the market, which is picked up by more and more people. Usually, though, $f_\sigma$ is more likely to be negative, which means that customers become saturated and demand decreases as customers receive their products. This corresponds with phenomena like novelties already being purchased by a majority of the public. As everyone starts to possess this novelty, its demand decreases. The Jacobian then explicitly reads

$$J = \frac{F^*}{P^*}\begin{pmatrix} f_0^{p_0} - f_1^{p_0} & -f_1^{p_1} & 0 & -f_1^{p_3} & 0 \\ f_1^{p_0} & f_1^{p_1} - f_2^{p_1} & -f_2^{p_2} & f_1^{p_3} & 0 \\ 0 & f_2^{p_1} & f_2^{p_2} - f_3^{p_2} & -f_3^{p_3} & 0 \\ 0 & 0 & f_3^{p_2} & f_3^{p_3} - f_4^{p_3} & -f_4^{p_4} \\ 0 & 0 & 0 & f_4^{p_3} & f_4^{p_4} - f_5^{p_4} \end{pmatrix}, \tag{53}$$

and its determinant reads

$$\Delta_J = \left(\frac{F^*}{P^*}\right)^5 (f_I - f_S)\left(f_I^4 + f_I^2 f_S^2 + f_I f_S^2 f_\sigma + f_S^4\right), \tag{54}$$

whence it is clear that varying $P^*$ has no effect on the sign of the determinant. Likewise, as $P^*$ and $F^*$ appear as factors in front of $J$, they will neither affect the stability. Moreover, when this determinant equals zero, $J$ has a zero eigenvalue, and a saddle-node bifurcation occurs. Hopf bifurcations occur when the resultant of $J$ is zero, while the Hopf number, $\chi$, is negative [2, 3], as explained in section 1.2.1.

The determinant evaluates to zero when $f_I^4 + f_I^2 f_S^2 + f_I f_S^2 f_\sigma + f_S^4 = 0$, or when, trivially, $f_I = f_S$, which means that both are trivially $0$, as $f_I \leq 0$ and $f_S \geq 0$. In the first case, the first, second and last term are necessarily positive, such that all negativity should come from the third term, i.e. an additional saddle-node type bifurcation exists when $f_\sigma = f_\sigma^*$, if

$$f_\sigma^* = -\frac{f_I^4 + f_I^2 f_S^2 + f_S^4}{f_I f_S^2}. \tag{55}$$

For $f_\sigma^* > 0$ equation 55 defines a cone-like surface, visible in Figure 7.2, which is denoted by $SN$ here.

Moreover, as discussed in [3], using the Resultant method, additional bifurcations can be found at the surfaces where the for completeness explicitly given Resultant,

$$R_5 = -\frac{1024F^{10}}{P^{10}} \cdot \left(f_S^4 - \left(3f_I + \frac{f_\sigma}{16}\right) f_S^3 + \left(\frac{67f_I^2}{16} + \frac{f_I f_\sigma}{8}\right) f_S^2 - 3f_S f_I^3 + f_I^4\right)^2 \cdot$$
$$\left(f_S^6 - \left(4f_I + \frac{f_\sigma}{4}\right) f_S^5 + \left(\frac{29f_I^2}{4} + \frac{f_I f_\sigma}{2} - \frac{f_\sigma^2}{64}\right) f_S^4 - \left(\frac{17f_I^3}{2} + \frac{f_I^2 f_\sigma}{4}\right) f_S^3 + \frac{29f_S^2 f_I^4}{4} - 4f_S f_I^5 + f_I^6\right)^2, \tag{56}$$

vanishes. Clearly, again, $P$ has no influence on the bifurcation locations.

Each of these factors can become zero, which only noticeably happens for relatively large $f_\sigma$ compared to $f_I$ and $f_S$. It is also more challenging to deal with them, as a sixth degree equation can not be solved analytically. The large factor (on the second row of equation 56), contains two branches, one for positive $f_\sigma$ and one for negative $f_\sigma$, shown in figure 7.2, coloured blue and red respectively. However, at the intersection of the former with the $SN$ surface, this branch changes from a Hopf bifurcation, denoted $H1$, to a real Hopf situation, denoted $RH1$. On the $RH1$ part, the symmetric eigenvalues are coincidentally also the eigenvalues closest to $0$, and hence constitute saddle index $1$. The latter always has two purely imaginary complex conjugate eigenvalues, corresponding with a Hopf bifurcation, denoted by $H2$.

The last factor of the Resultant, has two symmetric real eigenvalues, of which none have the smallest real part. This surface also intersects with the cone, shown in yellow in figure 7.2. As there are two symmetric real eigenvalues, it is a real Hopf situation, denoted by $RH2$.

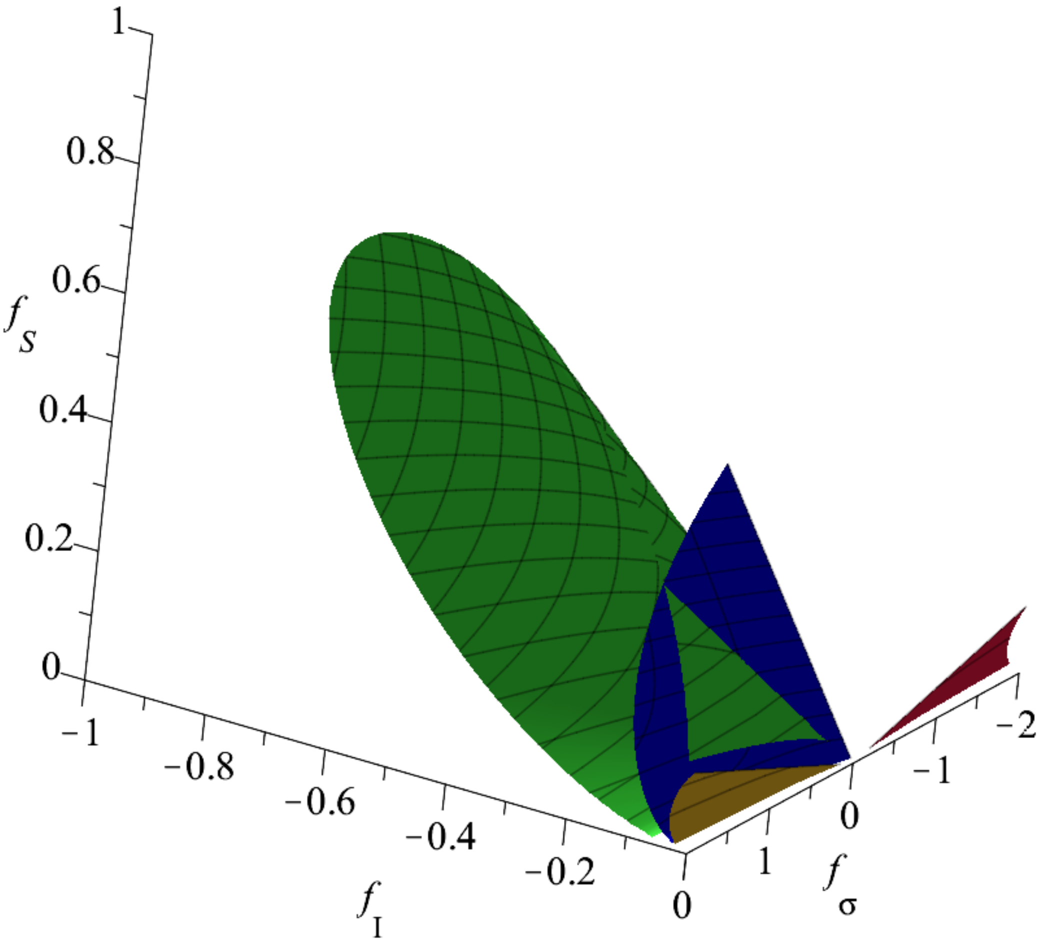


**Figure 7.2:** A 3D visualisation of the bifurcation surfacess. The green cone-like surface constitutes the $SN$ bifurcation. It appears to stop above the yellow surface due to the amount of points used, but it should persist connecting to itself at $f_S = f_I = 0$. The blue surface constitutes both $H1$ and $RH1$, and the yellow surface constitutes $RH2$. Lastly, the red surface constitutes $H2$.

$RH1$ and $RH2$ do not have an effect on he stability of the network, as they are not really bifurcations. However, on their surface, there exists a homoclinic orbit, as explained in [17].

Lastly, when $SN$ and $H1$ intersect, a Bogdanov-Takens, or $BT$ for short, bifurcation occurs, as two bifurcation surfaces meet there and briefly, two zero eigenvalues exist. This also happens at $f_S = f_i = f_\sigma = 0$, which corresponds with the creation of interactions on the network.

Figure 7.2 shows the bifurcation diagram for the current investigated network, whereas figure 7.3 shows two-dimensional projections of this bifurcation diagram. Evaluating the eigenvalues (not shown) in different regions of the parameter space reveals their stability. Following figure 7.2, the inside of the cone-like surface is unstable, and the area between the blue surface and the $f_I = 0$ plane is also unstable. Likewise, the region between the red Hopf bifurcation surface for negative $f_\sigma$, and the $f_I = 0$ plane is also unstable. In the remainder of te parameter space, the steady state $p = 1$ is stable. This is illustrated in figure 7.4.

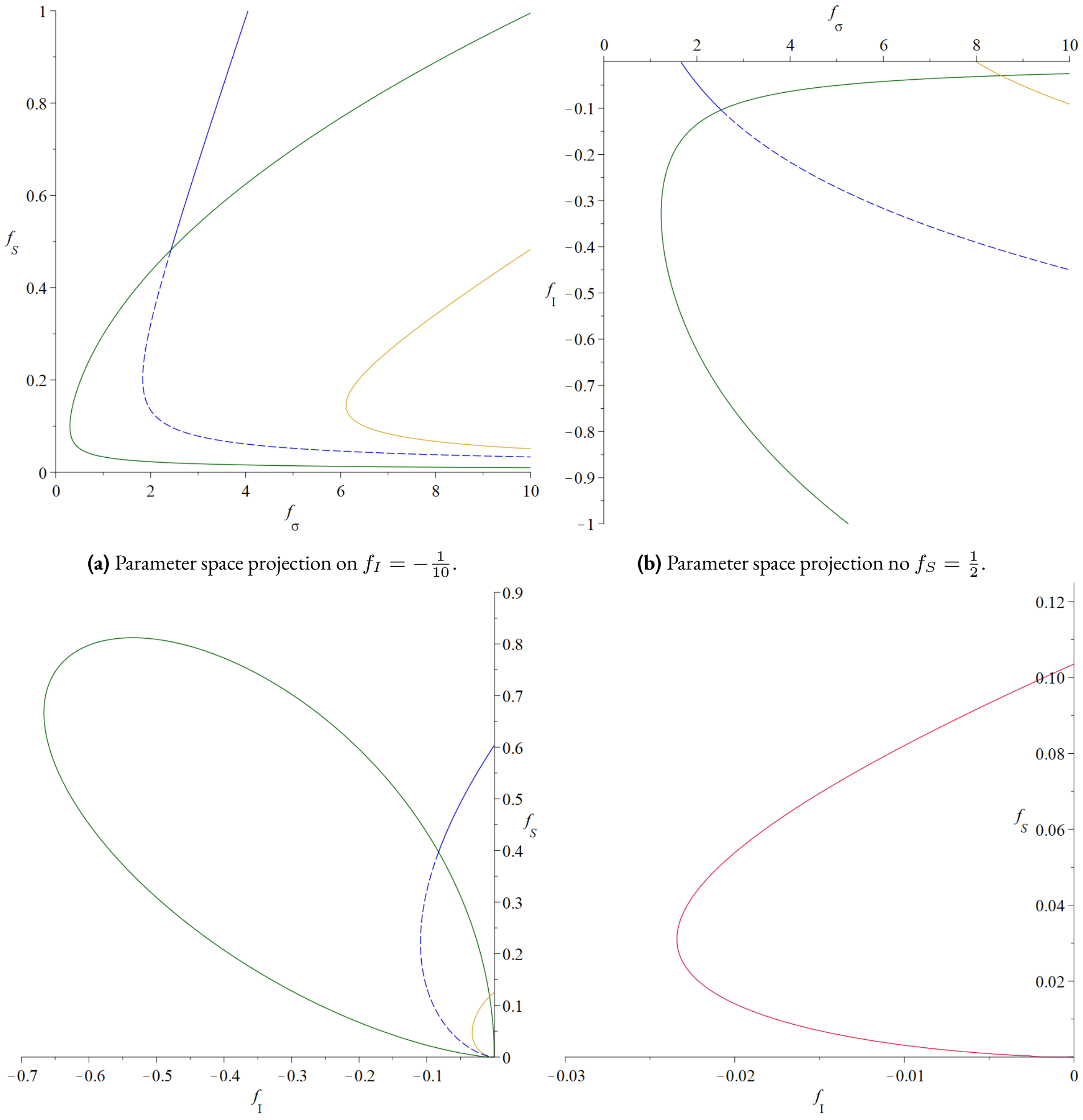
A similar analysis can be performed using the statistical ensemble method, henceforth referred to as SE method. In this method, the parameter space is sampled uniformly and on all these samples, the stability of the system is determined, by calculating all eigenvalues of the system, and determining whether the leading eigenvalue is positive. Such an analysis gives a representation similar to figures 7.2 and 7.3. However, rather than explicitly showing the stability regions, the stability of the system is demonstrated by a percentage of stable networks, which is commonly referred to as $PSW$ (Proportion of Stable Webs) in the literature [1]. The $PSW$ is plotted as a function of $f_I$, $f_S$ and $f_\sigma$ in figure 7.5.

In this figure, $10^8$ samples were taken in both ensembles. In the ensemble of identical paremeters, all $f_I$ are taken equal, as are all $f_S$, just like in the bifurcation analysis. In the ensemble of individual parameters, all $f_I$ and $f_S$ are sampled independently. The figure is then made by dividing the investigated parameter into intervals, and considering the proportion of stable networks in said intervals. When sampling independently, the intervals are defined for the average of the different $f_I$ and $f_S$. A keen-eyed reader can notice that the individual parameter ensemble plots do not completely reach the sides and squiggle a little. This is due to there existing five different $f_I$ and $f_S$, and hence their average has a very low chance of reaching the boundaries. This also explains the jagged edges, as only a few samples fit in those intervals.

To emphasize the difficulty of obtaining precise information using this method, and the importance of keeping the parameter ranges in mind, results taking $0 \leq f_\sigma \leq 1$, instead of the usual $0 \leq f_\sigma \leq 2$, have been added in figure 7.6.

For instance, as a smaller range of $f_\sigma$ is considered there, the projection of the cone in figure 7.2, as given in figure 7.3c, shrinks. This effect is reflected in the larger $PSW$ for all $f_\sigma$, but is also reflected in a $PSW$ that is $1$ over a larger portion of $f_I$ and a larger portion of $f_S$. This leaves the impression that $f_I$ and $f_S$ more vehemently impact the stability of the network, whereas this effect entirely comes forth from considering a smaller range for $f_\sigma$, and the proper explanation is that an increasing $f_\sigma$ brings more instability.

As it turns out, the ensemble with individual parameters yields networks that are unstable in a larger part of the parameter space, with exception of low $f_S$ (approximately $< 0.2$), for which the ensemble with identical parameters is less stable. This can be understood in the following way. Of course, the bifurcation diagram differs when considering each elasticity individually. However, their bifurcation surfaces should have some resemblance to figure 7.2, albeit being higher dimensional. Stability is lost by having just one positive eigenvalue. When multiple elasticities are varied, the odds rise of at least one of them being sampled such that the system is unstable. This is a drawback of the SE method; multiple elasticities may cause instability, but this cannot be distinguished well in a curve averaged over all similar elasticities.
Therefore, an average over all similar elasticities does still not illustrate particularly well which parts impact sta-



**(a)** Parameter space projection on $f_I = -\frac{1}{10}$.

**(b)** Parameter space projection no $f_S = \frac{1}{2}$.

**(c)** Parameter space projection on $f_\sigma = 2$.

**(d)** Parameter space projection on $f_\sigma = -2$.

**Figure 7.3:** Four projections of figure 7.2. $SN$ is depicted in green, whereas $H1$ and $H2$ are given in blue and red respectively. Lastly, $RH1$ and $RH2$ are respectively depicted in dashed blue and yellow. Keep in mind that the $f_\sigma = -2$ figure has entirely different axes compared to the $f_\sigma = 2$ figure.

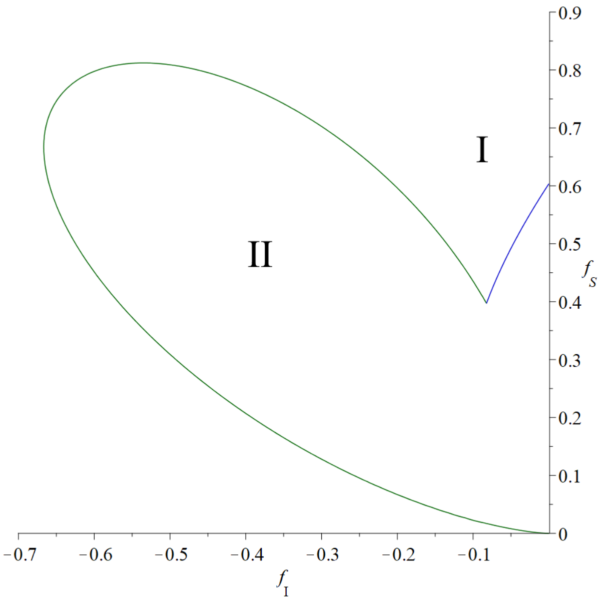
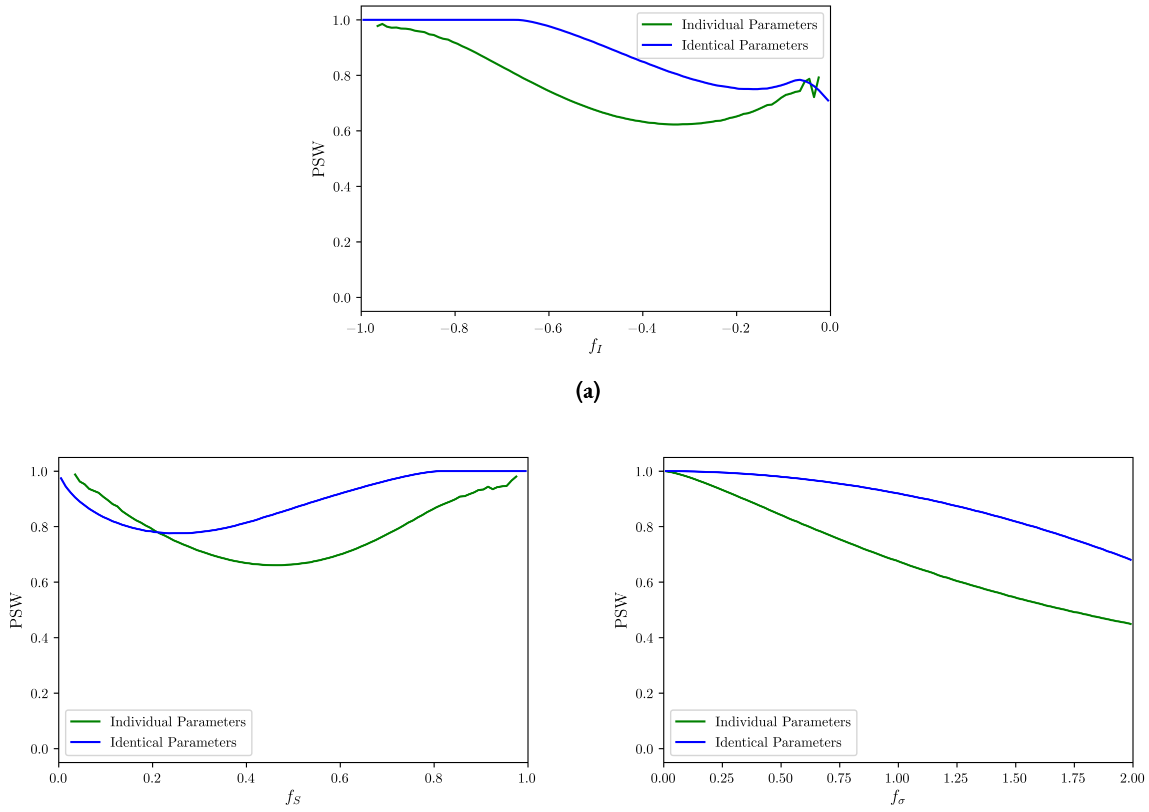


**Figure 7.4:** The stability regions of figure 7.3c. While eigenvalues become zero on the remainder of the blue curve, they do not affect stability, and hence are not relevant for stability purposes. Region I is stable, whereas region II is unstable. The blue blue curve on the right is $H1$ and the green curve on the left is part of $SN$.

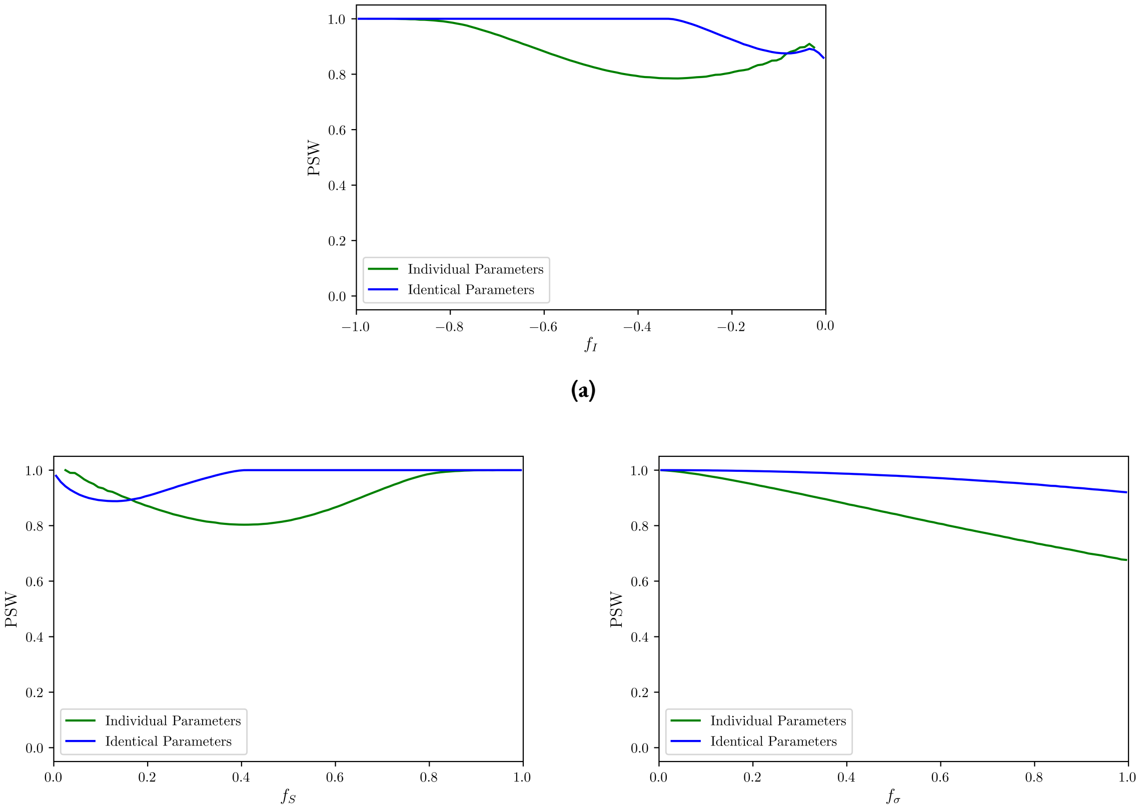


**(a)**

**(b)**

**(c)**

**Figure 7.5:** Results of the SE method performed on equation49, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green. In these ensembles, $10^8$ samples were taken, which were divided over 100 intervals.

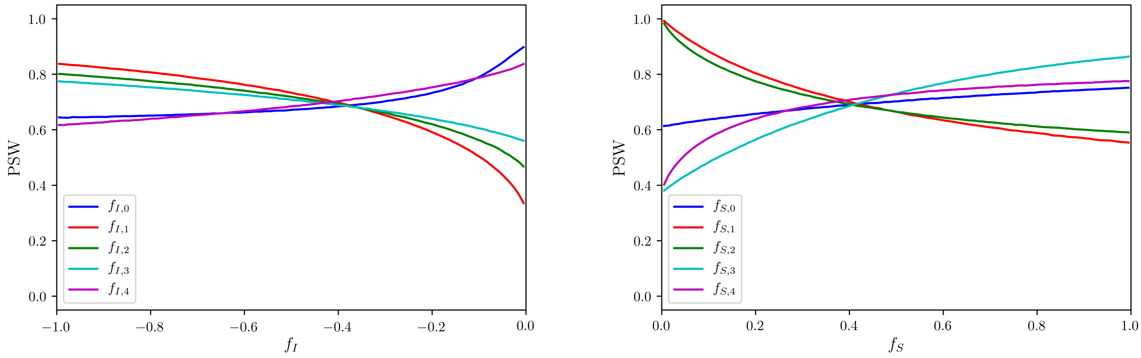


**(a)**

**(b)**

**(c)**

**Figure 7.6:** Results of the SE method performed on equation49, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green. Again, $10^8$ samples were taken, which were divided over 100 intervals. This figure is the same as figure 7.5, with the exception of $0 \leq f_\sigma \leq 1$, instead of $0 \leq f_\sigma \leq 2$.



**(a)** **(b)**

**Figure 7.7:** Results of the SE method performed on equation49, with an ensemble of individual parameters, in which the $PSW$s have been determined for each individual elasticity. $10^8$ samples were taken again, which were divided over $100$ intervals. The second subscript denotes the considered part. For instance, $f_{S,0} = f_1^{po}$. The $f_\sigma$ figure has not been added, as it is the same as the green curve in figure 7.5c, which has been made for the individual parameter ensemble, of course.

bility the most. On top of the SE with averaged intervals, it is thence illustrative to also look at an ensemble in which the $PSW$ is visualized for distinct elasticities. Its results are given in figure 7.7.

This figure contains three features that immediately stand out. Firstly, elasticities of the same kind are not necessarily monotonous in the same direction. Moreover, the elasticities that behave differently, do not pertain to the same nodes for $f_I$ and $f_S$. Lastly, they seem to cross each other at approximately the same point for each elasticity type, except the ones node 4; at about $f_I = -0.4$ and $f_S = 0.4$.

The results in figure 7.7 are easily interpreted, but not always intuitively understood. We restrict the discussion to figure 7.7b, as explaining the results for figure 7.7a is very similar. For instance, an example of taking $f_{S,4} = 0$ is when one orders food regularly, but somehow needs to satisfy a certain constant trash output, such that they have to keep ordering food to that end. Obviously, if the orders do not balance the required trash quota, the network drains or parts accumulate somewhere. A similar argument can be held for $f_{S,3}$ and $f_{S,0}$; taking $f_{S,3} = 0$ corresponds with a consumption that does not depend on the incoming deliveries and $f_{S,0} = 0$ corresponds with a constant social influence making customers want to order at a constant rate. When actually considering the stock of these parts, the system stability increases.
For parts $1$ and $2$, though, taking their $f_S = 0$ represents a network in which placed orders do not affect placed scheduled and placed schedules do not affect delivering packages respectively. They are not realistic as the amount of placed orders, for instance, is in reality not the same every day. Nonintuitively, the lack of these interactions makes the network always stable, as it is much simpler, but less efficient in real life. Hence, such models rely on only the other parameters to correct perturbations and trade off efficiency for simplicity. It should therefore be clear that the results for the individual ensemble, presented in this way, do not always yield physically relevant optima, and are also not always intuitively understood.

This result does immediately illustrates the advantage of the SE method pretty well, though; a bifurcation analysis can easily be employed for systems with $3$ varying parameters, and can still be used with $5$ or so varying parameters. Introducing more parameters in a generalized model really hampers the possibility to make useful statements about the entire parameter space, when using bifurcation analysis, without only considering projections, which may take away a lot of generality. With the SE method, quite a lot of parameters can be introduced and varied. Generally, most parameters are varied pretty well and represent quite a sizeable part of the parameter space, depending on the amount of taken samples. It should however be noted that for networks of increasing

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|---|---|---|---|---|---|
| $Se_i$ | 4.164 | 5.204 | 4.873 | 4.461 | 3.771 |
| $In_i$ | 3.828 | 4.511 | 4.940 | 5.260 | 4.270 |

| | $C$ | $S$ | $D$ |
|---|---|---|---|
| $Se$ | 17.600 | 10.077 | 9.334 |
| $In$ | 17.869 | 9.451 | 10.200 |

**(a)**

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|---|---|---|---|---|---|
| $Se_i$ | 3.160 | 4.846 | 4.804 | 4.331 | 3.250 |
| $In_i$ | 3.710 | 4.554 | 4.934 | 5.183 | 4.454 |

| | $C$ | $S$ | $D$ |
|---|---|---|---|
| $Se$ | 15.587 | 9.650 | 9.135 |
| $In$ | 17.901 | 9.488 | 10.117 |

**(b)**

**Table 7.1:** The Sensitivities and Influences of each part and each node in the one motive network from figure 7.1, when sampling identically (top) and individually (bottom). These have been determined with $10^6$ samples.

size, varying a lot of parameters hampers efficiency a lot, while making the results less intelligible as a lot of information has to be dealt with. If averaging over them to generate the figures, a lot of information is lost, but it can be tricky to sensibly visualize the results if the parameters are not averaged. Such a visualization, in which the parameters are not averaged, is as far as I know not ordinarily made in statistical ensembles, while it adds a tremendous amount of information. However, statistical ensemble methods are often employed in cases with a very large amount of parameters, in which extracting information becomes increasingly harder to do manually. Moreover, as the $PSW$ is an average over all other parameters, this does not paint the clearest picture of the situation. For example, averaging over elasticities, a potential set for all $f_I$ having $f_{I,0} = 0$ would yield an average over $f_I$ of $-0.4$. This intuitively corresponds to a $PSW$ of approximately $0.8$. This does however not generally hold and fixing a specific $f_I$ may alter the $PSW$ distribution significantly. Analogously, if the $f_I$ parameter set is taken to be $\{-0.5, -0.5, -0.5, -0.5, -0.5\}$, which clearly averages out to $-0.4$, the $PSW$, averaged over the remaining $f_S$ and $f_\sigma$ is not necessarily about $0.8$. It is therefore hard to gain precise information with the SE method, but overall stabilities are for sure more easily compared in the SE method, compared to the bifurcation analysis method.

On top of the $PSW$s, the sensitivity and influence of all parts has also been determined for all network parts, as defined by equations 4. By extension, the sensitivity and influence of each node have also been determined, as the sum of the sensitivities and influences of the connected parts. These have been given in table 7.1. Even when only considering barely unstable parameter sets, some seep through where the ln has a negative argument. These have been omitted, but the found values should still remain representative. For this one node network, the amount of omitted parameter sets amounts to up to $1\%$, depending on the considered sensitivity.

It turns out that $10^6$ already gives a one decimal accurate result. This is sufficiently accurate, when the goal is to know the relative importance of all nodes. This is relevant, as $10^8$ samples for the $PSW$ figures eliminates a lot of variance, but determining the sensitivities and influences costs a lot more processing power.

From table 7.1 it is clear that the sensitivities are generally quite close to each other. Moreover, when sampling identically, it looks like the sensitivities earlier in the network correspond with influences later in the network. That is, $Se_0 \approx In_4$ and so on. One has to keep in mind, though, that they have been scaled logarithmically. It is also worth mentioning how sensitivities and influences do not have a meaning on their own, but only relative to others. Hence, $P_3$ has the highest impact on the network, as it controls the feedback to $P_1$. Furthermore, $P_0$ has a lower impact on the network than the other parts. This can be understood in the following way; both $P_0$ and $P_4$ are only connected to one node, and hence are expected to have a lower impact on the network. However, $P_0$ flows at the beginning of the network and establishes and affects $P_1$. On the contrary, $P_4$ impacts $P_3$ backwards, which affects the feedback of $P_3$ to $P_1$. As $P_3$ impacts the network a lot, $P_4$ implicitly does so as well. As a result, $P_0$ influences the network a bit less strongly. $P_1$ is the most sensitive part, as it is directly affected by the most other parts.

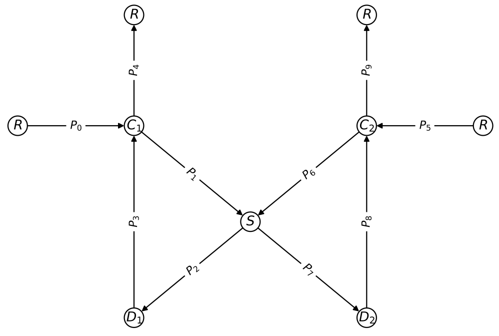


**Figure 7.8:** A network consisting of two merged simple motives, shown in figure 7.1, connected at the $S$ node. Again, the parts have been added.

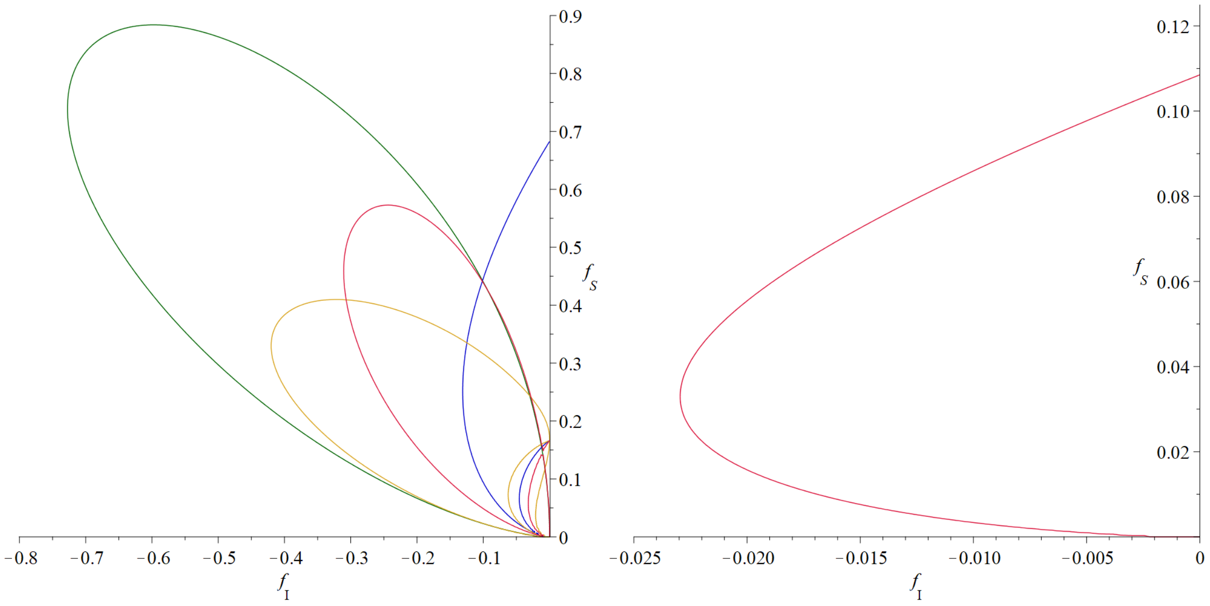
Additionally, it is also clear that the nodes with the highest degree are the most influential and sensitive in the network. It is however harder to justify the sensitivities and influences when sampling independently. Similar results hold as with an identical ensemble, but the influences and sensitivities are mostly weaker. This is most strongly visible for $P_0$ and $P_4$, which now have similar sensitivities, but a larger difference in influence. One can hence conclude that results for identical ensembles translate well to independent ensembles, but still have to be taken with a grain of salt.

As $10^8$ samples in the single motive network took about 5 hours to process, a boundary for practical use lies at about $10^9$ samples, computing parallel, which could easily be implemented, over 8 cores. This could then be run overnight. For larger networks, which take more time due to a larger Jacobian matrix, the ensembles might take significantly longer to run, as eigenvalue finding algorithms have a complexity of $\mathcal{O}(N^3)$. Comparing this to solving the equations numerically for $T$ time steps, gives a complexity of $\mathcal{O}(TN^2)$, which is generally larger, and can only easily be done for a chosen set of parameters.

This analysis illustrates the effect of different elasticities on the stability of the network. increasing $f_S$ and decreasing $f_I$ has a stabilizing effect, whereas increasing positive $f_\sigma$ and decreasing negative $f_\sigma$ has a destabilizing effect. Physically, these can be explained as follows; an increasing $f_\sigma$ indicates, as stated before, for instance a novelty or addiction. Orders receiving at the customer triggering an increase in placed orders is a positive feedback loop, which is prone to spiralling out of control. On the contrary, an excessively low $f_\sigma$ indicates that the customers are quickly saturated and hence no orders will be placed, which drains the system of its parts and disrupts the system further. Moreover, increasing $f_S$ indicates that supply of parts more prominently causes the parts to transform. This reduces build-up of parts (or orders) and corresponds to a faster handling of orders in a real life equivalent. Lastly, decreasing $f_I$ indicates that outflow of parts negatively impacts further production of these parts. This corresponds with a shortage of products or delivery vehicles. Knowing the stability properties of a single motive sets a precedent for more elaborated networks.

### 7.1.1 Two Motives

Let us move on to a network where two motives are merged at their common $S$ node, which corresponds to multiple customers or postal zones placing orders. This network is shown in figure 7.8.



**(a)** Parameter space projection on $f_\sigma = 2$.

**(b)** Parameter space projection on $f_\sigma = -2$.

**Figure 7.9:** Two-dimensional projections on $f_\sigma = 2$ and $f_\sigma = -2$ of the bifurcation surfaces in the network containing two motives joined at node $S$. Between the red Hopf curve and the $f_S$ axis, the system is unstable, whereas on the other side of the curve, the system is stable.

When joining both of these at the $S$ node, the system of equations become

$$\begin{cases} \dot{P}_0 = F_0(P_0) - F_1(P_0, P_1, P_3) \\ \dot{P}_1 = F_1(P_0, P_1, P_3) - F_2(P_1, P_2, P_6, P_7) \\ \dot{P}_2 = F_2(P_1, P_2, P_6, P_7) - F_3(P_2, P_3) \\ \dot{P}_3 = F_3(P_2, P_3) - F_4(P_3, P_4) \\ \dot{P}_4 = F_4(P_3, P_4) - F_5(P_4) \\ \dot{P}_5 = F_6(P_5) - F_7(P_5, P_6, P_8) \\ \dot{P}_6 = F_7(P_5, P_6, P_8) - F_8(P_1, P_2, P_6, P_7) \\ \dot{P}_7 = F_8(P_1, P_2, P_6, P_7) - F_9(P_7, P_8) \\ \dot{P}_8 = F_9(P_7, P_8) - F_{10}(P_8, P_9) \\ \dot{P}_9 = F_{10}(P_8, P_9) - F_{11}(P_9) \end{cases}, \tag{57}$$

in which the same elasticities will be taken as before. The Jacobian is not given here, as it is large and does not illustrate the problem much further, but it is constructed in a similar way as in equation 53. A three-dimensional bifurcation diagram becomes quite crowded when merging two motives. As a result of this, it is perhaps easier to keep the shapes of the previous bifurcation diagram in mind, and only consider a two-dimensional projection of such a diagram now. Such a projection is displayed in figure 7.9.

These projections, when compared to figure 7.3c, have some resemblance, but also contain some noticeably different striking features. For $f_\sigma = 2$, the same surfaces return, albeit in a slightly different size. The saddle-node cone-like surface is present, as are the Hopf surface and Real Hopf surfaces. In addition, there are two new cones, with additional small branches close to the $f_S$ axis. These all turn out to be Real Hopf situations as well. The figure containing real Hopf surfaces becomes rather crowded and hence unclear. Henceforth, as the situations will only become more complicated, only the stability surfaces are given, to allow for a more compre-

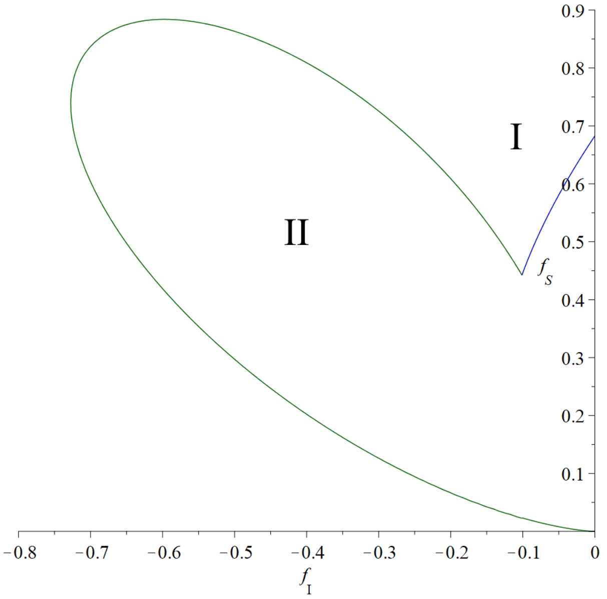


**Figure 7.10:** The stability regions present at $f_\sigma = 2$ for equation57. This figure is generated analogously to figure 7.4, with the minor difference that both the Hopf branch and the cone-like surface are a bit larger. Region I is stable, whereas region II is unstable. The blue curve on the right is again a Hopf bifurcation and the green curve is part of a saddle-node type bifurcation.

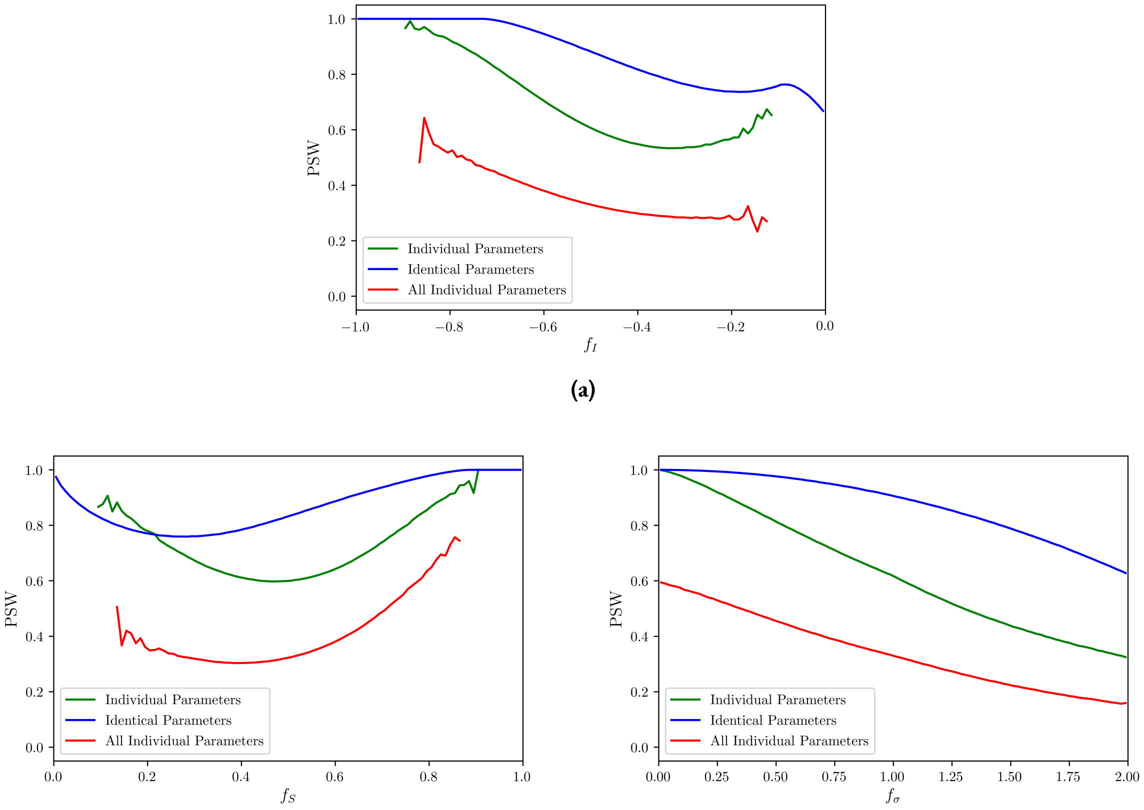
hensible picture of the situation. Doing that here yields figure 7.10.

The same regions as before are unstable. Upon taking a closer look at the actual bifurcations in these two situations, it appears that the cone-like surface has its highest point for a larger $f_S$ after adding the additional motive. Likewise, the Hopf surface intersects the $f_S$ axis at a slightly higher point this time around. This indicates that connecting multiple motives by node $S$ decreases stability slightly. This suggests that multiple customers put more strain on the system, which reduces stability on the system. The $f_S = 0$ and $f_I = 0$ planes are also saddle-node type bifurcations surface now. At the intersection of the Hopf curve with $f_I = 0$, there is one zero eigenvalue and two purely imaginary eigenvalues. This corresponds to a Gavrilov-Guckenheimer, also known as Fold-Hopf, bifurcation. The similarity between figures 7.10 and 7.4 gives rise to proposition 1:

**Proposition 1** *Stability behaviour for the small cyclic network is inherited by larger networks containing the small network as a sub-motive.*

The SE method results have been given in figure 7.11. An additional curve has been given, compared to 7.5. This time, $F_2$ and $F_8$ take place at node $S$ and are both impacted by both incoming and outgoing parts. The green curve represents the situation in which all parts have distinct elasticities. For instance, this implies that $f_8^{p2}$ still equals $f_2^{p2}$ (see equation 57). The red curve represents a situation in which all interactions have distinct elasticities. This introduces four new elasticities (two $f_S$ and two $f_I$). It is clear from the figure that this rather small change (splitting four elasticities into eight elasticities, compared to splitting three elasticities into 22 elasticities) severely enhances the instability in this system. That is, when the outflow of different parts and, by extension, orders of different customers are treated differently by the store, then the system becomes much more unstable. There is even more difference between the red and green curve than between the blue and green curve. As this difference is so huge, it is worthwhile to reiterate this result in the shape of a proposition:

**Proposition 2** *Arbitrarily treating different customers differently, significantly decreases the overall stability of the networks in multi-customer feedback flow networks.*



**(a)**

**(b)**

**(c)**

**Figure 7.11:** Results of the SE method performed on the network from figure 7.12 and its equations, with an ensemble of identical parameters in blue, an ensemble of individual parameters in green, in which each part has its own elasticities, and lastly an ensemble of individual parameters in red, in which each interaction has its own elasticity. $10^8$ samples were taken, divided over 100 intervals.

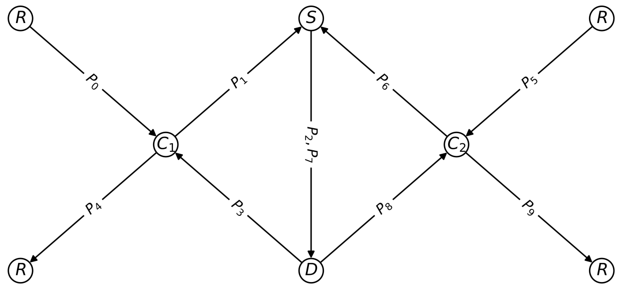


**Figure 7.12:** A network consisting of two simple motives, as shown in figure 7.1, merged at both the $S$ and $D$ node, with parts added on the edges.

From now on, in the following sections, only the latter way of independently sampling will be considered.

### 7.1.2 Equal $D$ nodes

After connecting the motives at node $S$, making the system symmetrical seemed to be an obvious way to change the situation. This situation corresponds to the customers being closer to each other in reality, meaning that they are nearby neighbourhoods or postal zones, and that products can thence be delivered by the same distribution centre. On the other side, node $D$ could also be more broadly regarded as a single postal company, which necessarily deals with all the deliveries of the store. These different ways to consider the $D$ nodes lead to different interpretations of the elasticities, but should ultimately result in similar behaviour. To keep the situation easily visualizable, the former concept is kept in this thesis. This network is given in figure 7.12.

This time around, the motives are connected at both the $S$ and $D$ nodes, which changes equation 57 slightly in the following way:

$$\begin{aligned} \dot{P}_2 &= F_2(P_1, P_2, P_6, P_7) - F_3(P_2, P_3, P_7, P_8), \quad \dot{P}_7 = F_8(P_1, P_2, P_6, P_7) - F_9(P_2, P_3, P_7, P_8) \\ \dot{P}_3 &= F_3(P_2, P_3, P_7, P_8) - F_4(P_3, P_4), \quad \dot{P}_8 = F_9(P_2, P_3, P_7, P_8) - F_{10}(P_8, P_9) \end{aligned} \tag{58}$$

This hence includes the consideration of the orders by both customers and the schedules for both of these orders in the creation of these schedules. Moreover, it also includes the consideration of both schedules and both deliveries for the manifestation of these deliveries.

As it turns out, a zero eigenvalue exists in the entire parameter space, which causes the determinant to be zero everywhere. Consequently, saddle-node type bifurcations cannot be found by solving $Det(J) = 0$ anymore. The constant term in the characteristic polynomial of $J$ is always zero. To actually find the bifurcation surfaces this time around, the coefficient of the linear term in the characteristic polynomial has to be equal to zero, as then the system contains two zero eigenvalues on a bifurcation surface now.

However, as a zero eigenvalue is present in the entire parameter space, stability cannot be determined as easily as in previous sections. This zero eigenvalue is related to the symmetry in the system of differential equations, and hence to a conserved quantity. By making both $P_2$ and $P_7$ dependent on the same $P$'s, with the same elasticities, $\dot{P}_2$ and $\dot{P}_7$ become locally identical around the steady state. Hence, they are linked and when one of them is known, another is as well. This observation does however not help determining the stability of the steady state.

Let us therefore consider a toy problem, in which the $F_i$ are simply chosen to be a power of all its arguments, to attempt to determine the stability. For instance;

$$\begin{aligned} \dot{P}_0 &= P_1^{f_I} - P_0^{f_S} P_1^{f_I} P_3^{f_\sigma}, \\ \dot{P}_2 = \dot{P}_7 &= P_1^{f_S} P_2^{f_I} P_6^{f_S} P_7^{f_I} - P_2^{f_S} P_3^{f_I} P_7^{f_S} P_8^{f_I}, \end{aligned} \tag{59}$$

and the other $\dot{P}_i$ are defined analogously.

Letting Maple determine the general steady states in such a situation yields

$$P_i = 1,\ \forall i \neq 2, 7, \quad P_7 = \frac{1}{P_2}, \tag{60}$$

in which $P_2$ is free, and another steady state in which $P_i$ are alternating positive and negative. As we require $P_i > 0$, this latter steady state won't be regarded. Equation 60 thence gives a steady state line, corresponding with the eigenvector belonging to the global zero eigenvalue.

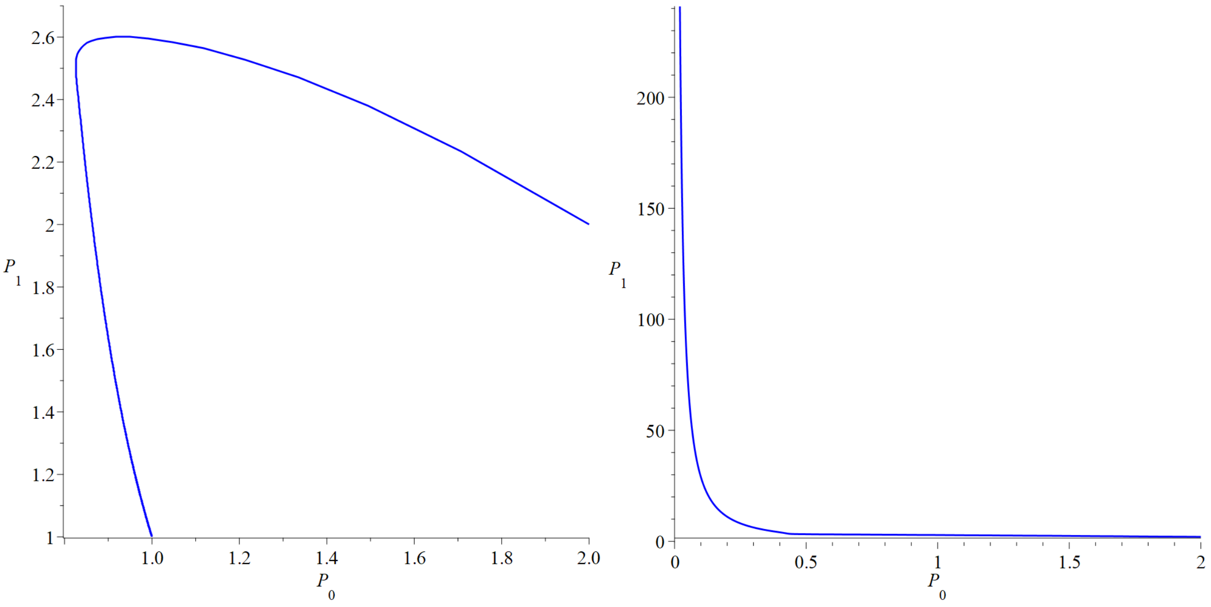
Arbitrarily taking $P_i(0) = 2,\ \forall i$ and letting Maple integrate the equation yields images like in figure 7.13. For $f_\sigma = 2$, $(f_I, f_S) = (-1, 1)$ is located outside the cone and hence there are only non-positive eigenvalues (of which one is of course zero). This corresponds with figure 7.13a, in which $P_0$ quickly drains to about $0.8$ and $P_1$ increases to about $2.6$ after which $P_1$ decreases to $1$, along with $P_0$ increasing a bit to $1$. For all $P_i$ the time evolutions are stable and move towards their steady states, typically $1$, with exception of $P_2$ and $P_7$, which move to each others inverse, which might incidentally be $1$. Moreover, $(f_I, f_S) = (-\frac{1}{2}, \frac{1}{2})$ is located inside the cone and thence there is at least one positive eigenvalue. Here, $P_0$ quickly drains to almost $0$, after which $P_1$ completely diverges. This behaviour has been given in figure 7.13b. This has been tested for various initial conditions and seems to hold quite generally, which indicates that the stability regions keep their stability properties from previous sections. This also makes sense, as otherwise the entire parameter space would suddenly become unstable, even for very weak elasticities.
In figure 7.14, two extra time evolutions have been given for parameter values just next to the blue Hopf surface, visible in figure 7.15. 7.14a gives a time evolution just to the left of the blue Hopf surface, and 7.14b gives a time evolution just to the right of the blue Hopf surface, in which a stable limit cycle can be seen around the steady state, and hence the system does not converge to the steady state.

Due to the size of the system, the non-linearities present, and the connectedness of all parts it is however hard to formally prove this result in this case, let alone in general, and hence it is only demonstrated in this way.

The changes applied in this section lead to the bifurcation diagram in figure 7.15. This diagram is similar to that of figure 7.9 and retains the same surfaces. One of the cone-like real Hopf surfaces (not shown) has shrunk, but overall the remainder retains a similar shape as before. However, the large saddle-node bifurcation cone has increased in size again, as has the Hopf surface, which yet again indicates that connecting the motives by both nodes $S$ and $D$ destabilizes the network further, giving rise to proposition 3:

**Proposition 3** *Having one node in a network perform the same role in multiple sub-networks decreases network stability.*

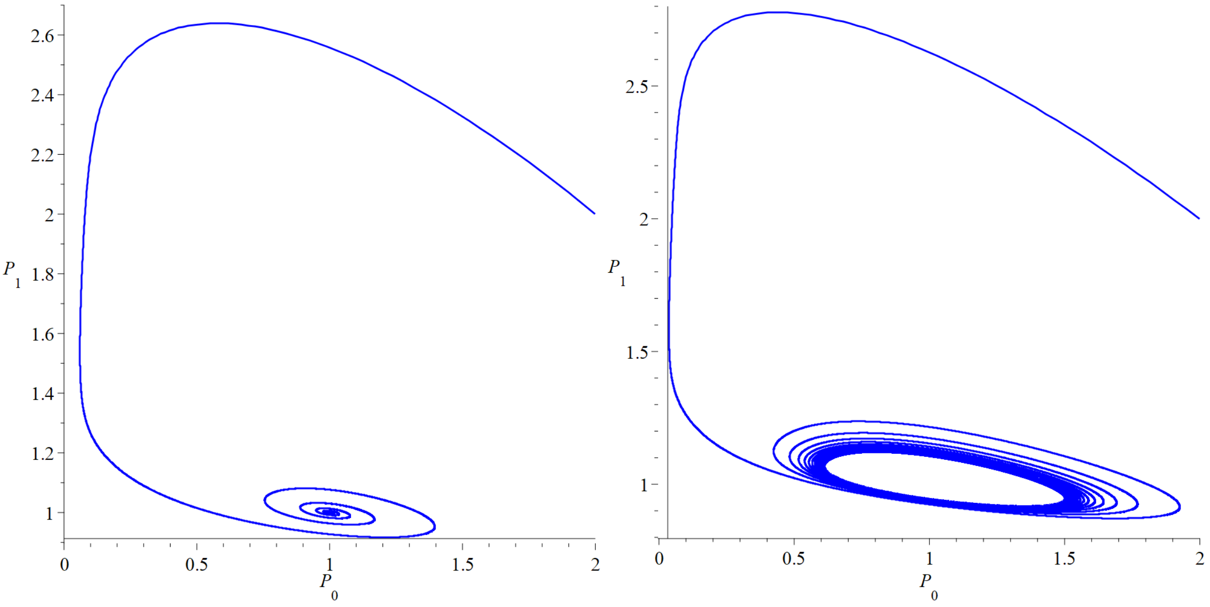
Due to the global zero eigenvalue, and floating point precision, approximately half of the eigenvalue sets will contain a positive eigenvalue. This can be remedied by only counting the unstable eigenvalue as valid when it is larger than some $\varepsilon$, which has to be chosen somewhere between $10^{-15}$ and $10^{-8}$ depending on the size and complexity of the network. Usually, larger networks, and networks with more interactions, require a larger



**(a)** $f_I = -1$ and $f_S = 1$.

**(b)** $f_I = -\frac{1}{2}$ and $f_S = \frac{1}{2}$

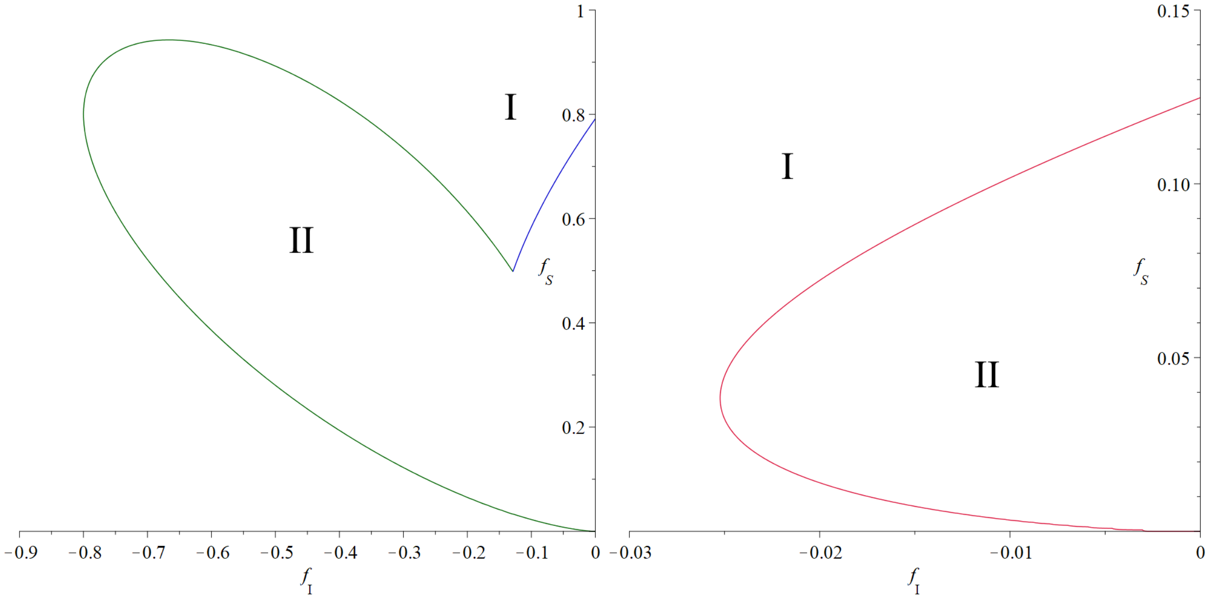
**Figure 7.13:** Time evolutions of $P_0$ and $P_1$ for $f_\sigma = 2$ and different values for $f_I$ and $f_S$, which place them in different stability regions. Subfigure a demonstrates a stable equilibrium, whereas subfigure b shows an unstable equilibrium. The initial conditions are in both cases $P_i(0) = 2\ \forall i$.



**(a)** $f_I = -\frac{1}{10}$ and $f_S = \frac{7}{10}$.

**(b)** $f_I = -\frac{9}{100}$ and $f_S = \frac{6}{10}$.

**Figure 7.14:** Time evolutions of $P_0$ and $P_1$ for $f_\sigma = 2$ on both sides of the Hopf surface. The initial conditions are in both cases again $P_i(0) = 2,\ \forall i$.



**(a)** Stability region projection on $f_\sigma = 2$.

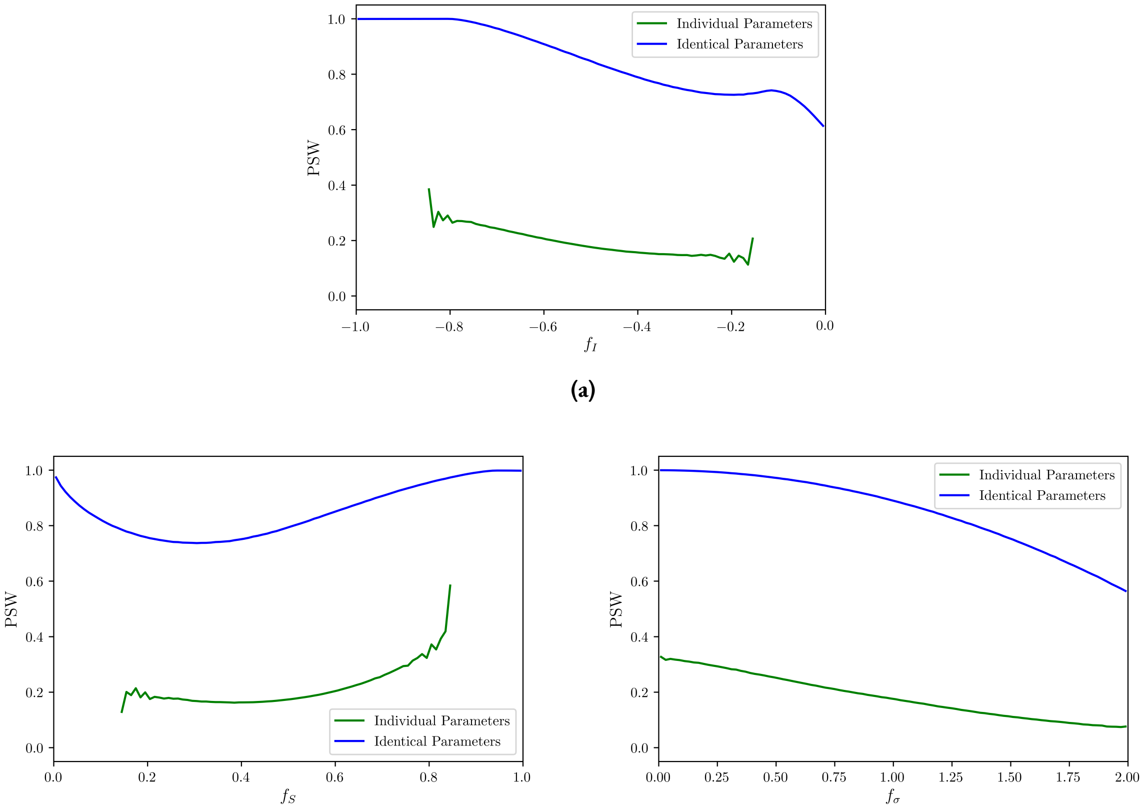
**(b)** Stability region projection on $f_\sigma = -2$.

**Figure 7.15:** The two-dimensional projections on $f_\sigma = 2$ and $f_\sigma = -2$ of the bifurcation surfaces in the network from figure 7.12, containing two motives joined at nodes $S$ and $D$. The same stability regions are present as in figure 7.10; region I is stable and region II is unstable. To the left of the red Hopf curve, the system is unstable and to the right the system is stable. In figure a, the blue curve on the right is a Hopf bifurcation and the green curve on the left is part of a saddle-node type bifurcation. The red curve in figure b is a Hopf bifurcation.

choice of this $\varepsilon$. When increasing the size of the network even further, one has to be even more careful, as $10^{-8}$ might not be small enough, but eigenvalues other than the global zero one might become small as well and wrongfully omitted. This is however no issue when sampling individually, as there are no global zero eigenvalues present in those cases.

By extension, the sensitivities and influences cannot be calculated with the definition if a global zero eigenvalue is present. Those terms are thence chosen not to be included in the sum in the ln, as they do not contribute to changes in the network anyway. They rather pertain to the line of steady states, and hence do not impact the remainder of the network.

Lastly, the results of the SE method, given in figure 7.16 reflect those of figure 7.11. The change in stability for the identical samples is marginal, but present, whereas the change in stability for the individual samples is larger. Their curves have a significantly smaller slope at the boundaries; the $f_\sigma$ and $f_I$ figures are much flatter, while the $f_S$ figure curves much less at both sides.



**(a)**

**(b)**

**(c)**

**Figure 7.16:** Results of the SE method performed on the network from figure 7.12 and its equations, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green, with $10^8$ taken samples, divided over 100 intervals.

### 7.1.3 Extending the model; adding elasticity to co-production, $f_C$

This set of equations can be further enhanced, with a similar treatment as in [1, 2]. $P_2$ and $P_7$, together with $P_3$ and $P_8$ are created in parallel. This gives rise to the before mentioned elasticity to co-production, $f_C$, which, like $f_I$ is varied for $-1 \leq f_C \leq 0$. An example of a situation in which this elasticity is present and relevant is for instance around holidays. Customers order products en masse. If the store has enough stock, the postal services suffer, because suddenly a lot more orders need to be delivered. As a result, the orders take longer to deliver. This corresponds with a negative $f_C$; this $f_C$ is basically a special kind of $f_I$, which can be taken for $f_2^{p7}$, $f_8^{p2}$, $f_3^{p8}$ and $f_9^{p3}$. However, the productions of $P_2$ and $P_7$ have not been assumed to compete for each other. Rather, as no reservoir supplies store $S$ with products, the stock of $P_2$ and $P_7$ have implicitly been assumed infinite.

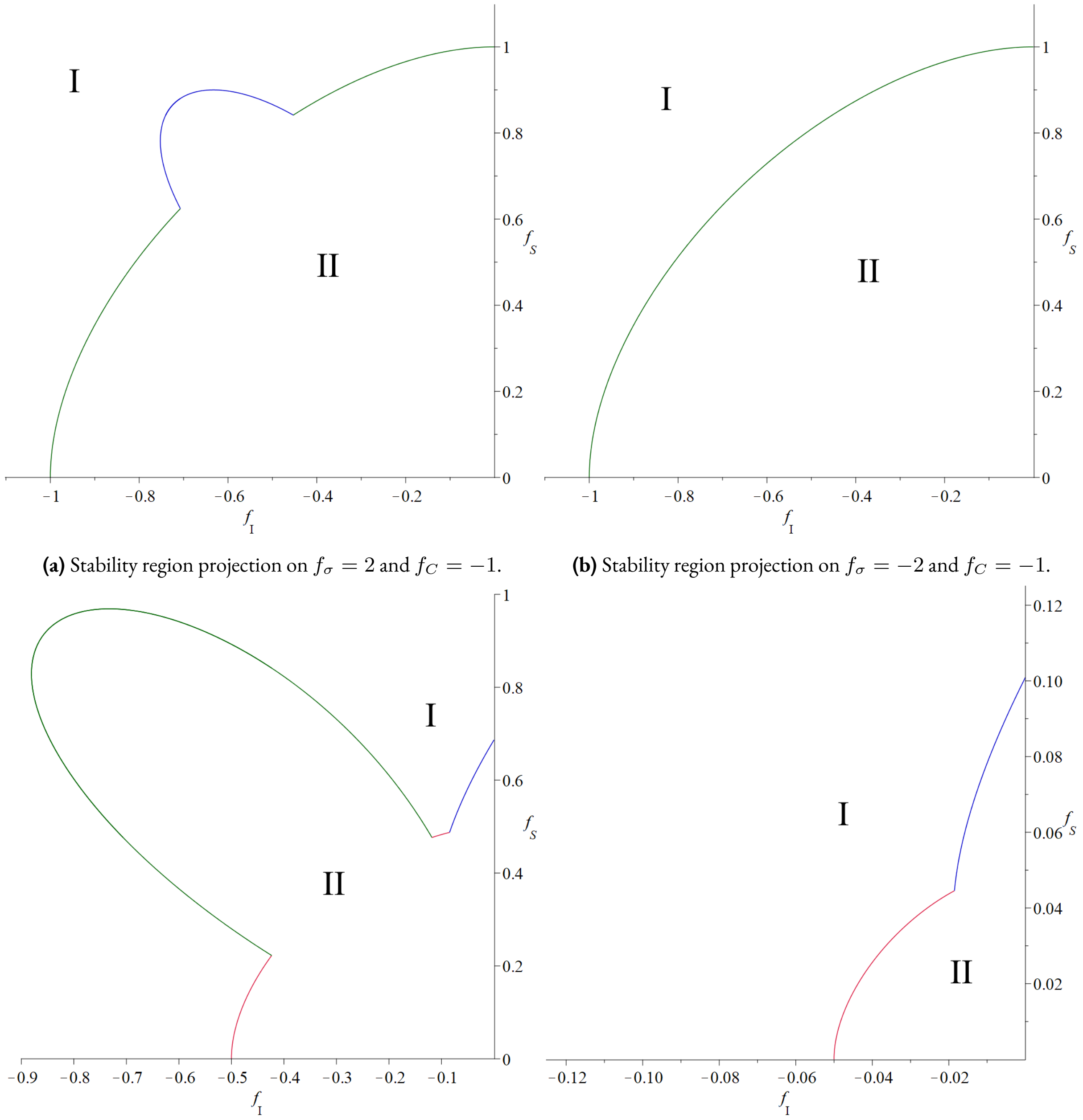
As a result, only $f_3^{p8}$ and $f_9^{p3}$ are taken to equal $f_C$ here, as they can compete for the availability of deliverers. The bifurcation diagram in this case is given by figure 7.17. At this point, adding real Hopf curves makes the image entirely unintelligible. They run rampant and squiggle all over the place inside the arc. Hence, only the stability regions are given here. Just like when joining the network at both $S$ and $D$, the determinant of the Jacobian in this case is 0. However, by the same reasoning as before, the stable regions can be found by omitting the global zero eigenvalue.

It turns out that the red arc in figure 7.17 constitutes another saddle-node type bifurcation surface, which is a cylinder-like surface with radius $-f_C$ around $f_S = f_I = 0$. The arc is not exactly circular; it is less concave than a circle. However, as it still resembles a circle closely enough, it will be referred to as such, and as a cylinder, taking $f_\sigma$ as its axis. The inside of that cylinder is unstable. The inside of the recurring cone-like surface remains unstable as before. The intersection of these curves gives rise to Bogdanov-Takens bifurcation, as does the intersection of the cone-like surface with the Hopf surface, like before. Lastly, as shown in figures 7.17c and 7.17d, the previously present Hopf surface may intersect with the arc. On this intersection, there exist two zero eigenvalues (including the global zero eigenvalue), and two purely imaginary eigenvalues, which again constitutes a Fold-Hopf bifurcation. It turns out that the bifurcation surface at $f_S = 0$ is now no longer present, but the $f_I = 0$ plane is still a saddle-node type bifurcation. Its intersection with the blue Hopf curve is again a Fold-Hopf bifurcation.

Lastly, when looking at figures 7.17a and 7.17c, it turns out that the size of the cone reduces a bit when reducing $f_C$. Conversely, the circle enlarges for decreasing $f_C$, which more than cancels the shrinking cone. Overall, the stability thence decreases for decreasing $f_C$.

This can readily be seen in the SE method results again, given in figure 7.18. This time around, there's a new figure in which the $PSW$s are given as a function of $f_C$. Again, the slopes are pretty tame, mostly when sampling $f_\sigma$ and $f_C$ individually. On the flip-side, the $f_I$ figure has more of a slope, but it also ends lower for $f_I \to 0$. The ensembles with identical parameters curve more drastically, but this effect is smallest in figure 7.18c. This can be understood as the only way $f_\sigma$ impacts the stability is with the cone, but the cone is often included in the cylinder for small $f_\sigma$.

One can intuitively understand this in the following way. $f_C$ can be interpreted as a special kind of $f_I$. Therefore, it fulfils a similar role and the difference between $f_I$ and $f_C$ determines which part is produced most as a result of a perturbation. Suppose somehow that a build-up of deliveries occurs, which perturbs the steady state, such that, say, $p_3$ is larger than 1 (this could have been $p_8$ as well). This corresponds with, for instance, some delivery driver being suddenly unavailable, and hence its packages are not delivered for the day. Immediately, the productions of both $p_3$ and $p_8$ lower to reduce strain on the delivery company. Therefore, $p_8 < 1$ just after the perturbation occurs. Now, $p_3 > 1$ and $p_8 < 1$. If $f_C < f_I$, the system favours build-up of the largest part and drainage of the smallest part. This drainage can persist for sufficiently small $f_S$, and for larger



**(a)** Stability region projection on $f_\sigma = 2$ and $f_C = -1$.

**(b)** Stability region projection on $f_\sigma = -2$ and $f_C = -1$.

**(c)** Stability region projection on $f_\sigma = 2$ and $f_C = -\frac{1}{2}$.

**(d)** Stability region projection on $f_\sigma = -2$ and $f_C = -\frac{1}{20}$.

**Figure 7.17:** The two-dimensional projections of the stability regions in the network containing two motives joined at nodes $S$ and $D$, with a newly considered elasticity $f_C$, for several combinations of $f_\sigma$ and $f_C$. The same stability regions are defined as usual; region I is stable and region II is unstable. In figures a and b, the blue and green curves are saddle-node type bifurcations. In figures c and d, both the green and red curves are saddle-node type bifurcations, whereas the blue curve is a Hopf bifurcation.

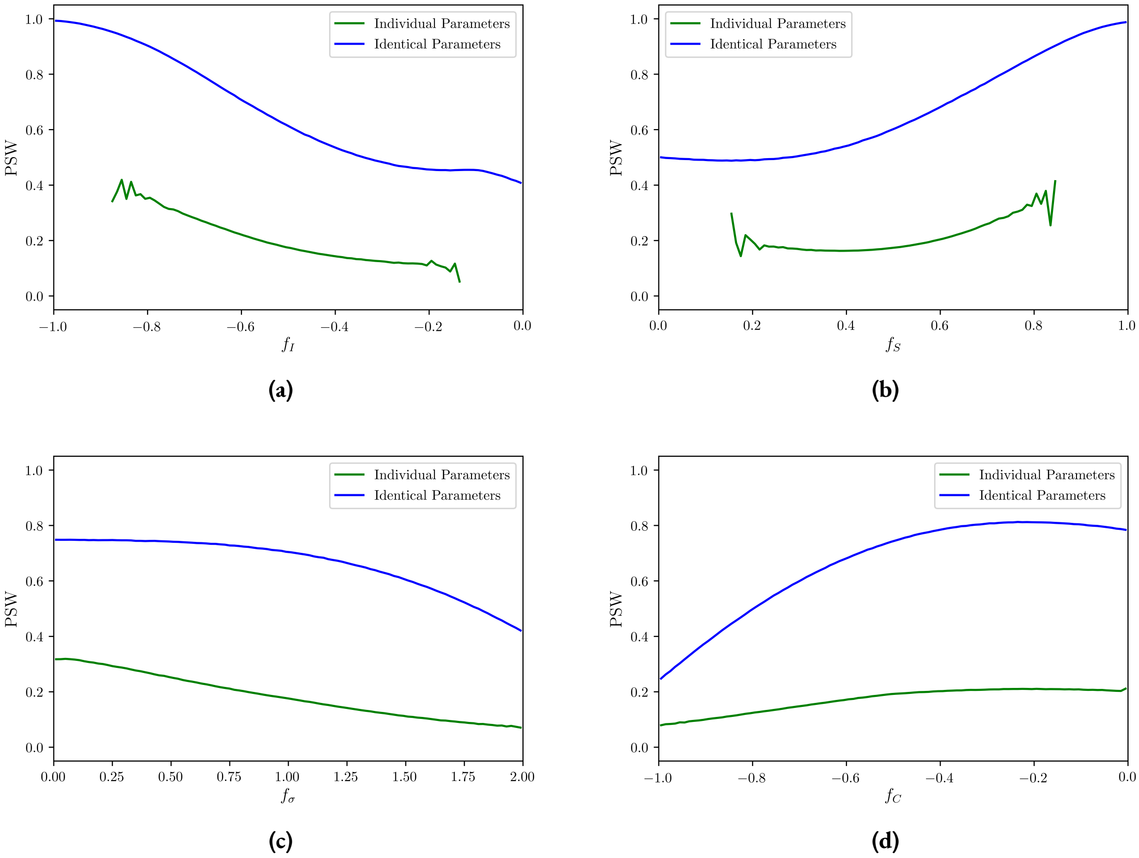


**Figure 7.18:** Results of the SE method performed on the network from figure 7.12 and its equations, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green. $f_C$ was added in this ensemble and $10^8$ samples were taken, divided over 100 intervals.

$f_S$ the system is robust enough to return to $p_i = 1, \ \forall i$. Hence, if $f_S$ and $f_I$ are small compared to $f_C$, the perturbation is allowed to progress, draining and accumulating parts in the network. The arc in figure 7.17 is the representation of the region in which $f_S$ and $f_I$ cannot compensate for $f_C$.
In brief, choosing $f_C < f_I$ gives rise to behaviour that worsens build-up and drainage on the parts that are co-produced. This behaviour can still be remedied by the impact of the $f_S$ in this network.

It is also striking that for decreasing $f_C$, figure 7.18d shows a maximum around $f_C = -0.2$, below which it rapidly decreases. This corresponds to the growing cylinder. For small $f_C$, the decrease in size of the cone briefly outweighs the increase in size of the cylinder, but that balance does not hold on for long.

The most important lesson to be learnt here is that the parameters should be chosen much more carefully to ensure a stable system. This translates to a more carefully crafted management policy employed by each company in the network, to ensure more stable outcomes. It turns out that introducing $f_C$ brings a lot of instability, if the $f_C$ are not chosen wisely with respect to the individual $f_I$s.

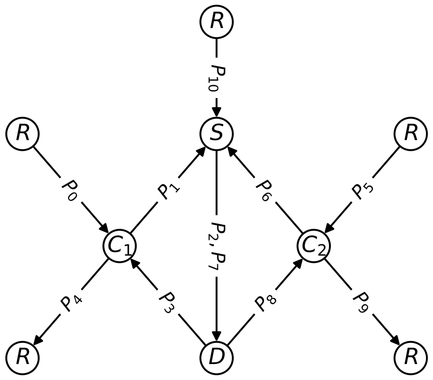


**Figure 7.19:** A network consisting of two simple motives, merged at both the $S$ and $D$ node, with an additional reservoir connected to $S$ node, with parts added on the edges.

### 7.1.4 Addition of a reservoir at $S$

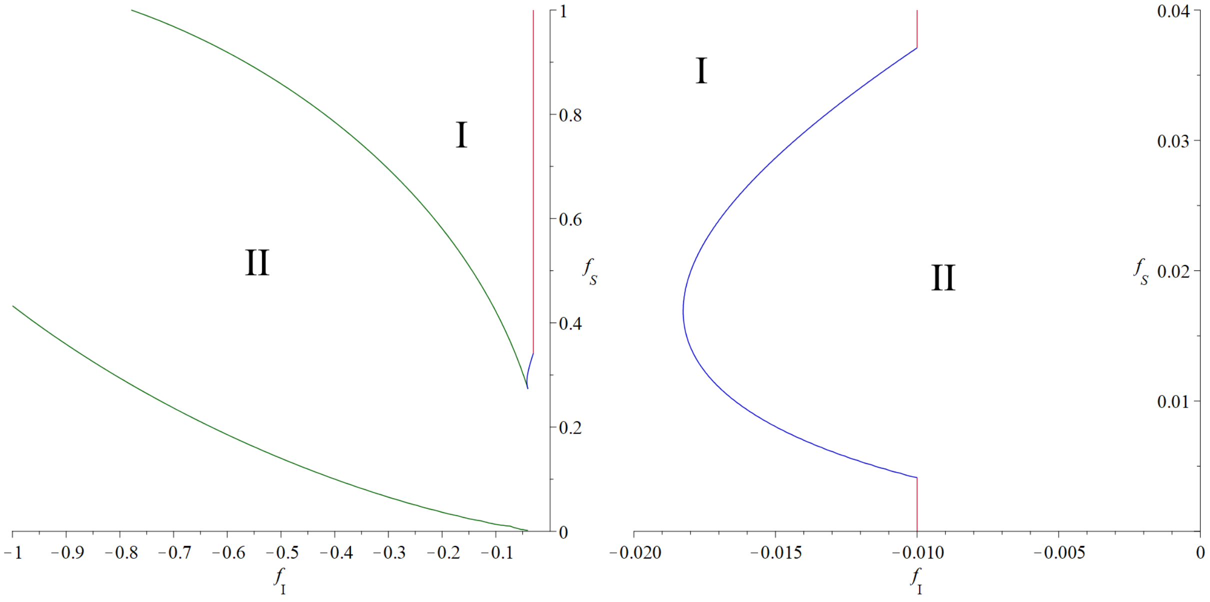
As discussed previously, to have a notion of stock, there has to be a notion of supply to the store. The most straightforward implementation of this supply is an external reservoir, added to figure 7.12 as shown in figure 7.19, which can be described by a new $F_{12}$ which creates a new $P_{10}$;

$$\begin{aligned} &F_2 \to F_2(P_1, P_2, P_6, P_7, P_{10}), \quad F_8 \to F_8(P_1, P_2, P_6, P_7, P_{10}), \\ &\dot{P}_{10} = F_{12}(P_{10}) - F_2(P_1, P_2, P_6, P_7, P_{10}) - F_8(P_1, P_2, P_6, P_7, P_{10}), \end{aligned} \tag{61}$$

in which it is implicitly assumed that one order placement and one product are needed for the respective store to be able to schedule an order. This change implies that $F_{12}^* = 2F$, in which $F$ is again $F_i^*$ for all other $i$. This change subsequently also implies that there exists a new co-production from the supply to the stores; the orders shipped out from the store are now co-produced from the same resource $P_{10}$. Hence, there are now four elasticities to co-production; the two new elasticities being $f_2^{p7}$ and $f_8^{p2}$.

Having assumed all of this, figure 7.20 depicts the stability regions for this network. It turns out that $Det(J) = 0$ can be factored into the familiar cone-like surface, the newly introduced line $f_I = f_C$ and the familiar cylinder with radius $-f_C$. If $f_C \leq f_I \leq 0$, the system is unstable; this region includes the cylinder at all times, and it does not affect the stability, as crossing it changes the sign of another eigenvalue. This $f_I = f_C$ line is a saddle-node bifurcation. The cone remains a saddle-node bifurcation and the blue wedge-shaped surface is a Hopf bifurcation as before. At their intersection in figure 7.20a lies the familiar Bogdanov-Takens bifurcation and at the intersection of the $f_I = f_C$ surface with the Hopf surface is a Fold-Hopf bifurcation point. Lastly, the intersection of the $f_I = f_C$ surface with the cone-like surface (not visible, but present in region II) is an additional Bogdanov-Takens bifurcation. The $f_I = 0$ surface is now a Bogdanov-Takens bifurcation, as both the determinant and resultant of $J$ are zero there. The intersection of the Hopf surface with the $f_I = 0$ surface (also not visible) is therefore a co-dimension 3 bifurcation.

The system is, again, also unstable inside the cone-like surface, and hence the stability regions are easily elicited. It turns out that the cone-like surface is quite a bit larger (even when compensating for the difference in $f_C$ between figure 7.20 and 7.17). It is hence clear that mostly $f_C$ determines the stability of the network this time around; the remainder of the figure is similar to 7.17, barring the significant increase in instability. Other



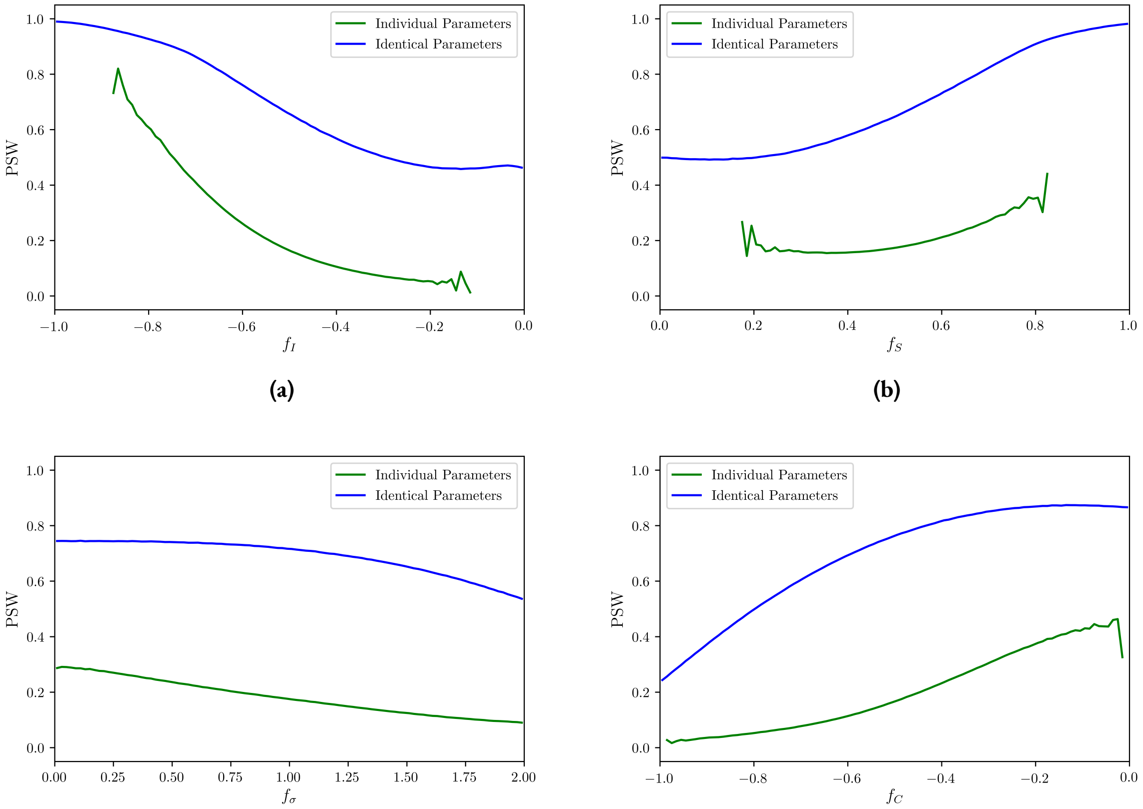
**(a)** Stability region projection on $f_\sigma = 2$ and $f_C = -\frac{3}{100}$. **(b)** Stability region projection on $f_\sigma = -2$ and $f_C = -\frac{1}{100}$.

**Figure 7.20:** The two-dimensional projections of the stability regions in the network containing two motives joined at nodes $S$ and $D$, with a reservoir at $S$, for $f_\sigma = 2$ and $f_\sigma = -2$. The same stability regions are defined as usual; region I is stable and region II is unstable. The green and red curves are saddle-node type bifurcations, and the blue curve is a Hopf bifurcation.

values of $f_C$ have been omitted, as they only shift the $f_I = f_C$ and doing so does not add qualitative new behaviour.

The $f_C = f_I$ plane dividing the stability regions originates from the addition of co-production of parts 2 and 7, while they need stock to be produced. If the system is perturbed such that, say, $p_2 > 1$ (or $p_6$), then briefly $p_2$ decreases and so does $p_6$, such that it is slightly below 1. Like when adding $f_C$ in section 7.1.3, both drainage and build-up are worsened over time if $f_C < f_I$. In contrast to section 7.1.3, $f_S$ cannot contain this phenomenon. Because there is an excess of $p_2$, the elasticity to co-production inhibits the production of $p_6$. $p_{10}$ also builds up, as neither $p_2$, nor $p_6$ can be produced. Changing $f_S$ cannot solve this problem. Therefore, the projection of the bifurcation diagram in the $f_I$,$f_S$-plane shows that changing $f_S$ does not possibly change stability right of the $f_I = f_C$ line.
In brief, taking $f_C < f_I$ again worsens the build-up and drainage of parts. However, this time, the stock for the store, $p_{10}$ also builds up and this cannot be remedied.

The SE results are given in figure 7.21. For individually sampled parameters, it is striking how unstable the network again is. Decreasing $f_I$ still hugely affects the stability of the network; $f_I \to -1$ increases the stability by a lot, while $f_I \to 0$ makes the system have almost 0 $PSW$. However, $f_S$ and, even moreso $f_\sigma$ barely impact the overall stability. This is due to the behaviour the network inherits from the stability properties of the surface $f_I = f_C$ in the bifurcation diagram, and hence in the identical ensemble. Lastly, $f_C \to 0$ also drastically reduces the number of unstable networks, of course, as the only instability is introduced by the inside of the conic curve. $f_C \to 0$ appears to be more stable in figure 7.21, compared to figure 7.18, which differs from the results found with the bifurcation analysis. This is an example of individual ensembles not matching well with the simplified bifurcation analysis. $f_C \to -1$ is less stable in this case, compared to figure 7.18. This at least partially holds due to the introduction of more $f_C$s in figure 7.21, and corresponds with the line $f_I = f_C$ dividing the stability regions in the bifurcation diagram.



**(a)**

**(b)**

**(c)**

**(d)**

**Figure 7.21:** Results of the SE method performed on the network from figure 7.12 and its equations, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green. $f_C$ was added in this ensemble and $10^8$ samples were taken, divided over 100 intervals.

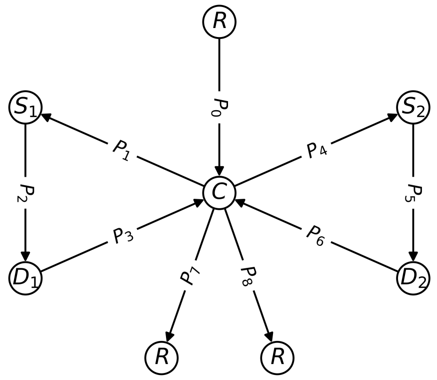


**Figure 7.22:** A network consisting of two simple motives, merged at the $C$ node, with parts added on the edges.

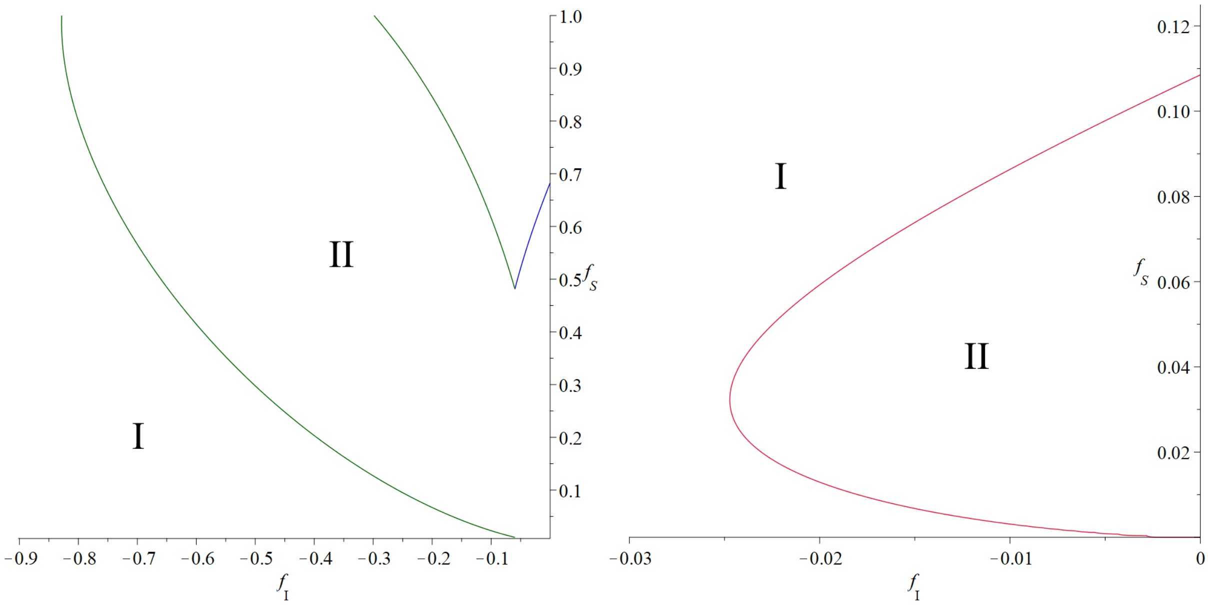
## 7.2 Competition: Two stores and one customer

Another interesing dynamic is the interaction when two stores compete with each other. A first attempt at analyzing this includes one customer, two stores and two delivery companies as given in figure 7.22:

$$\begin{cases} \dot{P}_0 = F_0(P_0) - F_1(P_0, P_1, P_3, P_4, P_6) - F_4(P_0, P_1, P_3, P_4, P_6) \\ \dot{P}_1 = F_1(P_0, P_1, P_3, P_4, P_6) - F_2(P_1, P_2) \\ \dot{P}_2 = F_2(P_1, P_2) - F_3(P_2, P_3) \\ \dot{P}_3 = F_3(P_2, P_3) - F_7(P_3, P_7) \\ \dot{P}_4 = F_4(P_0, P_1, P_3, P_4, P_6) - F_5(P_4, P_5) \\ \dot{P}_5 = F_5(P_4, P_5) - F_6(P_5, P_6) \\ \dot{P}_6 = F_6(P_5, P_6) - F_8(P_6, P_8) \\ \dot{P}_7 = F_7(P_3, P_7) - F_9(P_7) \\ \dot{P}_8 = F_8(P_6, P_8) - F_{10}(P_8) \end{cases} \tag{62}$$

In this model, $P_0$ again represents the social influence on the customer. $P_1$ and $P_4$ are orders placed on both stores, $P_2$ and $P_5$ are schedules created by the stores and $P_3$ and $P_6$ are deliveries performed by the delivery companies. Lastly, $P_7$ and $P_8$ are consumptions of $P_3$ and $P_6$. In steady state, we find that $F_i^* = F, \ \forall i \neq 0$, $F_0^* = 2F$. The corresponding Jacobian of this system reads, taking all $P_i^* = P$ equal:

$$J = \frac{F^*}{P^*} \begin{pmatrix} 2f_I - 2f_S & -2f_I & 0 & -2f_\sigma & -2f_I & 0 & -2f_\sigma & 0 & 0 \\ f_S & f_I - f_S & -f_I & f_\sigma & f_I & 0 & f_\sigma & 0 & 0 \\ 0 & f_S & f_I - f_S & -f_I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & f_S & f_I - f_S & 0 & 0 & 0 & -f_I & 0 \\ f_S & f_I & 0 & f_\sigma & f_I - f_S & -f_I & f_\sigma & 0 & 0 \\ 0 & 0 & 0 & 0 & f_S & f_I - f_S & -f_I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & f_S & f_I - f_S & 0 & -f_I \\ 0 & 0 & 0 & f_S & 0 & 0 & 0 & f_I - f_S & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & f_S & 0 & f_I - f_S \end{pmatrix} \tag{63}$$



**(a)** Stability region projection on $f_\sigma = 2$.

**(b)** Stability region projection on $f_\sigma = -2$.

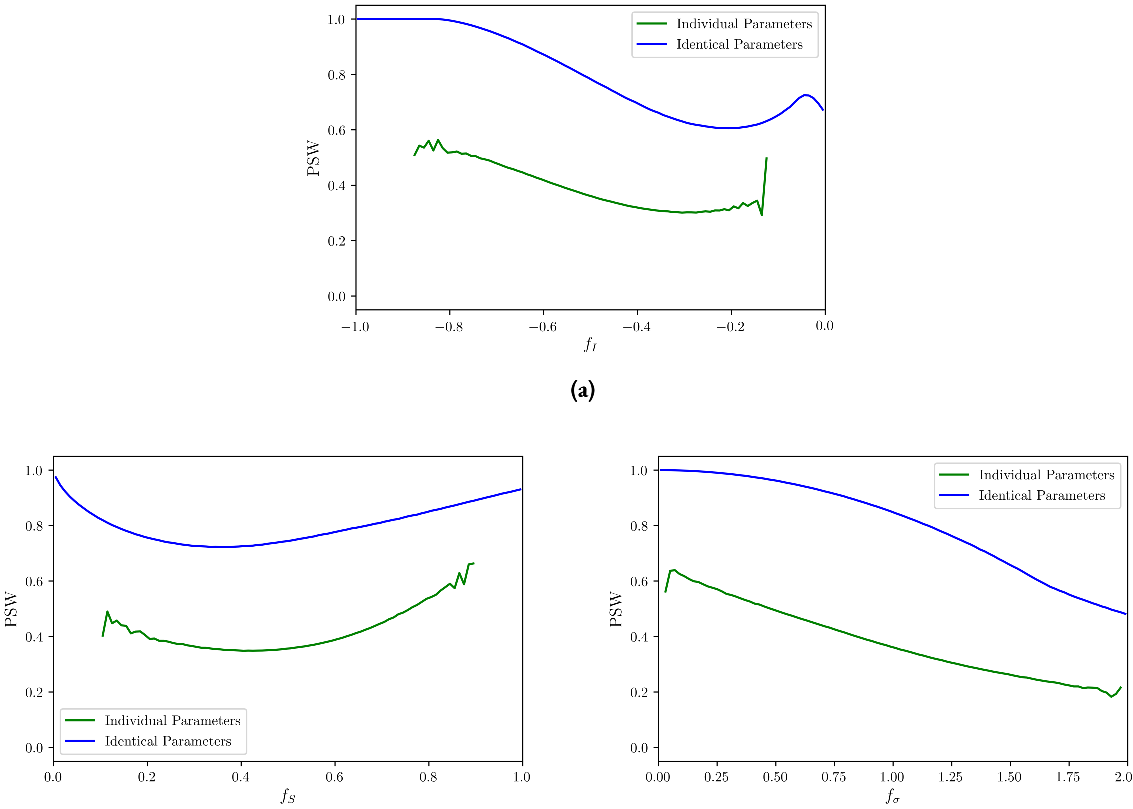
**Figure 7.23:** The two-dimensional projections of the stability regions in the network containing two stores and one customer, shown in figure 7.22, for $f_\sigma = 2$ and $f_\sigma = -2$ . As usual, region I is stable and region II is unstable. The green curve is a saddle-node type bifurcation, and the blue and red curves are Hopf bifurcations.

It stands out that the cone-like surface is much taller than before. This is even without any extra interactions; just the two stores and the one customer, without the addition of an $f_C$ or reservoirs at the $S$. This indicates that $f_S$ has a much less stabilizing effect in this network. Compared to its counterpart network from figure 7.8, the instability regions figure 7.23 are much larger. This can be concluded in proposition 4:

**Proposition 4** *Two stores competing for customers makes a network less stable than two customers ordering the products from one store*

The only qualitative difference, when compared to figure 7.9, is that now only $f_S = 0$ is a saddle-node type bifurcation, and $f_I = 0$ is not.

When comparing the SE method results, presented in figure 7.24, one can find that the flat region with $PSW = 1$ in the identical ensemble is gone when plotting against $f_S$ and is significantly smaller when plotting against $f_I$. Moreover, in the identical ensemble, increasing $f_\sigma$ brings more instability than before. When sampling individually, the behaviour is very similar. This is again a prime example of te benefit of the SE method, compared to the bifurcation analysis.



**(a)**

**(b)** **(c)**

**Figure 7.24:** Results of the SE method performed on the network from figure 7.22 and its equations, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green. $f_C$ was added in this ensemble and $10^8$ samples were taken, divided over 100 intervals.

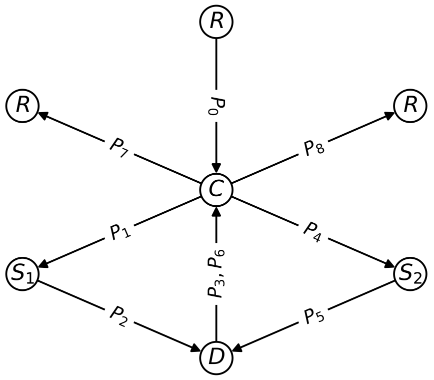


**Figure 7.25:** A network consisting of two simple motives, merged at the $C$ node, with parts added on the edges.

### 7.2.1 Only one deliverer

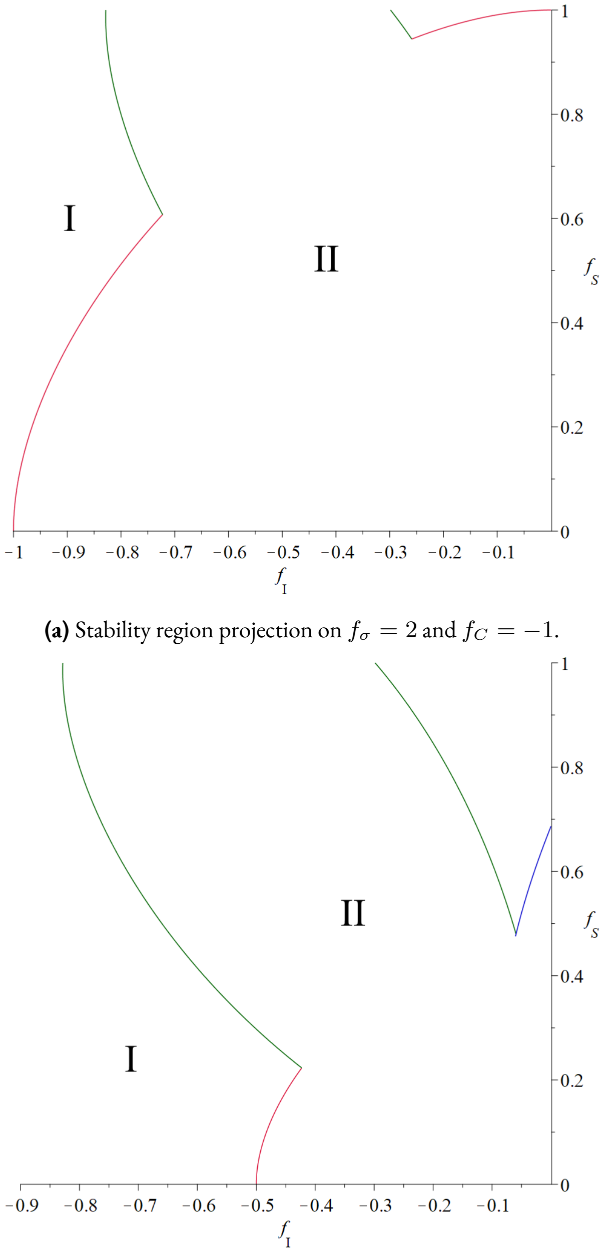
If, additionally, only one deliverer exists, the equations are modified slightly, such that

$$F_3 \to F_3(P_2, P_3, P_5, P_6), \quad F_6 \to F_6(P_2, P_3, P_5, P_6) \tag{64}$$

This network is given in figure 7.25, and the projections of its bifurcation surfaces are given in figure 7.26. By the introduced symmetry, this system has a global zero eigenvalue, which should be kept in mind again, when constructing the stability regions and performing the statistical ensemble.

These stability diagrams are similar to those in figure 7.17. When comparing, the saddle-node and wedge-like Hopf surfaces are much larger and hence decrease the overall stability. This also holds for the Hopf branch for negative $f_\sigma$ (not shown). It reaches the $f_S$ axis at around $f_S = 0.12$, which is greater than in figure 7.17d. The red arc surface is precisely the same as the one in section 7.1.3, and it exists here for the same reason. In this network, there aren't any boundary bifurcations on $f_I = 0$ or $f_S = 0$.

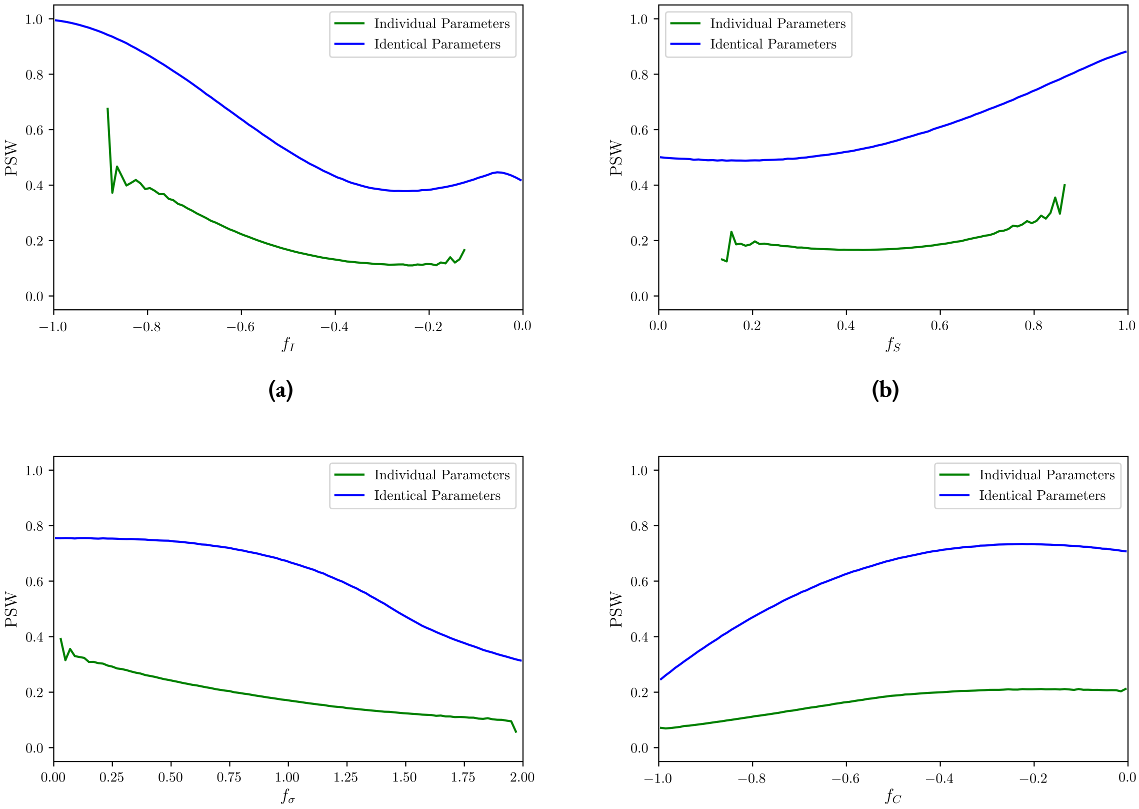
The SE method results are given in figure 7.27, which are, again, comparable to figure 7.16. However, the stability for the identical ensemble is significantly lower, especially in figure 7.27c, which reflects how the cone is larger with two stores, than it is with two customers. The drop around $f_\sigma = 1.25$ occurs due to the cone on average touching the cylinder at that point, and hence bringing more instability. The remainder of the analysis is analogous to that of section 7.1.3.



**(a)** Stability region projection on $f_\sigma = 2$ and $f_C = -1$.

**(b)** Stability region projection on $f_\sigma = 2$ and $f_C = -\frac{1}{2}$.

**Figure 7.26:** The two-dimensional projections on $f_\sigma = 2$ of the stability regions in the network containing two stores and one customer joined at nodes $S$ and $D$, as sown in figure 7.25, with a newly considered elasticity $f_C$. In this figure, the green and red curves are saddle-node type bifurcations, and the blue curve is a Hopf bifurcation.



**(a)** **(b)**

**(c)** **(d)**

**Figure 7.27:** Results of the SE method performed on the network from figure 7.12 and its equations, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green. $f_C$ was added in this ensemble and $10^8$ samples were taken, divided over 100 intervals.

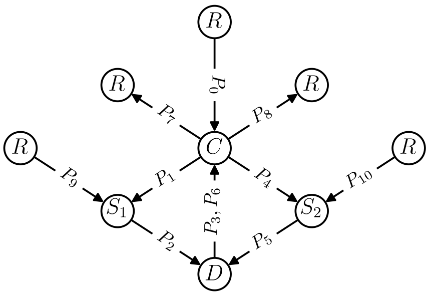


**Figure 7.28:** A network consisting of two simple motives, merged at the $C$ node, with parts added on the edges.

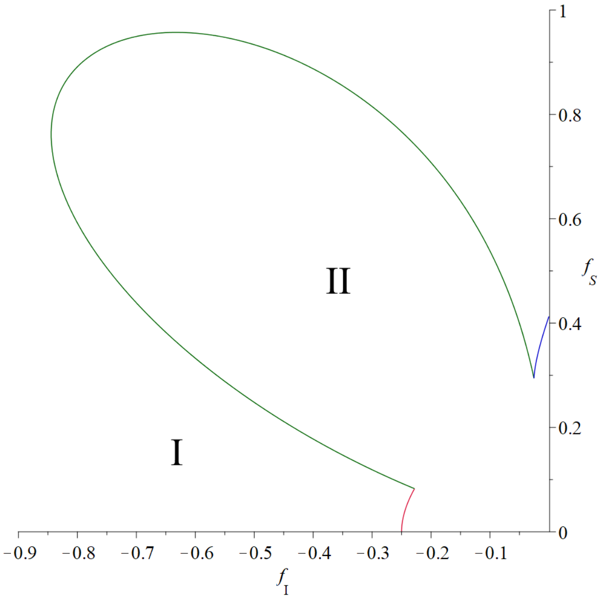


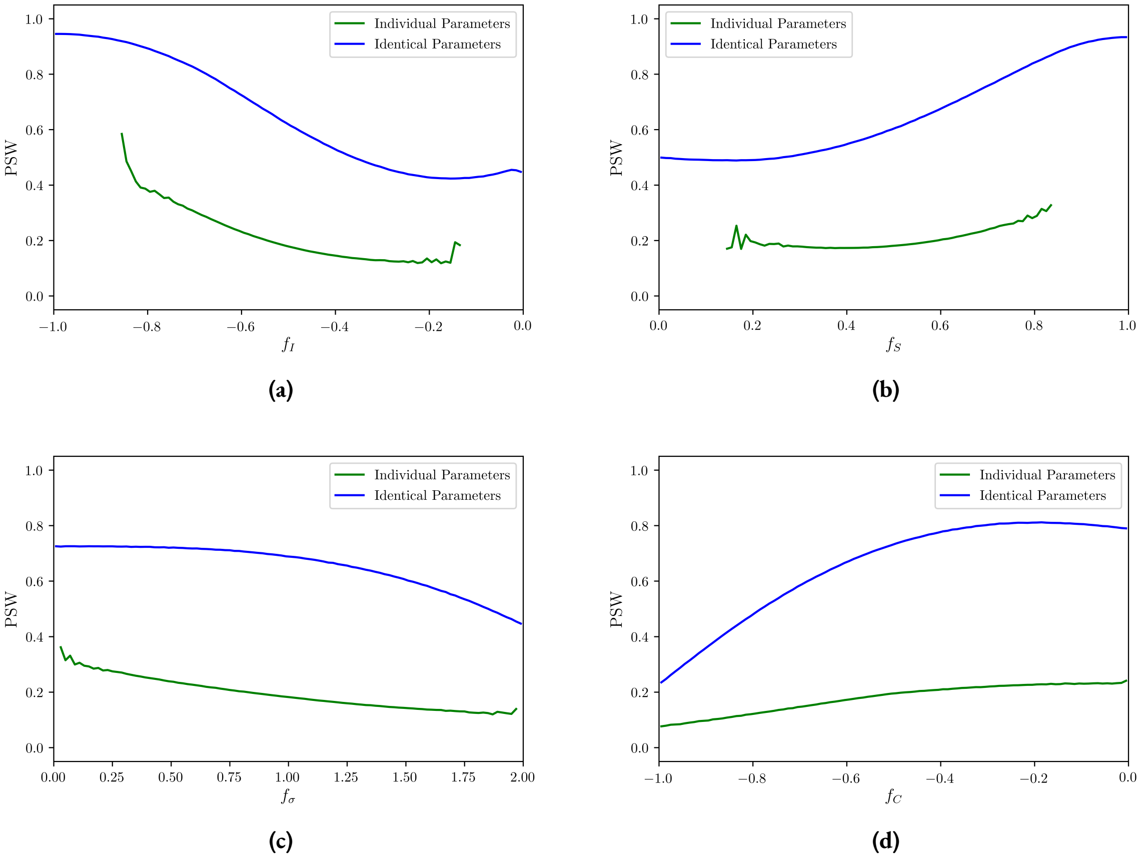
**Figure 7.29:** The two-dimensional projections on $f_\sigma = 2$ of the stability regions in the network containing two motives joined at nodes $S$ and $D$, with reservoirs connected at the $S$ nodes. The figure is generated for $f_C = -\frac{1}{4}$. The green and red curves are again saddle-node type bifurcations, and the blue curve is a Hopf bifurcation.

### 7.2.2 Addition of reservoirs at the $S$ nodes

Lastly, the impact of the addition of two reservoirs and, by extension, stock at the store nodes, is studied when there are two stores. Its network is given in figure 7.28.

The effect of these reservoirs is investigated by introducing two additional parts, $P_9$ and $P_{10}$, which add two transformation terms: $F_{11}(P_9)$ and $F_{12}(P_{10})$, and consequently $\dot{P}_9 = F_{11} - F_2$, $\dot{P}_{10} = F_{12} - F_5$. Moreover, $F_2 \to F_2(P_1, P_2, P_9)$ and $F_5 \to F_5(P_4, P_5, P_{10})$.

Figure 7.29 shows the stability diagram for positive $f_\sigma$ only, as showing a more exhaustive set of images feels redundant at this point. The familiar cone and cylinder are again saddle-node bifurcations, and the blue wedge-like surface is again a Hopf-bifurcation. Their intersection is as usual a Bogdanov-Takens bifurcation. Where the cylinder and cone meet, lies a Bogdanov-Takens bifurcation, and the intersection of the cone-like surface and the Hopf surface constitutes a Fold-Hopf bifurcation. The $f_I = 0$ plane is now again a Bogdanov-Takens surface and its intersection with the blue Hopf curve is a co-dimension 3 bifurcation.



**(a)**

**(b)**

**(c)**

**(d)**

**Figure 7.30:** Results of the SE method performed on the network from figure 7.12 and its equations, with an ensemble of identical parameters in blue and an ensemble of individual parameters in green. $f_C$ was added in this ensemble and $10^8$ samples were taken, divided over 100 intervals.

The results of the SE method are given in figure 7.30 and confirm what can also be seen from figure 7.29; compared to figure 7.20, there is no $f_I = f_C$ surface. This is due to the lack of additional co-production introduced on the stores with external stock, which was the source of the additional instability in section 7.1.4. This makes this network comparatively more stable than the network from figure 7.19. For $f_C \rightarrow -1$, the $PSW$ does not go to 0 and $f_I$ does not increase as much for $f_I \rightarrow -1$. This result is formulated in the proposition 5:

**Proposition 5** *Adding external stock supply to nodes that experience co-production interactions, heavily destabilizes the network.*

Hence, adding stock to multiple stores competing for a customer does not necessarily bring more instability, unless the stores are competing for multiple customers, which probably necessarily happens in real life networks of this kind.

## 7.3 Summary of the results for the GMs

In this chapter, we looked at several variations of a multi-commodity supply flow network. These networks consisted of customer nodes, store nodes and deliverer nodes. When initialising the generalized model, it is immediately clear that the definitions for these nodes are not necessarily precise. For instance, customers can be seen as postal zones or individual persons. This immediately demonstrates the versatility of these generalized models. As interactions are not specified precisely, they can be taken quite general.

After the initialisation, we compared the results of the bifurcation analysis and statistical ensemble method for all considered networks. Both methods yielded complementary results that helped complete the analysis of the networks. Using only the bifurcation analysis requires a lot of simplification, and still needs the use of multiple projections of the parameter space to visualize what happens if more than two parameters are varied. The SE method can only give a proportion of stable networks, which tells how stable a single parameter value is on average, but information for entire parameter sets cannot easily be extracted. The sensitivities and influences of each part were determined as well, which usually verified that central nodes and its connected parts are the most important.

This chapter extensively studied the networks containing the cyclical structures from figure 7.1. Their results manifest themselves in an understanding of how the system behaves for different parameter sets. An example of this is the cone-like bifurcation surface seen in figure 7.2. The larger networks showed similar behaviour to the single motive network, but often added new interactions and therefore bifurcation surfaces as well. The SE method results show that more complex networks contain lower $PSW$s and are hence less stable. This is generally expected, although certain added interactions may increase stability on a subset of the parameter space. This can be seen when comparing figures 7.18 and 7.21. Generally, though, decreasing $f_I$ and increasing $f_S$ increase stability. Increasing $f_\sigma$ and decreasing $f_C$ decreases stability.

Additionally, the results obtained in the other sections gave rise to five propositions, which can be enumerated as follows.
1. The general shape of bifurcation surfaces does not change, when adding interactions or nodes. Sometimes, new surfaces arise, but old ones remain qualitatively the same shape. This indicates that behaviour for smaller networks is inherited by larger networks containing the smaller network as a sub-motive.
2. It turns out that when a store node arbitrarily discriminates between two customers, the network doubles its instability in the SE.
3. When merging two sub-motives at both the store and deliverer nodes, the network seems to destabilize slightly. This indicates that when a node performs the same role in multiple sub-networks, the network is destabilized.
4. Two stores competing for customers has a larger unstable region when compared to two customers ordering

products from one store.
5. When adding the elasticity to co-production, the network is less stable, because of the added interactions. This is even clearer when stock is not assumed infinite on the store nodes, and customers essentially compete for products. However, when only finite stock is added, and no elasticity to co-production, the network is more stable, as happens with two stores and one customer. This means that the act of adding both a finite stock and co-production interactions on the same node heavily destabilize a network.

These propositions constitute the most important results from this chapter and have two functions. On the one hand, they demonstrate what kind of in results can be obtained by using GMs, and on the other hand, it presents an extensive view of the considered cyclic network motive.

# Chapter 8: Conclusions & Future research

A couple of models were investigated in this thesis; agent-based models (ABMs), (differential) equation-based models (EBMs), an optimization approach to obtain an easily integrable EBM, and generalized models (GMs). The former three have been employed for use on logistic networks in which vehicles explicitly pick up packages, after which they deliver these packages, and repeat the process. The latter model was used in a different setting, in which the spatial nature of delivering packages was not as explicitly taken into account, but rather the interactions between delivery, scheduling and order placement on its own. These two topics essentially split the thesis in two parts. Therefore, we discuss and draw conclusions for each part separately.

## 8.1 The spatial models; ABMs and EBMs

Following the structure of the thesis, first up are the ABMs and EBMs. The goal of these models was to investigate whether an EBM could be found, which describes the ABM solutions on average, and partly to examine network dynamics in a multi-commodity flow setting on different networks.

Starting with the former, a variety of networks were examined in chapter 5. The considered models always exhibited some form of start-up process, in which the network was initiated emptily, as packages had to be assigned to vehicles, before they could be able to be sent off, and both models dealt with this pretty well. Moreover, in networks with secluded nodes, this start-up process is extended somewhat by the clearer distinction of two kinds of network dynamics; local and global ones. This should be understood as an counterpart to clustering, but rather on the edge of the network. Taking for instance the network in section 5.2, shown in figure 5.1b, one can identify branches that are mostly separated from the remainder of the network, except maybe one node connecting these two. In such situations, the amount of packages on the node evolves in a significantly different way from central nodes. First, the packages are locally generated, and partly delivered from its only neighbouring node. Next, they suddenly begin to trickle in from the remainder of the network, which creates a discontinuity in the inflow, and by extension a bend in the time evolution of the package amounts on the individual nodes. This phenomenon applies to a variety of networks, assuming the network contains nodes that are somewhat secluded. Another example of this phenomenon is the $2D$-lattice graph, in which the central node is visited by nearly all paths, but the corner nodes rarely. It appears that this phenomenon is poorly captured in the EBM, and the reason for it has eluded me; the bend in the time evolutions happens at a higher point, and subsequently converges to the correct steady state. With the overwhelming evidence showing that the start-up process for secluded nodes is described in different ways by the ABM and EBM, it is clear that something is not quite right. Fortunately, the steady states are correct, but this would be the obvious thing to consider, if an eventual future attempt to improve the EBM were to be attempted. Central nodes are well approximated, though, and generally exhibit the expected behaviour, that is, that central nodes experience more traffic, partially depending on their outgoing edge length. These findings hold for both EBMs with and without congestion.

When congestion is added, though, the situation becomes decidedly more complex. The delays in delivery are edge length dependent, which are in turn time-dependent. This brings with it, a plethora of problems to be dealt with, which are treated in section 3.4. When applying the congestion EBM to a simple line network, the solutions perfectly match the ABM and expectations. However, when trying to apply this same EBM to the most discussed custom-made graph from figure 5.1a, it seems not to do its job properly. This time around, the EBM almost perfectly follows the start-up process, up until the steady state is almost reached, and by then the solution keeps diverging. Reducing the time steps seems to remedy this, but doing so is highly expensive in this case. A lot of variables need tracking and this translates to an exorbitant amount of required memory. Moreover, this EBM is costly to compute, and hence the required amount of time to integrate the EBM quickly becomes unreasonable. This could possibly be repaired in a future attempt by using the EBM for the start-up process and a Picard's fixed point method implementation for the steady state, after which these are joined together in a meaningful way at their point of intersection. This shows that the benefit of using ABMs is the easiness of extending the models and adding more complex interactions, which is harder to do for EBMs.

Overall, though, the congestionless EBMs and ABMs seem to reasonably agree with exception of the discrepancy for secluded nodes, up to a certain error due to the discrete nature of the simulations. Using Picard's fixed point method to obtain a steady state gives results that agree well with the ABMs too. The ABMs themselves could in a potential future research be expanded upon by implementing a sophisticated scheduling algorithm, and considering more realistic congestion models, such as making velocity scale inversely with vehicle density. Another avenue for exploration could be some research into the behaviour resulting from time-dependent demand, supply and travel time. Furthermore, a more concrete relation between the variance in the amount of packages and their outgoing path lengths, centralities and degrees could be investigated, along with testing these models on real world networks. One could additionally investigate variations in different parameters such as edge lengths and vehicles more extensively . It is possible to define all variables and their derivatives for the kind of ABM discussed in chapter 2. If, for instance, an ABM were considered in which the network is temporal, such definitions would be tougher to achieve. For example, if the ABM allows for edges to be added or removed, which in this case could mean present roadworks, this has to be included somehow in the EBM.

Lastly, after exploring the ABMs and EBMs, some attempts were to find an EBM that approximated the ABM well, working back from the desired solution. This was realized by taking a pretty general equation and estimating parameters with a least-squares optimization solver, in order to minimize the average difference for all nodes and all time steps. These attempts yielded better results than expected and always reached the correct steady state, whether the EBMs were constructed by comparing to the ABM solutions and to the congestionless EBM solution. When comparing this method to the ABMs with congestion, the EBM approximation overlapped tremendously well, due to the start-up process stretching out and not ending so abruptly. Hence, this way of obtaining an EBM with congestion truly trumps the method of chapter 3, as it does not display large errors whatsoever and it runs an order of magnitude quicker. The only notable flaw was that the start-up process was not described very well for the congestionless equations, as it ends so abruptly, and that cannot be captured well by continuous functions. This could in future research be improved by coming up with a cleverer way to describe the time-dependencies. Another topic to investigate could to seek a relation between the decision rules in the ABM and the parameters in the optimized EBM.

## 8.2 The generalized model

In the second topic of the thesis, in chapter 7, the generalized models were investigated in their own setting. The main interest was to investigate what kind of other information this way of modelling could yield in addition to ABMs, as ABMs necessarily assume the nature of interactions, whereas GMs keep these general. This goal shifted to describing a logistic network in another way, which does not consider the spatial aspect of package delivery, and hence essentially looks at longer time scales, in which packages are or aren't available for delivery, due to supply and other such interactions. This was done by considering three kinds of nodes, a customer (or a postal zone), a store, and a delivery company (or distribution center, if interpreted more locally), with an additional wild card node which deals with all external interactions, like social influence, stock and consumption of wares, called the reservoir. These have then been cyclically connected such that the consumer receives feedback in the shape of arriving products. These three kinds of nodes have then been arranged in two physically sensible ways; one in which two stores compete for a customer and one in which two customers need products from one store. These situations have been expanded to include increasingly complicated interactions, which yield different bifurcation diagrams and statistical ensemble results. Additionally, in each situation, the influence and sensitivities were computed for all parts and nodes in the network, which gave more insight into the relative importance and dependence of each part.

Starting with two customers and one store, there are three elasticities; the elasticity to supply, inventory level and saturation. It appears that increasing the elasticity to saturation increases the instability region, whereas increasing the elasticity to supply stabilizes the network, and decreasing the elasticity to inventory level also stabilizes the network. For more complex networks, an additional final elasticity is introduced; the elasticity to co-production. Decreasing this elasticity generally destabilizes the network significantly.

In brief, both the bifurcation analysis and the statistical ensemble method show that increased network complexity and heterogeneity of the interactions decrease stability substantially. This is mostly evident when assessing the results of the statistical ensemble method with invididual samples. In the later sections of chapter 7, the $PSW$s begin to lie close to zero, and often around or close to $0.2$ for a sizeable part of the parameter space. Not only does it increase the instability, but the bifurcation surfaces become more complicated and contain more interesting higher co-dimension bifurcations as well.

It has been demonstrated that the bifurcation and statistical ensemble methods complement each other. The bifurcation analysis focusses more on the shape of the stability region, which is a more thorough breakdown of the problem. It however heavily relies on oversimplification of the problem. The statistical ensemble allows for this simplification to be omitted, but loses a lot of information about specific locations of stability regions and only gives a proportion of stability. These proportions for identical ensembles mirror the bifurcation analysis, but their results may deviate from the latter, when looking at an individual ensemble. The sensitivity and influence of parts and nodes in the network were determined along with the SE method. These portray the relative importance of parts and nodes in the network. Using both methods together is the key to the most complete picture of the problem.

Using the described model and solutions, a couple of propositions were made. These are given below, with an eventual explanation.

**Proposition 1** *Stability behaviour for the small cyclic network is inherited by larger networks containing the small network as a sub-motive.*

**Proposition 2** *Arbitrarily treating different customers differently, significantly decreases the overall stability of the networks in multi-customer feedback flow networks.*

This means, more concretely, that if customer orders are processed differently by the store, without any careful way to chose this, the network easily becomes much more unstable.

**Proposition 3** *Having one node in a network perform the same role in multiple sub-networks decreases network stability.*

In other words, splitting responsibilities and tasks, and distributing them more locally improves the overall stability of the network. In this case, if one distribution centre has to deliver products to two postal zones, the bifurcation diagram shows a small decrease in stability.

**Proposition 4** *Two stores competing for customers makes a network less stable than two customers ordering the products from one store*

**Proposition 5** *Adding external stock supply to nodes that experience co-production interactions, heavily destabilizes the network.*

This instability manifest itself in partitioning the parameter space in an unstable and stable part by a simple line, in the simplification made for the bifurcation analysis.

Using the two analysis methods, the stability of the system is then well described and yields non-trivial interesting results for the considered networks. These results apply to rather small motives, and can hence be expected to extrapolate in some way to larger networks containing these same smaller sub-motives. It is however also clear when these two methods are lacking. As mentioned before, the bifurcation method is hardly usable for networks with a lot of parameters, and the statistical ensemble method does not present information that can be used to infer a lot for particular parameter sets on its own.

Also, for generalized models, many areas are not yet explored. An interesting question would be to investigate the effect of heterogeneity in the parameters on the system's stability. For instance, one could explore whether splitting the elasticity to co-production on one node into two separate elasticities significantly changes the bifurcation surfaces. Another option would be to consider networks of increasing size and verifying the extent to which results for smaller networks can be extrapolated.

# Appendix A: Results for the first EBM attempt

Because the results obtained with the method described in section 3.1 did not come out completely as desired, they are presented in this separate appendix.

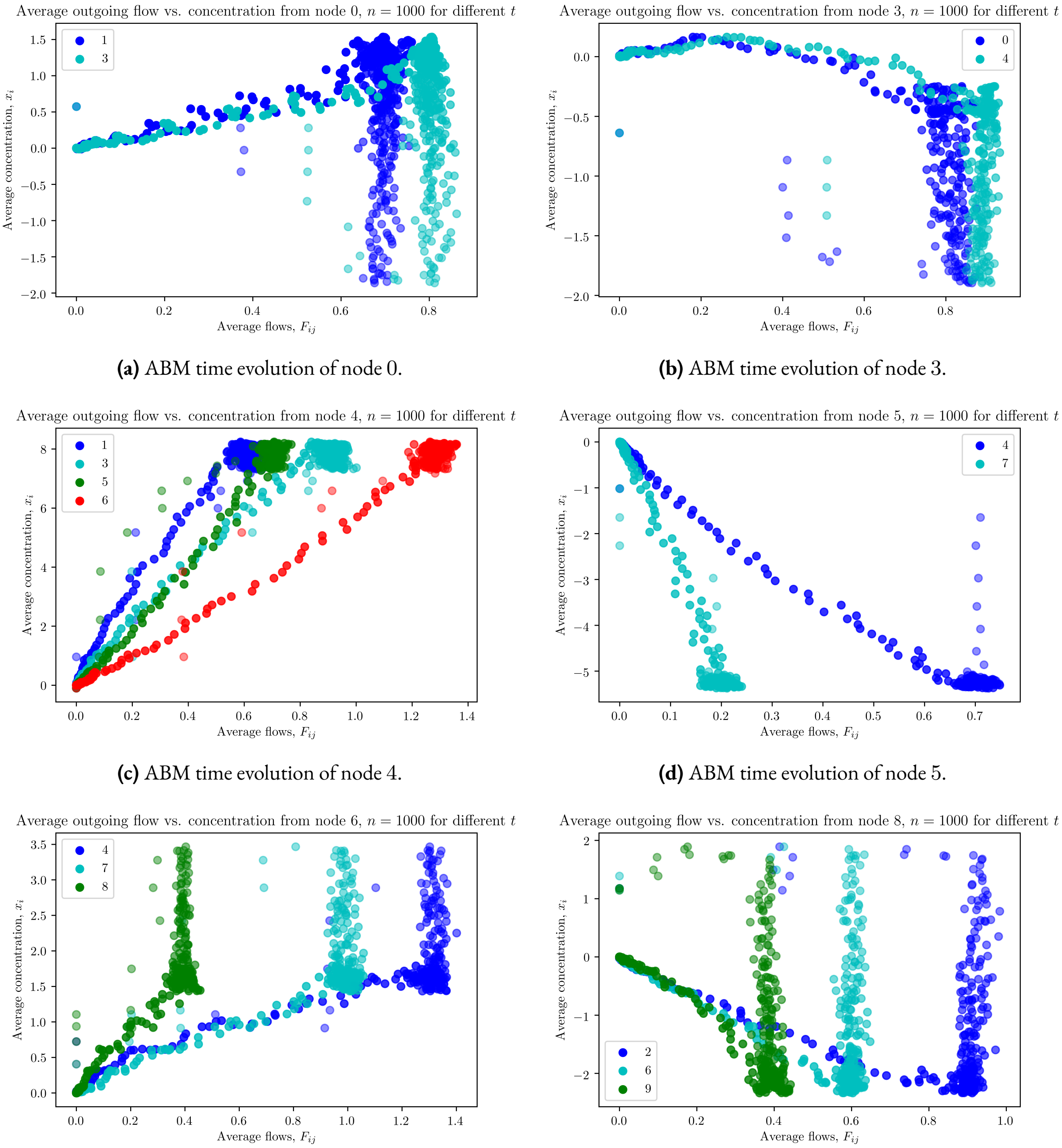
The approach turned out to generally not work, partially depending on the shapes of $u_i$ and $w_i$, because the flows did not seem to be linearly dependent on $x$ in general.
In the early stage in which this model was under consideration, the ABM was not being run under the same assumptions as now. For starters, the supply was fixed and decided uniformly at the start of the simulations. The demand on the following time steps was balanced such that this supply was perfectly countered. This construction is hence contrary to the approach in chapter 2, where the supply was perfectly balanced by the demand, and basically assumed an external reservoir. Also, in this ABM, consumption was not yet properly implemented, but this was rather approached by assuming demand as a negative amount of packages. Lastly, the amount of trucks was not superfluous, such that the delivery of packages was throttled, and the amount of leaving trucks could be throttled each time step. Except for the addition of congestion, the remainder of the ABM is left pretty much the same.

Typical behaviour for such simulations on the custom-made network from figure 5.1a, is depicted in figure A.1. These simulations have been run $1000$ times in this figure. Moreover, it is clear that all nodes perform differently under the same decision rules, due to their different centralities and connected edge lengths. For instance, nodes $4$ and $6$ show pretty varied and non-linear behaviour, due to them being pretty central in the network. Similarly, node $5$ shows little spread and is fairly well-behaved as not a lot happens on that node, due to it being so secluded. The remainder of the nodes are chosen to show and represent the more mundane behaviours that can occur on nodes that are neither central, nor remote.

All figures start at zero , and quickly jump over discontinuously to another point, which happens due to the time discretization. Soon after, they tend to jump over again to some other value for $x$, which indicates that more flow is happening on the outgoing node, and hence to more time having passed. When a jump in $F_{ij}$ occurs, trucks basically have had the time to traverse their initial edge(s) and move on to the next edge in their path.

Depending on the node, the blots indicate a steady state, such as in A.1c, A.1d and A.1e, or the lack of blots indicate that the amount of packages generated in the network is not enough, or barely enough, for a steady state to occur for $x$. However, a steady state appears to occur consistently for $F$. Afterwards, the steady state drains away to zero again, as the packages have all been delivered. This seems to happen in a more continuous fashion, due to the randomness in the total generated demand, and the large amount of states the system can be in after a larger amount of time steps.

Figure A.2 shows what would happen to the central node $4$ and more remote node $0$ in the case when there are too many packages to deliver in the available amount of time steps. This situation gives a clearer picture of how the steady states are formed. For node $0$ there's a fairly large tail which originates from the start-up process, where not all flows are present yet and the majority of paths still have to be passed. For node $4$, as it is so central, most paths passing through it already arrive quite early, such that the tail does not form.



**(a)** ABM time evolution of node 0.

**(b)** ABM time evolution of node 3.

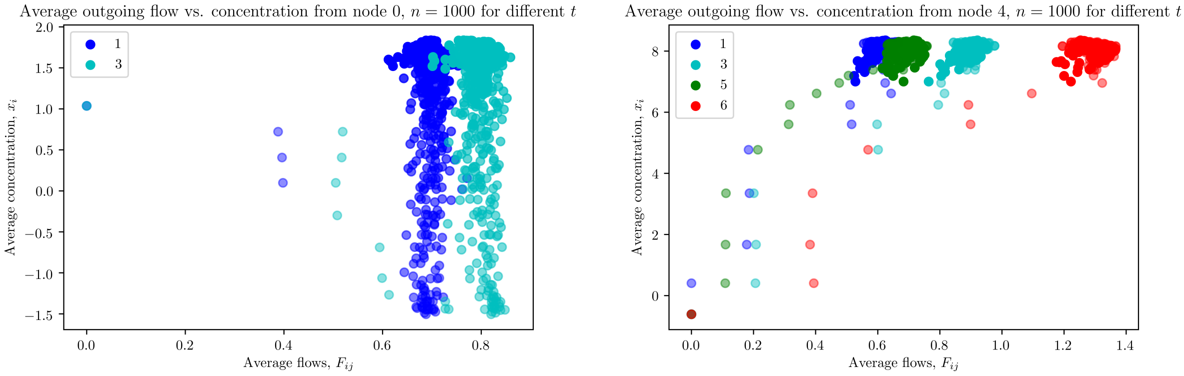
**(c)** ABM time evolution of node 4.

**(d)** ABM time evolution of node 5.

**(e)** ABM time evolution of node 6.

**(f)** ABM time evolution of node 8.

**Figure A.1:** Time evolutions of the package amount pertaining to a respective node from the network from figure 5.1a, plotted against the outgoing flows. Dots with higher opacity signify that these states occur earlier on in the simulations. $n$ denotes the amount of simulations, which is 1000 here. Additionally, the averageamount of packages generated per node is 240, and the amount of time steps is 576. Only one vehicle can leave each time step.



**(a)** ABM time evolution of node 0.

**(b)** ABM time evolution of node 4.

**Figure A.2:** Simulations similar to the ones performed in figure A.1, but now with an abundance of packages (600).

After doing this, and discovering that the relation between $F_{ij}$ and $x_i$ could not be exploited, this particular model was abandoned and the research moved on to the current form of the ABM treated in chapter 2.

# Appendix B: Derivation of the general solution to $N$-dimensional differential equations

Let $p_p = e^{At}q(t)$ be the particular solution, and $s - c = b(t)$, such that

$$\begin{aligned}\dot{p}_p = Ap + b &\implies Ae^{At}q(t) + e^{At}\dot{q}(t) = Ae^{At}q(t) + b(t)\\ &\implies e^{At}\dot{q}(t) = b(t) \implies q(t) = q_0 + \int e^{-At}b(t)dt\\ &\implies p(t) = e^{At}\left(p_0 + \int e^{-At}b(t)dt\right).\end{aligned} \tag{65}$$

This can be integrated numerically in the following fashion:

$$\begin{aligned}p(t+\delta t) &= e^{A(t+\delta t)}p_0 + \int_0^{t+\delta t} e^{A(t+\delta t - t')}b(t')dt'\\ &= e^{A\delta t}\left(e^{At}p_0 + \int_0^t e^{A(t-t')}b(t')dt'\right) + \int_t^{t+\delta t} e^{A(t+\delta t-t')}b(t')dt'\\ &= e^{A\delta t}p(t) + \int_t^{t+\delta t} e^{A(t+\delta t-t')}b(t')dt'\\ &\approx e^{A\delta t}p(t) + \delta t\frac{(b(t+\delta t) + e^{A\delta t}b(t))}{2} \approx e^{A\delta t}p(t) + \delta t\frac{(b(t+\delta t)+b(t))}{2},\end{aligned} \tag{66}$$

making use of the trapezium approximation up to order $\delta t$.

If, however, in addition $A = A(t)$, then we need to consider the time-ordered exponential: $\mathscr{T}\left(e^{\int_0^t A(t)dt}\right)$, which is defined as

$$\mathscr{T}\left(e^{\int_0^t A(t)dt}\right) = \lim_{n\to\infty}\prod_{i=1}^{n} e^{A(t_i)\Delta t} = \lim_{n\to\infty} e^{A(t_n)\Delta t}\cdot e^{A(t_{n-1})\Delta t}\dots,\qquad \Delta t = \frac{t}{n}, \tag{67}$$

having taken the left product.

To find the inverse of the time-ordered exponential, one has to look for a function such that

$$\lim_{n\to\infty} e^{A(t_n)\Delta t}\cdot e^{A(t_{n-1})\Delta t}\dots\left[\mathscr{T}(e^{\int_0^t A(t)dt})\right]^{-1} = I. \tag{68}$$

As $e^{A(t_i)\Delta t}$ obviously commutes with itself,

$$e^{A(t_i)\Delta t}e^{-A(t_i)\Delta t} = e^{A(t_i)\Delta t - A(t_i)\Delta t} = e^0 = I.$$

Necessarily, thence:

$$\mathscr{T}^{-1}(e^{\int_0^t A(t)dt}) \equiv \left[\mathscr{T}(e^{\int_0^t A(t)dt})\right]^{-1} = \lim_{n\to\infty}\prod_{i=n}^{0} e^{-A(t_i)\Delta t}, \tag{69}$$

in which the left product has been taken, as before. Hence, in a similar way;

$$p(t) = \mathscr{T}\left(e^{\int_0^t A(t)dt}\right)\left(p_0 + \int \mathscr{T}^{-1}\left(e^{\int_0^t A(t)dt}\right) b(t)dt\right), \tag{70}$$

which constitutes the solution of the inhomogeneous time-dependent problem $\dot{p} = A(t)p + b(t)$.

# Appendix C: Derivation of the matrix exponential derivative

When using the scipy.optimize package in Python, the solvers appreciate an analytical Jacobian. To that end, an attempt was made to produce said Jacobian. To be able to do so, first of all, it is necessary to be able to create $\frac{d}{d\alpha_{ij}}e^{At}$. Let $d\alpha_{ij}$ denote the matrix $\delta\alpha_{ij}E_{ij}$, which is a matrix with a small increment on entry $i, j$.

$$\frac{d}{d\alpha_{ij}}e^{At} = \lim_{\delta\alpha_{ij}\to 0} \frac{e^{A+\delta\alpha_{ij}E_{ij}} - e^{At}}{\delta\alpha_{ij}} \tag{71}$$

The former exponential in the numerator can be expanded using the Baker-Campbell-Hausdorff's equation:

$$\begin{aligned} e^{(A+d\alpha_{ij})t} &= e^{At}e^{d\alpha_{ij}t}e^{-\frac{t^2}{2}[A,d\alpha_{ij}]}e^{\frac{t^3}{6}(2[d\alpha_{ij},[A,d\alpha_{ij}]]+[A,[A,d\alpha_{ij}]])} \\ &\cdot\ e^{-\frac{t^4}{24}([[[A,d\alpha_{ij}],A],A]+3[[[A,d\alpha_{ij}],A],d\alpha_{ij}]+3[[[A,d\alpha_{ij}],d\alpha_{ij}],d\alpha_{ij}]+\dots)}, \end{aligned} \tag{72}$$

which is of no help, as the exponents still contain infinitely many order $\delta\alpha_{ij}$ terms, and hence we would be stuck with an infinite product anyway.

Working out the exponent from its definition yields

$$\begin{aligned} e^{(A+d\alpha_{ij})t} &= \sum_{n=0}^{\infty}\frac{(A+d\alpha_{ij})^n t^n}{n!} = I + (A+d\alpha_{ij})t + \frac{(A+d\alpha_{ij})^2t^2}{2} + \dots \\ \implies \frac{d}{d\alpha_{ij}}e^{At} &= \lim_{\delta\alpha_{ij}\to 0}\frac{\sum_{n=0}^{\infty}\left(\frac{(A+d\alpha_{ij})^n t^n}{n!} - \frac{A^n t^n}{n!}\right)}{\delta\alpha_{ij}} \\ \implies \frac{d}{d\alpha_{ij}}e^{At} &= E_{ij} + \frac{AE_{ij}+E_{ij}A}{2} + \frac{A^2E_{ij}+AE_{ij}A+E_{ij}A^2}{6} + \dots, \end{aligned} \tag{73}$$

which is still not a closed expression, but it is sure to converge due to the $n!$ in the denominator. However, this cannot readily be provided to python.

An efficient poor man's solution to numerically approximate the matrix exponential derivative is given by Higham's Complex Step Approximation 16:

$$\frac{d}{d\alpha_{ij}}e^{At} = \lim_{\delta\alpha_{ij}\to 0} Im\left(\frac{e^{A+i\delta\alpha_{ij}E_{ij}}}{\delta\alpha_{ij}}\right), \tag{74}$$

which can be practically used for small $\delta\alpha_{ij}$, say $\delta\alpha_{ij} = 10^{-20}$, to yield a very accurate answer, with relatively low cost. This method filters out the derivative of the exponential up to order $h^3$, as the zeroth and second order term in the Taylor expansion are real.

# Appendix D: Sensitivity and Influence

This appendixx contains the tables with sensitivities and influences for each network considered in chapter 7, except the network with only one motive, as that table has been treated in in chapter 7.

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.850 | 4.506 | 4.549 | 4.168 | 3.499 | 3.850 | 4.506 | 4.549 | 4.168 | 3.499 |
| $In_i$ | 3.459 | 4.162 | 4.312 | 4.700 | 3.723 | 3.459 | 4.162 | 4.312 | 4.700 | 3.723 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 16.023 | 16.023 | 18.109 | 8.717 | 8.717 |
| $In$ | 16.045 | 16.045 | 16.949 | 9.012 | 9.012 |

**(a)**

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 2.723 | 4.366 | 4.059 | 3.541 | 2.544 | 2.707 | 4.365 | 4.062 | 3.539 | 2.550 |
| $In_i$ | 2.856 | 3.611 | 4.414 | 4.457 | 3.815 | 2.862 | 3.608 | 4.412 | 4.460 | 3.826 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 13.173 | 13.161 | 16.852 | 7.600 | 7.602 |
| $In$ | 14.740 | 14.755 | 16.045 | 8.871 | 8.872 |

**(b)**

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.632 | 4.339 | 4.437 | 3.841 | 3.035 | 3.616 | 4.328 | 4.434 | 3.841 | 3.027 |
| $In_i$ | 3.410 | 4.179 | 4.451 | 4.504 | 3.722 | 3.403 | 4.170 | 4.437 | 4.504 | 3.721 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 14.847 | 14.812 | 16.852 | 8.278 | 8.275 |
| $In$ | 15.815 | 15.799 | 17.238 | 8.956 | 8.941 |

**(c)**

**Table D.1:** The Sensitivities and Influences of each part and each node in the two motive network from figure 7.8, when sampling identically (top), individually for each part (middle), and individually for all elasticities (bottom). These have been determined with $10^6$ samples.

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.939 | 4.583 | 4.492 | 4.320 | 3.648 | 3.939 | 4.583 | 4.492 | 4.320 | 3.648 |
| $In_i$ | 3.596 | 4.303 | 4.649 | 4.698 | 3.722 | 3.596 | 4.303 | 4.649 | 4.698 | 3.722 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 16.489 | 16.489 | 18.149 | 8.812 | 8.712 |
| $In$ | 16.319 | 16.319 | 17.904 | 9.347 | 9.347 |

(a)

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.779 | 4.335 | 4.603 | 4.090 | 3.319 | 3.798 | 4.337 | 4.593 | 4.096 | 3.315 |
| $In_i$ | 3.472 | 4.256 | 4.569 | 4.609 | 3.767 | 3.493 | 4.257 | 4.570 | 4.608 | 3.764 |

| | $C_1$ | $C_2$ | $S$ | $D$ |
|---|---|---|---|---|
| $Se$ | 15.523 | 15.545 | 17.868 | 17.382 |
| $In$ | 16.104 | 16.123 | 17.651 | 18.356 |

(b)

**Table D.2:** The Sensitivities and Influences of each part and each node in the two motive network, with the same deliverer, like shown in figure 7.12, when sampling identically (top) and individually (bottom). These have been determined with $10^6$ samples.

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.844 | 4.369 | 4.727 | 4.035 | 3.175 | 3.844 | 4.369 | 4.727 | 4.035 | 3.175 |
| $In_i$ | 3.649 | 4.351 | 4.642 | 4.824 | 3.978 | 3.649 | 4.351 | 4.642 | 4.824 | 3.978 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 15.523 | 15.423 | 18.191 | 8.762 | 8.762 |
| $In$ | 16.802 | 16.802 | 17.985 | 9.466 | 9.466 |

(a)

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.752 | 4.309 | 4.577 | 4.056 | 3.270 | 3.774 | 4.304 | 4.574 | 4.067 | 3.301 |
| $In_i$ | 3.458 | 4.230 | 4.531 | 4.585 | 3.739 | 3.471 | 4.239 | 4.541 | 4.585 | 3.747 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 15.387 | 15.445 | 17.764 | 8.633 | 8.641 |
| $In$ | 16.012 | 16.042 | 17.541 | 9.116 | 9.126 |

(b)

**Table D.3:** The Sensitivities and Influences of each part and each node in the two motive network, with the same deliverer, like shown in figure 7.12, with an additional introduced $f_C$, when sampling identically (top) and individually (bottom). These have been determined with $10^6$ samples.

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ | $P_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.701 | 4.449 | 4.675 | 3.859 | 2.962 | 3.701 | 4.449 | 4.675 | 3.859 | 2.962 | 3.809 |
| $In_i$ | 3.516 | 4.218 | 4.678 | 4.837 | 3.997 | 3.516 | 4.218 | 4.678 | 4.837 | 3.997 | 2.659 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 14.971 | 14.971 | 22.056 | 8.534 | 8.534 |
| $In$ | 16.569 | 16.569 | 20.452 | 9.516 | 9.516 |

**(a)**

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ | $P_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.739 | 4.336 | 4.568 | 4.015 | 3.217 | 3.775 | 4.264 | 4.627 | 4.004 | 3.209 | 3.930 |
| $In_i$ | 3.313 | 4.078 | 4.555 | 4.566 | 3.728 | 3.531 | 4.294 | 4.623 | 4.669 | 3.824 | 3.304 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 15.308 | 15.251 | 21.724 | 8.583 | 8.631 |
| $In$ | 15.685 | 16.317 | 20.853 | 9.121 | 9.291 |

**(b)**

**Table D.4:** The Sensitivities and Influences of each part and each node in the two motive network, with the same deliverer and a reservoir connected to $S$, like shown in figure 7.19, when sampling identically. These have been determined with $10^6$ samples. In this ensemble, the omitted samples that yielded complex logarithms were approximately 11000 of the 18000 for the identical ensemble and 19000 of the 22000 for the individual ensemble.

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ |
|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 4.013 | 4.652 | 4.411 | 4.068 | 4.652 | 4.411 | 4.068 | 3.437 | 3.437 |
| $In_i$ | 3.341 | 3.968 | 4.487 | 4.823 | 3.968 | 4.487 | 4.823 | 3.747 | 3.747 |

| | $C$ | $S_1$ | $S_2$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 28.328 | 9.063 | 9.063 | 8.479 | 8.479 |
| $In$ | 28.417 | 8.455 | 8.455 | 9.310 | 9.310 |

**(a)**

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ |
|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.307 | 4.588 | 4.251 | 3.639 | 4.600 | 4.262 | 3.640 | 2.864 | 2.867 |
| $In_i$ | 3.292 | 4.095 | 4.451 | 4.774 | 4.091 | 4.451 | 4.774 | 3.914 | 3.923 |

| | $C_1$ | $C_2$ | $S$ | $D_1$ | $D_2$ |
|---|---|---|---|---|---|
| $Se$ | 25.505 | 8.839 | 7.890 | 8.862 | 7.901 |
| $In$ | 28.863 | 8.546 | 9.225 | 8.542 | 9.225 |

**(b)**

**Table D.5:** The Sensitivities and Influences of each part and each node in the two store network, as shown in figure 7.22, when sampling identically (top) and individually (bottom). These have been determined with $10^6$ samples.

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ |
|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 4.051 | 4.697 | 4.377 | 3.966 | 4.697 | 4.377 | 3.966 | 3.142 | 3.142 |
| $In_i$ | 3.443 | 4.056 | 4.667 | 4.906 | 4.056 | 4.667 | 4.906 | 3.956 | 3.956 |

| | $C$ | $S_1$ | $S_2$ | $D$ |
|---|---|---|---|---|
| $Se$ | 27.661 | 9.074 | 9.074 | 16.686 |
| $In$ | 29.280 | 8.723 | 8.723 | 19.147 |

(a)

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ |
|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.851 | 4.723 | 4.488 | 4.001 | 4.751 | 4.522 | 4.061 | 3.248 | 3.312 |
| $In_i$ | 3.284 | 4.209 | 4.614 | 4.847 | 4.306 | 4.655 | 4.868 | 3.944 | 3.952 |

| | $C$ | $S_1$ | $S_2$ | $D$ |
|---|---|---|---|---|
| $Se$ | 27.946 | 9.211 | 9.273 | 17.072 |
| $In$ | 29.410 | 8.823 | 8.961 | 18.984 |

(b)

**Table D.6:** The Sensitivities and Influences of each part and each node in the two store network, with the same deliverer, as shown in figure 7.25, when sampling identically (top) and individually (bottom). These have been determined with $10^6$ samples.

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ | $P_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.731 | 4.874 | 4.179 | 3.708 | 4.874 | 4.179 | 3.708 | 2.854 | 2.854 | 3.640 | 3.640 |
| $In_i$ | 3.090 | 3.668 | 4.715 | 4.888 | 3.668 | 4.715 | 4.888 | 3.965 | 3.965 | 3.411 | 3.411 |

| | $C$ | $S_1$ | $S_2$ | $D$ |
|---|---|---|---|---|
| $Se$ | 26.603 | 12.692 | 12.692 | 15.774 |
| $In$ | 28.132 | 11.794 | 11.794 | 19.207 |

(a)

| | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ | $P_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Se_i$ | 3.699 | 4.699 | 4.390 | 3.780 | 4.754 | 4.423 | 3.833 | 3.008 | 3.059 | 3.407 | 3.442 |
| $In_i$ | 2.937 | 3.879 | 4.560 | 4.720 | 4.022 | 4.635 | 4.769 | 3.854 | 3.891 | 3.341 | 3.303 |

| | $C$ | $S_1$ | $S_2$ | $D$ |
|---|---|---|---|---|
| $Se$ | 26.832 | 12.496 | 12.619 | 16.427 |
| $In$ | 28.073 | 11.780 | 11.960 | 18.684 |

(b)

**Table D.7:** The Sensitivities and Influences of each part and each node in the two store network, with the same deliverer and reservoirs connected to the stores, as shown in figure 7.28, when sampling identically (top) and individually (bottom). These have been determined with $10^6$ samples. In the identical ensemble, about 12750 of the 20500 samples were omitted.

# References

[1] G. Demirel, B.L. MacCarthy, D. Ritterskamp, A.R. Champneys & T. Gross. Identifying Dynamical Instabilities in Supply Networks Using Generalized Modeling. Wiley, Journal of Operations Management, vol 65 (2), Pages 136–159. 2019.

[2] D. Ritterskamp, G. Demirel, B.L. MacCarthy, L. Rudolf, A.R. Champneys & T. Gross. Revealing instabilities in a generalized triadic supply network: A bifurcation analysis. Chaos 28, 073103. 2018.

[3] T. Gross & U. Feudel. Analytical search for bifurcation surfaces in parameter space. Physica D 195, 292,302. 2004.

[4] H. Aufderheide, L. Rudolf, T. Gross & K.D. Lafferty. How to predict community responses to perturbations in the face of imperfect knowledge and network complexity. Proceedings of the Royal Society B, Volume 280, Issue 1773. 2013.

[5] B. Werner. Computation of Hopf Bifurcation with Bordered Matrices. SIAM J. Numer. Anal. Vol. 33, No. 2, pp. 435-455. 1996.

[6] J.Y. Yen, Finding the k Shortest Loopless Paths in a Network. Management Science. vol 17 (11), Pages 712–716. 1971.

[7] S. Floyd, M. Handley, J. Padhye & J. Widmer. Equation-based congestion control for unicast applications. SIGCOMM '00 Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Pages 43-56. 2000.

[8] R. Ahlswede, N. Cai, S.-Y.R. Li & R.W. Yeung. Network information flow. IEEE Transactions on Information Theory. vol. 46 (4): 1204 - 1216. 2000.

[9] G. Nilsson, G. Como, & E. Lovisari. On Resilience of Multicommodity Dynamical Flow Networks. 53rd IEEE Conference on Decision and Control. 2014.

[10] G. Como. On resilient control of dynamical flow networks. Annual Reviews in Control, vol. 43, Pages 80-90. 2017.

[11] L. Zhao, Y.-C. Lai, K Park & N. Ye. Onset of traffic congestion in complex networks. Phyiscal Review E, vol. 71, issue 2, 026125. 2005.

[12] H. Rahmandad & J. Sterman. Heterogeneity and Network Structure in the Dynamics of Contagion: Comparing Agent-Based and Differential Equation Models. Management Science. vol. 54 (5), Pages 998-1014. 2008.

[13] H. Van Dyke Parunak, R Savit, and R.L. Riolo. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. Proceedings of Multi-agent systems and Agent-based Simulation (MABS'98), 10-25, Springer, LNAI 1534. 1998.

[14] A. Namatame & S.-H. Chen. Agent-Based Modeling and Network Dynamics. Oxford University Press: Oxford. 2016.

[15] G. Cencetti, F. Bagnoli, F. Di Patti & D. Fanelli. The second will be first: competition on directed networks. Scientific Reports 6, Article number 27116. 2016.

[16] A.H. Al-Mohy & N.J. Higham. The Complex Step Approximation to the Fréchet Derivative of a Matrix Function. The University of Manchester. 2009.

[17] Y.A. Kuznetsov. Elements of Applied Bifurcation Theory, Chapter 6. Applied Mathematical Sciences 112, Springer. 2004.

[18] A. Dhooge, W. Govaerts, Y.A. Kuznetsov, H.G.E. Meijer & B. Sautois. New features of the software MatCont for bifurcation analysis of dynamical systems. MCMDS 2008, Vol. 14, No. 2, pp 147-175. 2008.

[19] E.J. Doedel. AUTO-07P : Continuation and Bifurcation Software for Ordinary Differential Equations. Concordia University, Montreal, Canada. 2007.

[20] Maple 2017. Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario.

[21] Matlab R2019a. Natick, Massachusetts: The MathWorks Inc.

[22] E. Picard. Sur l'application des méthodes d'approximations succesives à l'étude de certaines équations différentielles ordinaires. Journal de Mathematiques, p.217. 1893.

[23] C.A. Hermans, A. Stam & J.L.A Dubbeldam. A comparison between agent-based and equation-based model approaches for multi-commodity traffc ow in logistic networks. IEEE Conference on Decision and Control. Submitted, 2019.