Master of Science Thesis

Designing Express Networks with Multi-Agent Modelling

The Use of Multi-Agent Modelling to Solve the Hub Location Problem in Express Networks

T.G. Wijnsma BSc

19 December 2011

Faculty of Technology, Policy and Management - Delft University of Technology

Delft University of Technology

Designing Express Networks with Multi-Agent Modelling

The Use of Multi-Agent Modelling to Solve the Hub Location Problem in Express Networks

Master of Science Thesis

For obtaining the degree of Master of Science in Systems Engineering, Policy Analysis and Management at Delft University of Technology

T.G. Wijnsma BSc

19 December 2011

Faculty of Technology, Policy and Management - Delft University of Technology

Delft University of Technology

Delft University Of Technology
Department Of
Systems Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Technology, Policy and Management for acceptance a thesis entitled **"Designing Express Networks with Multi-Agent Modelling"** by **T.G. Wijnsma BSc** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 19 December 2011

Head of department: ______________________
Prof.dr. F.M.T. Brazier

Supervisor: ______________________
Dr. M.E. Warnier

Reader: ______________________
drs. J.H.R. van Duin

External supervisior: ______________________
W.J.M. Meuffels MSc

Management summary

The ORTEC Consulting Group (OCG) advises express service providers on efficient network design. A part of the challenge to this design is the hub location problem (HLP). The HLP consists of selecting optimal locations for large sorting centres (hubs) from a range of smaller sorting centres (depots) to which the rest of these depots can connect. The resulting network aims to transport parcels in the most cost effective manner achievable. The OCG is interested in multi-agent modelling technique as a new approach to the HLP, since this technique provides a natural way of modelling complexity that arises from interacting nodes. Literature showed no research on solving the HLP from a multi-agent systems (MAS) perspective. This thesis aims to fill this gap.

With the main research goal being *Designing a model that can solve the hub location problem for express networks with use of multi-agent modelling*, the model was designed using the Prometheus methodology. The model uses information on the volume distributions of a set of fixed depots to determine optimal hub locations from these depots. Simultaneously, the model ensures that parcels can be sent from any location and will be delivered within the set service time. Optimal hub locations in this sense means that the total network cost consisting of hub cost and transport cost is as low as possible. The inputs of the designed model are the depot locations, their volume distributions, the driving times between nodes, hub cost and transport cost. The outputs are the network cost, the number of hubs, the locations of these hubs and the routes of all the different parcels. The designed MAS consists of three main phases. Phase 1 is responsible for creating hubs based on volume distributions. Hubs are placed in regions that have a lot of parcels to be transported between them. Phase 2 creates routes via the hubs that resulted from Phase 1. Although the most efficient routes are calculated, the main focus of this phase is to create routes for every parcel in the first place. During Phase 3 the main focus is cost reduction through reducing air transport cost, reducing road transport cost and reducing hub cost.

In addition, part of this design is implemented into a proof of concept using the JACK Agent Language (a Java based language) to show the added value of multi-agent modelling. This proof of concept, Preliminary Organisation of Hub Location

Tool (POHST), contains the implementation of the first part of Phase 1. Hence, it creates hubs based on the volume distribution. A graphical user interface is added to turn POHST into an easily accessible tool. This tool is applied to two datasets to demonstrate its use in the preliminary phase of network research aiding in data gathering and the generation of initial hub configurations. Experts of the OCG confirmed the usefulness of POHST as strategic analysis tool. In addition, the type of model outputs did not allow for thorough integral testing of the tool, although other verification methods showed that the tool behaved as intended by the model design. Due to the lack of quantitative output, the tool could not be validated using traditional methods. Instead, the same experts were asked to validate the model. Although they had quite a few recommendations for further improvement of the tool, there was a consensus on its validity.

The design process revealed interesting benefits and drawbacks of using the agent paradigm to solve the HLP. The source of much of the complexity of the HLP is the interaction between nodes. One of the major advantages of using the agent paradigm is that it provides a natural way of modelling such interactions. Furthermore, the scalability of agent models is an attractive feature. When a few types of agents have been designed, an unlimited amount of such agents can be used when applying the model, scaling along with the inserted data. Next, the research shows the benefit of agents adapting to local circumstances. By locally looking around for inefficiencies agents are capable to enhance the solution with limited data. Another advantage of using agents is the detailed level of statistics gathering it enables. Every agent decision can be tracked. Consequently, an agent model does not only produce an outcome, it can also show how and why this outcome evolved as it did.

The major challenge of using agents to solve the HLP is the difficulty of making local decisions that might impact the entire infrastructure. To be absolutely sure that a local change will lead to lower network cost, all possible consequences are checked and valued for their change in cost. This process can lead to a vast amount of communication, because the agents are practically considering global data. Thus, undermining the multi-agent modelling values of local data views. Although this risk exists in the presented model, it does not prove that it is impossible to achieve a design that reaches a global optimum using strictly local information. However, it is the greatest challenge that the HLP poses to the use of multi-agent modelling.

In conclusion, it can be said that this research has proved the potential of using MAS to solve the HLP. It is also certain that many interesting properties of agents can be further investigated. Subsequently, these models could add great value to compete with and possibly even defeat current HLP models.

Acknowledgements

This research would not have been possible without the support and advice of a number of people whom I would like to thank. First of all, I would like thank Martijn Warnier, supervisor, for his advice on my research. He played an important role in guarding the quality of the report in terms of both structure and textual clarity. I enjoyed the freedom he gave me, on one hand, to approach the research the way I thought was best (e.g. delving deep into the design of the model). On the other hand, he helped me refocus on the report as it is the most important deliverable of this research, while the model forms only part of it. Next, I would like to thank Frances Brazier, professor in my committee, for her guidance especially at the start of my research. She helped me determine a good agent definition and guarded the academic level of my research. Also, my thanks go out to Ron van Duin, reader, for his support especially during the later stages of my research. His enthusiasm about the subject was inspiring and I would like to thank him for honouring the practicality of the tool that resulted as a side product of the research.

Special thanks go out to two great women that played a major part in this research. First of all, Ineke Meuffels, my external and daily supervisor, whom I thank her for her great support, intensive supervision and her comments during the design phase as well as on the report and the project planning. She was a great motivator, has extensive knowledge on the subject and brought many, many laughs during my research. Secondly, I would like to thank Iris, my girlfriend, who was invaluable for the success of the project. She provided me with moral support and was a great motivator. She helped with checking spelling and grammar, altering the structure of text, and was always available for discussion on research related topics. Furthermore, I would like to thank her for not minding my programming in the evening at home.

My gratitude goes out to all involved people at ORTEC who made it possible for me to perform this research. My colleagues at the OCG-TNT department for their technical and moral support in the past months. I had a great time. My colleagues at the ORTEC SPI department, where I continued to work part-time one day a week, for being flexible with my workload while I was working on my thesis four days a week.

Of course, a big thank you goes to my parents, for their invaluable support not only during the time of this research, but throughout my entire curriculum. Thanks for being there when times were difficult and for always believing in me.

I would like to thank Aart and Angelien for providing me with perspective, motivation and determination when necessary. I would like to thank Igor for sharing his experience with LaTeX, Java and general issues most graduates encounter during their final thesis. And finally thanks to all other people that in one way or another helped me during this research.

> *Not everything that can be counted counts, and not everything that counts can be counted.* - Albert Einstein (1879-1955)

Delft, The Netherlands T.G. Wijnsma BSc
19 December 2011

Contents

List of Figures

Chapter 1

Introduction

This thesis focuses on designing a model for express service providers to increase the efficiency of their express networks. More specifically, through the use of the multi-agent system paradigm the so-called hub location problem is solved.

The introduction gives a short description of the subject of the research and serves the purpose of providing context to the topic. This chapter starts with introducing express service providers, multi-agent modelling and gives the company profile of ORTEC, where the research was performed. Next, the research goal and corresponding research questions are described. This is followed by a more detailed introduction to the characteristics of express networks, the hub location problem and the ontology, which includes a description of express networks in terms of multi-agent system characteristics. The methodology and the scope of the thesis are described in 1.8 and 1.9 respectively. The chapter ends with an outline of the thesis.

1.1 Express service providers

Express service providers

Express service providers transport parcels via their door to door road and air networks. Those parcels have to be delivered within predefined service windows varying from delivery on the same day to one, two and multiple day delivery. Furthermore express service providers guarantee to service any address within the countries they operate. As a consequence, they have an extensive, complex network of vehicles and sorting centres to keep up with service requirements. Most express service providers thus make use of computational models which represent the real-life situation of their complex express networks to make fact based decisions on network design.

1.2 Multi-agent systems

Multi-agent systems

As the term multi-agent system (MAS) suggests, it is a system consisting of many agents. There is no general consensus on the definition of an agent (Russell & Norvig, 2003). Most definitions describe an agent as being an independent object that can make decisions without being controlled by another object. A much used definition of agents is made by Wooldridge & Jennings (1995). They divide agents in two categories: the weak definition of agents and the strong definition of agents.

Weak agent definition The weak definition is the most general one. It describes agents as entities with the following properties: autonomy, social ability, reactivity and pro-activeness. Autonomy relates to the ability of an agent to have control over its own actions. Social ability describes the fact that agents can interact with other agents through some sort of agent-communication language. Reactivity means that agents perceive their environment and can act upon changes in it. Pro-activeness means that an agent is able to act on the environment, or other agents, without the need for an external trigger. The weak definition is commonly used as approach to practical modelling challenges.

Strong agent definition The strong agent definition is an extension to the weak definition. It adds mental notions to agents like: beliefs, desires and intentions (see Shoham, 1993 for an extensive introduction to these BDI-agents). This definition is often used when modelling more human-like behaviour. It allows agents to have beliefs about the knowledge of other agents, or even beliefs about the beliefs of other agents. This can be helpful when modelling strategic behaviour of complex social systems. Since the problem modelled in this research is of a practical technical kind the weak definition of agents will suffice.

1.3 ORTEC Consulting Group

ORTEC This research was performed at ORTEC bv, more specifically, at the ORTEC Consulting Group (OCG). ORTEC is one of the largest providers of advanced planning and optimisation software solutions and consulting services. Their solutions result in optimised fleet routing and dispatch, vehicle and pallet loading, workforce scheduling, delivery forecasting and network planning. ORTEC provides best-of-breed, custom made and SAP® certified and embedded solutions, supported by strategic partnerships. In the area of Advanced Planning Solutions, ORTEC has over 1,450 customers worldwide, over 550 employees and offices in Europe, North America, Asia and the Pacific Region (ORTEC, 2010).

OCG The ORTEC Consulting Group is the strategic consultancy branch of ORTEC. They assists companies and organizations to make informed, fact-based decisions based on thorough analyses. This provides decision-making confidence that is highly valued by their customers. By analysing historical data and projected forecasts, their specialist consultants can calculate scenarios that provide a clear view of the facts and figures. This information is vital for balanced decision-making (ORTEC Consulting Group, 2010). One type of customer of the OCG are express service providers for which the OCG analyses and designs express networks.

1.4 Express network characteristics

Express networks An express network is the network that is utilised by an express service provider to transport parcels from their origin (O) to their destination (D). Any transportation method could be used in computing an express network. This research focuses on computing an efficient road network. The air network is merely used when the road network is not fast enough to deliver the parcel on time. A cost penalty is assumed for air travel. Other transport modalities are excluded.

The simplest network imaginable is to connect each origin with each destination. However, this will incur massive cost, since each origin-destination (OD) combination will need its own truck. Most of which will probably never be full. To save on cost, points are introduced at which parcels from multiple origins will be sorted. These points of consolidation are also known as sorting centres. Consolidating at sorting centres saves on the total driven kilometres and reduces the total amount of needed trucks, thereby saving costs (Abdinnour-Helm & Venkataramanan, 1998).

Depot *Hub* Generally, the transport process is as follows. Parcels are collected at the origins and transported to local sorting centres, which are referred to as depots. From these depots the parcels are transported to larger sorting centres, known as hubs. After sorting all parcels at the hubs, they are brought to the hub closest to the destination depot. Following arrival at that hub, they are transported via a depot near the destination to their final destination. An efficient network foremost depends on the volumes of parcels, their origin-destination patterns, their service type (one-day or multi-day delivery), the transportation cost at different levels of the network and the fixed and variable costs of hubs and depots.

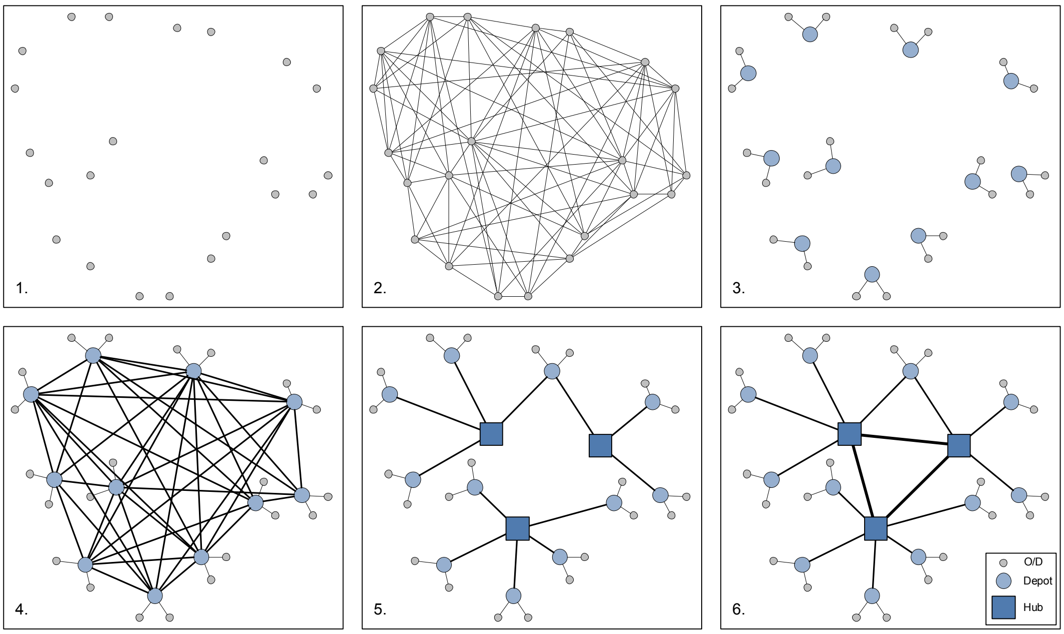


Figure 1.1: A fully connected network versus a depot-hub configuration

Need for consolidation Figure 1.1 illustrates the need for consolidation. Part 1 shows a fictive dispersion of origins and destinations. From every origin, parcels should be able to flow to any destination, resulting in a fully connected network as depicted in Part 2 of Figure 1.1. To reduce local complexity depots can consolidate demand and supply as shown in Part 3. Part 4 illustrates the still elaborated network resulting from connecting the depots. In Part 5 hubs are introduced to solve this issue. By connecting the hubs as in Part 6 all origins and destinations can reach each other, but the number of connections is drastically reduced. Transporting the same amount of cargo through less routes means that a single route will have more cargo to transport. This leads to full trucks and hence less trucks are needed. For instance, three routes that have half a truck load to transport require three trucks. Bundling these three routes into one route combines to a load of one and a half truck. Now, only two trucks are needed and the cost of one truck is saved.

Research focus This research focuses on choosing these hub locations to create a network that is as efficient as possible. This topic is called the Hub Location Problem (HLP). It means that depot locations will be fixed (as in part 4) and hub locations will be chosen by the model and their interconnection will be determined (Parts 5 and 6). The hub location problem will be extensively introduced in Chapter 2.

1.5 Ontology

Deciding upon a modelling approach is not a matter of choosing the right or wrong method for a problem, but a question of finding a suitable way to look at a problem. Any type of problem can be modelled with any type modelling paradigm, however choosing an approach that suits the problems characteristics can help easing the modelling process and creating a better understanding of the dynamics of the system under investigation. The next two sections explain what problem characteristics make mathematical modelling and more specifically the agent modelling paradigm a suited modelling approach.

1.5.1 Mathematical modelling

Enserink et al. (2003) state three reasons for choosing a mathematical model as an approach to solving a problem:

Many factors
1. The problem contains many factors that have many relations between them.

Dynamic interactions
2. The systems behaviour is a consequence of many dynamic interactions in the system.

Data availability
3. Data is available to construct and validate the model.

All three reasons apply to the HLP. First of all, express networks contain a lot of factors. For instance, the express network of TNT Express in France alone already contains almost one hundred nodes. Each node collects and distributes thousands of parcels a day, requiring hundreds of trucks for transport. Second, the nodes are tightly coupled and therefore heavily interact. For instance, changing a single depot into a hub can influence all other nodes in the network and the chosen modelling approach should be able to deal with such dynamic interaction. Third, data of an express service provider is available to ORTEC to construct and validate the model. Furthermore, standard data sets are available in literature to be able to compare different modelling solutions.

1.5.2 Multi-agent modelling

Of all possible mathematical modelling approaches, the decision is made to use the MAS perspective. This decision if foremost based on similarities between the characteristics of the HLP and characteristics of MAS. The main reasons are as follows:

Type of complexity

1. As Luck et al. (2003) state, the complexity of many systems results from the reciprocity of components of those systems and the agent modelling approach provides a natural way to model such interactions. The network characteristic of the HLP is a typical example thereof, since it is the tightly coupled interdependence between nodes that creates the complexity.

Network characteristic

2. It is very well suited for making a distributed model since the network like problem has its similarities with the network structure of a distributed solution.

Autonomous character

3. The autonomous character of agents matches the real-life processes in express networks, where managers try to get the best performance at their specific node. On a higher aggregation level, countries are autonomous on their national network design and international freight often influences the overall network structure in a limited manner.

Distributed data

4. Although local changes often affect a great part of the system, a decision on making such a change can often be made with merely local data.

With a clear picture of the researched problem and chosen approach the research goal is stated next, followed by the research questions.

1.6 Research goal

When the OCG advises express service providers on efficient network design, they use operations research methods. They have never approached network design and specifically the hub location problem (HLP) from a multi-agent systems (MAS) perspective. Moreover, literature research shows that the HLP has never been approached by anyone using an MAS perspective. However, the MAS perspective could turn out to be a valuable new approach to the HLP, because it provides a natural way of modelling interactions between tightly coupled, interdependent nodes; which characterizes the HLP. Since the OCG is always keen on untried modelling techniques to generate new insights; they are very interested in the MAS paradigm as an additional approach to the HLP. As such, the goal of the research is to:

Research goal

Design a model that can solve the hub location problem for express networks with use of multi-agent modelling.

1.7 Research questions

The research questions section states the main questions and related sub-questions that will aid in structuring the research. The research questions are divided in three separate categories: Domain specific research questions, design related research questions and evaluation related research questions. These classes typify the origin and are therefore used to introduce the research questions.

Domain specific research questions

The research starts with describing the context of the subject. The domain specific questions help to get a thorough understanding of the researched matter and therefore form the knowledge basis for the design that follows. The research problem stems from express networks that harnesses the theoretical Hub Location Problem. Therefore these two subjects form the first two main questions. The problem will be approached from a MAS perspective and hence, MAS is discussed in both questions.

Research questions

1. How do concepts of MAS apply to express networks?
   (a) What is the multi-agent system paradigm?
   (b) Which definition of agents is most suited to this research?
   (c) What are the characteristics of express networks?
   (d) Why do express service providers need computational modelling?
2. What is the Hub Location Problem?
   (a) What is the formal description of the Hub Location Problem?
   (b) How did the Hub Location Problem originate?
   (c) How can the characteristics of MAS be applied to the HLP?

Design related research questions

Once a solid knowledge base is established the research moves towards the design. The questions in this category aid in determining what the design should look like, how it can be made, what is needed to create the design and finally the way a conceptual model can be implemented in code. Prometheus is the methodology used for this research. It is an extensive methodology that supports design from early requirements gathering through detailed design and can even generate a code framework. It is introduced in detail in Section 1.8. The language used to code the model is JACK Intelligent Agents. The choice for JACK mainly follows from using Prometheus, which can generate a code framework of the design in JACK. Further introduction to JACK follows in Section 1.8.4.

3. What are the requirements of the design?
   (a) What requirements do involved actors impose?

   (b) What is the scope of the model?

4. What does a multi-agent express network model look like?
   (a) What data is needed to run and test the model?
   (b) What are the variables of interest of the model?
   (c) What does the system specification of the model look like?

5. How can the conceptualisation be translated to a tool?
   (a) What does the detailed design of the model look like?
   (b) How do the Prometheus artefacts relate to the concepts in the JACK Agent Language?

Evaluation related research questions

Once the design is finished and implemented its quality can be determined. The questions in this category help identifying strengths and weaknesses of the product of this research and aid in convincing the reader of the quality of the model and the conclusion following its design.

6. What is the added value of the tool?
   (a) How can the output of the tool be represented to aid express network analysis?
   (b) What is the impact of varying the parameters of the tool?

7. How well is the tool constructed, according to the verification methods used?
   (a) Are there modules in the tool producing unexpected output?
   (b) What are the strengths and weaknesses of the implemented model structure according to experts?

8. How valid are the results of the tool according to experts?
   (a) What are the strengths and weaknesses of the output of the tool according to experts?

9. How well suited is multi-agent modelling to solve the hub location problem in express networks?
   (a) What are experienced advantages of multi-agent modelling?
   (b) What are experienced drawbacks of multi-agent modelling?

This concludes the research questions. In the remainder of the introduction the methodology will be described, followed by the outline of this thesis.

1.8 Methodology

To translate the HLP into a multi-agent model a methodology is selected. To make sure a suited methodology is chosen, factors that should influence this decision are identified. Literature is explored to identify methodologies that are well enough described to score them on the identified factors. Furthermore, only methodologies of which literature shows that they are mature are considered. The candidate methodologies are scored on the identified factors after which the selected methodology is described in detail. Following, the agent language used to code the model is described after which the section concludes with the further outline of this thesis.

1.8.1 How to decide on a methodology

Just as with choosing a modelling approach, there is no right or wrong methodology. However, a well suited methodology eases the design process and can significantly improve the quality of the conceptual design. The following factors were considered while choosing a design methodology:

Level of detail

1. The level of detail that the conceptualisation of a methodology supplies should correspond with the detail required to ease the programming process, without involving more effort than needed.

Tool support

2. A tool to support the use of the methodology. It quickens the constructions of the needed diagrams, can provide consistency checking and in some cases even provides code generation.

Integration with other design steps

3. The level of integration of the methodology with other design steps than conceptualisation, like the specification, verification and validation. Although these are not necessarily the main purpose of a methodology, it can help synchronizing the entire design cycle. For instance, it may require explicit definitions of the specifications and check these with model constructs therewith aiding the verification of the resulting model.

1.8.2 Several options

Four well known methodologies were considered. This section provides a comparison of which the conclusions are foremost based on Sterling & Taveter (2009) and Dam & Winikoff (2003). Dam & Winikoff (2003) give a detailed comparison of MaSE, Tropos and Prometheus, scoring the methodologies on forty different factors. Sterling & Taveter (2009) provide a more descriptive comparison between Gaia, MaSE, Tropos and Prometheus. Besides these two, other papers are used to underpin the characteristics of the analysed methodologies.

Gaia

- In **Gaia**, systems are organizations of interacting roles (Wooldridge et al., 1999). It is an easy to understand methodology, which has made it popular with industry and students.

1. *Level of detail*: High, interaction between agents and the workings of services within the agents is specified.
2. *Tool support*: Limited.
3. *Integration with other design steps*: Limited, since Gaia does not cover requirement gathering.

MaSE

- **MaSE** stands for Multiagent Systems Engineering and is one of the earliest methodologies for agent-oriented systems engineering (DeLoach et al., 2001).

1. *Level of detail*: Very high, use of state diagrams to model the different agent behaviours.
2. *Tool support*: agentTool (Developing Multi-agent Systems with agentTool) is a plug-in for the Eclipse software development environment[1] (DeLoach & Wood, 2001). It provides consistency checking and limited code generation to aid verification.
3. *Integration with other design steps*: More detailed than Gaia, because of (limited) requirements support.

Tropos

- **Tropos** is an agent-oriented methodology that assumes the strong (BDI) definition of agents (Bresciani et al., 2004).

1. *Level of detail*: Highest, detailed diagrams for each stage and use of UML for information modelling.
2. *Tool support*: TAOM4E, Tool for Agent Oriented Modelling, is an Eclipse plug-in (Perini & Susi, 2004). Its support each stage of the methodology.
3. *Integration with other design steps*: Tropos is an extensive methodology that covers the entire software life cycle. From early requirements to agent code, testing and acceptance levels.

Prometheus

- **Prometheus** is developed for ease of use, focusing on modellers learning the multi-agent paradigm (Padgham & Winikoff, 2002). It support the use of both the weak and strong definition of agents.

1. *Level of detail*: High, code generation out of conceptual diagrams, interaction diagrams in AUML (Padgham et al., 2007).
2. *Tool support*: Well supported with the Prometheus Design Tool (PDT), a plug-in for the Eclipse (Thangarajah et al., 2005). The PDT is able to help the verification of the model with consistency checks during the design and the option of code generation out of the conceptual diagrams.
3. *Integration with other design steps*: Not as extensive as Tropos but more elaborate than Gaia and MaSE. Especially because of the goal oriented early requirements phase.

It is the Prometheus methodology that suits this research the best out of these four. Since the research uses the weak definition of agents, Tropos is ruled out. The

[1] www.eclipse.org

downside of Gaia is that it does not cover requirements gathering. MaSE is a very mature methodology and just as Prometheus has good tool support. Both have a high level of detail and provide consistency checking and code generation and therefore each of the two would fit this research. The decision is made to use Prometheus, because the designers of Prometheus aimed on developing a very practical methodology and that brings further confidence in the accessibility of the tool. Now that the decision for Prometheus is made, a thorough introduction to this methodology is given in the next section.

1.8.3 Prometheus

This section will describe the phases of the Prometheus methodology that are visualised[2] in Figure 1.2. In each phase reference is made to the artefacts that belong to the methodology and the overviews that are part of the Prometheus Design Tool (PDT). For a separate overview of the artefacts the reader is referred to Appendix B.1. For an extensive description of the Prometheus methodology the reader is referred to Padgham & Winikoff (2004).

Prometheus Design Tool

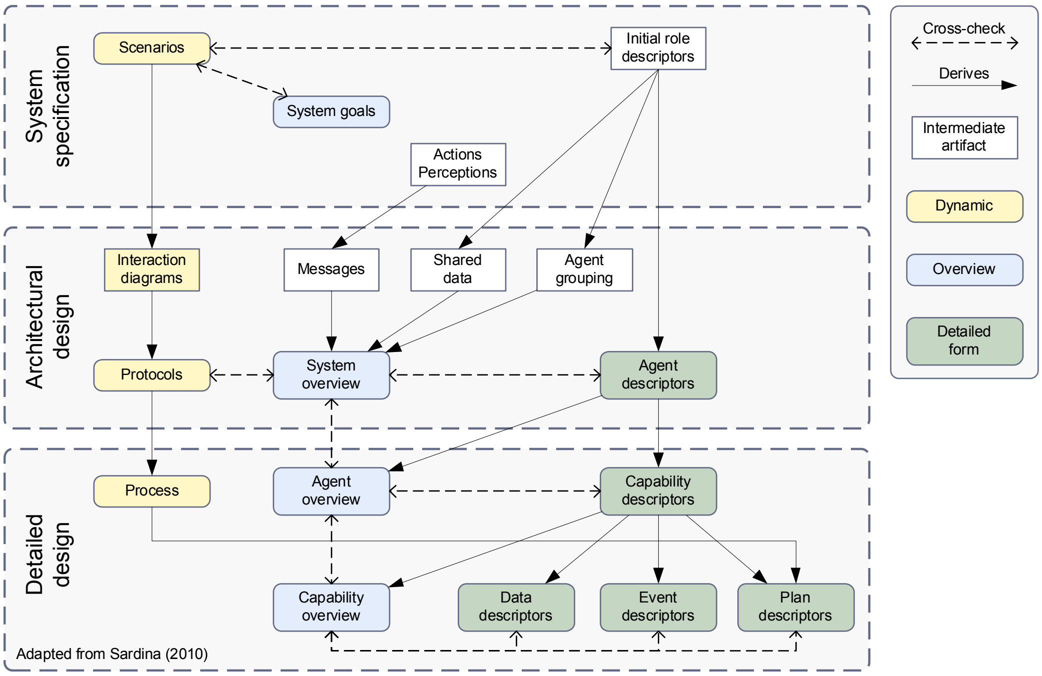


Figure 1.2: Prometheus methodology

System specification

The system specification is the highest aggregation level design phase of Prometheus. It aids in determining the scope of the model by specifying the interaction of the

[2] Adapted from Sardina (2010)

system with the environment. It utilises scenarios, system goals, roles, actions and *Percepts* percepts. Percepts are bits of information that come from the environment and go into the system. For instance, this could be a user trying to influence the system or a robot utilising a camera to retrieve details about its surrounding. Information *Actions* that flows out of the system, going into the environment are depicted as actions. An action could for example be a robot physically pushing a button or a software agent *Scenarios* providing a book suggestion to a website visitor. Scenarios created in the scenario overview, are use cases in which specific situations that occur in reality can be logged and hence help specifying what situations the model should be able to facilitate. *Goals* These situations can then be translated to the goals of the model in the goal overview of the PDT. In its turn, the goal overview serves as input for the system role overview. *Roles* There, similar goals and goals that apply a same procedure are grouped into roles. All these identified artefacts are then used in the analysis overview to identify their interaction with the environment. This is, as all steps in the methodology are, an iterative process.

Architectural design

Agents The second phase is the architectural design. Its aim is to identify the agents in the model, what they will be able to do and how they will interact. The agent role overview is used to link the roles that were created during the system specification to agent entities. The data coupling overview serves the purpose of identifying the *Databases* databases that the different roles use and/or produce. The newly created agents and databases are together with the actions and percepts identified in the system specification, automatically put in the system overview. The agents have to be linked to the actions, percepts and databases according to the roles they represent. Next, the rest of the interaction between the agents can be established in this same overview[3]. This is done by specifying the messages and protocols that exist between agents. *Messages* Messages are signals from one agent to another or of one agent internally. Messages are a single instance one directional occurrence. When multiple messages are sent *Protocols* as part of the same topic a protocol is defined. A protocol can be depicted as a conversion built up out of messages, specifying the direction and order of occurrence of those messages.

Detailed design

In the detailed design, the internals of the agents are modelled. All artefacts that are linked to an agent in the previous phases are inserted in the agent overview diagram of that agent. They are then linked to each other by use of plans and capabilities. *Plans* A plan defines what to do when a certain trigger is received, which can either be a message or percept. Since an agent overview can quickly become quite complex, *Capabilities* capabilities are introduced that group a part of the functionality of the agent. The internals of capabilities consist therefore of the same content as an agent, meaning

[3] The already determined actions, percepts and databases are also part of the interaction between agents.

that capabilities can be nested going into increasing detail. Capabilities can be viewed in the capability overview.

Research using Prometheus

Complex Surveillance Valencia-Jimenez & Fernandez-Caballero (2006) describe the application of MAS as a solution to increasingly sophisticated surveillance systems. They used Prometheus to design a MAS that allows the information of sensors to be coherently interpreted with information of other nearby sensors. This relieves problems with temporal and spatial isolation of sensors and allows for auto-adaptation to unforeseen events. In a similar way, the model in this research has optimisation logic that tries to improve the system performance as a whole by looking for local conditions that suggest an imperfection. Improvements made to local conditions are, as in the referred research, done coherently with nearby nodes.

Robotics application Gascueña & Fernández-Caballero (2009) use the Prometheus methodology to model behaviour for a robot that should be able to detect and follow humans. The advantages they experienced by using Prometheus are diverse. One reason for choosing Prometheus are the guidelines of the methodology which help to determine the modelling elements. Another reason is the communicative strength. In their view, designing a model in Prometheus creates a clear picture of the models functionality and thus makes it easier to explain how it is build and why it is build this way. Furthermore, they value the fact that Prometheus supports auto-generation of code from the conceptual diagrams[4]. Although the model described in this paper is not comparable with the model presented here, it does provide an insight in the advantages of the chosen methodology.

First non-BDI Prometheus model Although more and more models are being developed using Prometheus, the resulting systems have so far always been modelled using the strong definition of agents[5] (Padgham, 2011). Prometheus is definitely suited for designing BDI-agent systems, but it has always been developed with the goal of supporting both the strong and weak definition of agents. This research will be the first described use of the Prometheus methodology for the design of a multi-agent system with agents according to the weak agent definition.

1.8.4 JACK Agent Language

A conceptual agent model can be implemented with use of many different agent languages. Since Prometheus allows code generation in the JACK Agent Language (Howden et al., 2001) the choice for this particular language followed naturally. JACK is a commercial agent language that proved itself in numerous operational applications by businesses and university researches most of which are performed at RMIT[6] at the

[4] Note that the Prometheus Design Tool is not able to fully generate the model code, but it generates a code framework. This helps the consistency of the code during programming.

[5] See Section 1.2 for an explanation of the different agent definitions.

[6] RMIT stands for Royal Melbourne Institute of Technology and is located in Melbourne, Victoria, Australia. Agent research is performed there at the Agent Group (http://www.cs.rmit.edu.au/agents/www/) within the School of Computer Science and Information Technology.

same department responsible of the development of the earlier mentioned Prometheus Design Tool. JACK is a Java based language that tries to ease technology transfer from research to industry, which is another argument underpinning the choice to use this language. To provide insight in the structure of JACK an explanation is given of the way it extends Java.

JACK extends Java in three major ways. It adds a set of syntactic terms, it adds a compiler that translates these syntactic terms into pure Java code that can be interpreted by any Java compatible platform, and it adds a kernel for required run-time support. The syntactic terms encompass keywords like *agent* or *plan* to support these Prometheus artefacts. Furthermore, terms are added to support agent decision making, like the choosing the most suited plan out of all that are applicable to the current situation. The compiler, as said, transforms these semantic terms into general Java entities. The main advantage that follows this process, is that the program can run on any platform that supports Java and therefore does not have to particularly tailored to handle JACK semantics. The kernel that is added takes on tasks like the management of concurrency among tasks, handling failure of tasks and supporting efficient agent communication. A last beneficial feature of JACK is its modular setup. This allows easy extension of any model produced with this language. For instance, when one would want to use a high-level symbolic protocol for agent communication such as KQML of FIPA's Agent Communication Language (Labrou et al., 1999), the standard Java class used for communication can simply be overridden. This concludes the section on methodology selection and introduction.

1.9 Chapter conclusion

This chapter introduced the research goal and the related topics, thereby answering the first research question: *how do concepts of MAS apply to express networks*? Specifically, it was shown how concepts of MAS apply to express networks by stating the definition of MAS and the used agent definition, and describing express networks. Multi-agent systems as the name suggests, are systems of many agents that are designed to achieve their individual goals, either jointly or in a competitive manner. The weak agent definition is used in this research where agents are defined to be autonomous, to have social ability, to be reactive and to be pro-active, but the agents do not have mental notions like beliefs, desires and intentions. As such, the agents can make their own decisions, can interact with other agents, can respond to changes in the environment and can act on the environment without the need of an external trigger. The agents are used to model express networks. Express networks transport parcels from and to any location within a pre-defined service time. To aggregate the parcels in efficient truck loads, use is made of two types of consolidation centres: depots and hubs. Depots are the first and last stage consolidation centres and hubs aggregate and distribute the parcels in between the depots. The complexity of designing express networks evolves from the heavy interaction between these depots and hubs. MAS is used to create a model for the design of express networks, because agents provide a natural way of modelling such complexity. The following section will put forward the layout of this report.

1.10 Thesis outline

The research questions as mentioned in Section 1.7 form the basis of the report structure. The first questions have been partially answered in the introduction. The next chapter covers the hub location problem in detail. Then the design requirements and the scope of the model are specified in Chapter 3. Chapter 4 specifies the model design process using a systems goal overview. Because the timeline of the research does not allow for a full implementation of the model design, an isolatable part of the model is programmed as proof of concept. This process is described in Chapter 5. Consequently, Chapter 6 demonstrates how the model can be applied to real world situations and create added value. Expert verification and validation is depicted in the following chapter. Based on all the findings in the previous chapters a conclusion is drawn on the added value of MAS in solving the HLP in express networks. Recommendations for further research are given accordingly. Finally, a reflection of this research is covered in Chapter 9.

Chapter 2

The hub location problem

As first described in Paragraph 1.4 this research focuses on the hub location problem. To fully grasp the nature of this problem and how it fits this research, this section describes it in detail and places the research in context with available literature. First, a formal description of the specific type of the HLP addressed in this research is given, followed by the origin of the HLP. Next, research regarding the HLP in the fields of operations research and multi-agent systems (MAS) is described. This introduction on the HLP finishes by classifying the HLP in terms of MAS characteristics.

2.1 Problem

To provide further insight in the HLP a description of the problem as it is depicted in the operations research field is given in the next section. It is followed by an elaboration on the origin of the HLP.

2.1.1 Formal description

Hub location problem

In the operations research field there has been a lot of research on the hub location problem (HLP). More specifically, the problem is described in this field as an uncapacitated multiple allocation hub location problem (UMAHLP). The classification uncapacitated means there are no capacity restrictions on the hubs and transport connections between the hubs. Since this is a strategic research with a long time horizon, it is assumed that in practice a location of the required size can always be found near the designated location. This fact together with the reduced complexity of modelling an uncapacitated hub problem has led to this decision. Note that although this leads to less mechanisms that need to be modelled, it does not mean that the problem becomes less complex to solve. The term multiple allocation depicts that depots may be allocated to multiple hubs instead of one single hub. This will help achieving a solution that meets the service requirements.

UMAHLP

Uncapacitated

Multiple allocation

2.1.2 Origin

Spatial and Location theory

Research has been conducted on spatial interaction theory and location theory since the late 1950s (B. Kara & Taner, 2011). Spatial interaction theory investigates the travel demand[1] between fixed locations. Location theory on the other hand, treats travel demand as fixed and focuses on finding the best locations to serve the this demand. It was O'Kelly (1986) who first thought of the synergy between these two fields and described it in his classical paper *The location of interacting hub facilities* in 1986. He realised the importance of the interdependence between chosen consolidation locations (hubs) and the flows through the network. This can be explained as a location theory problem with the addition of allocation assessment. Before O'Kelly, the locations were determined based on exogenous factors, like distance to the origins and destinations of the demand. Later, the locations were determined based on endogenous factors as well (e.g. how the flows in the network will be allocated to the hub). The reason for O'Kelly to approach the problem in this manner is because he realised that the way locations are chosen, should be based on the reason hubs exist, namely cost savings. To create the most cost savings one should try to consolidate only in case this results in cost savings. Hence, not only exogenous factors like distance should be considered, but the resulting flows (partly consolidated and partly unconsolidated) should be considered as well, since they show the realised cost savings. A more thorough description of the history of the HLP is given by B. Kara & Taner (2011).

O'Kelly

2.2 Research fields

As stated before, the HLP is mostly addressed by the operations research field. The next section will therefore briefly appoint reference to work done in that field. Secondly, literature research is conducted to identify applications of MAS to the HLP and similar problems.

2.2.1 Operations research

HLP in operations research

Since O'Kelly's introductory paper, a lot of research has been done on the hub location problem. Most of this research took place in the operations research field and focused on three parts. First of all, many different types of the HLP have been defined like the single versus multiple allocation problems and the capacitated versus uncapacitated versions of the HLP. Second, a lot of research focused on creating stricter mathematical definitions of the HLP, making it faster to solve or providing a better result. The third major focus is on finding new algorithms to solve the HLP (like Benders decomposition (Benders, 2005), Tabu search (Glover, 1989, 1990),

[1] Travel demand in this case refers to the demand of passengers to fly from one place to another, since early spatial and location research was often focused on airline networks. The travel demand in this context may however be interpret as being data that needs to be transported through a communication network or parcels that need to be transported through an express network or the like, as well.

interior point method (Karmarkar, 1984), Lagrangian relaxation (Everett, 1963), genetic algorithms (Holland, 1975), etc.).

HLP overview papers

Recent overviews of papers addressing hub network design are given by Alumur & Kara (2008), ReVelle et al. (2008) and the before mentioned paper of B. Kara & Taner (2011). A good overview of operations research applied to the hub location problem as investigated in this research (UMAHLP) can be found in Van Essen (2009) and Hekmatfar & Pishvaee (2009).

2.2.2 Multi-agent systems

Although operations research provides many different approaches to HLP, there is no literature in which the HLP is solved by the use of MAS. However, there are some related problems in which MAS have been applied. Pick up and delivery problems are the most alike type of problem, since those problems are also defined by resource allocation issues.

PUD routing using MAS

A first example of MAS applied to pick-up and delivery (PUD) problems in networks with depots and hubs is by Claes et al. (2010). This research focuses on adapting the routes of parcels to a fixed network; more specifically, it considers the PUD between depots and customers. It assumes the network between depots and hubs to be given and determines PUD routes based on expected parcel flows. This is in contrast to the proposed research, which considers the PUD as a given and optimises the network between depots and hubs to be as efficient as possible.

Operational planning using MAS

Similar is the research of Sharma & Tran (2004) on dynamic scheduling of transports. They have designed an approach to real-time transport planning, which means that they have a schedule with for example PUD rounds and their model deals with real-time disturbances. For instance, if a trailer breaks down, the model will find a different trailer as replacement or it reschedules the parcels meant for the broken trailer by allocating the parcels to different, already scheduled, trailers that have spare space. This operational planning problem contrasts the presented research since that focuses on long term strategic planning. Once a solution is calculated, it will stay fixed. It will not have the goal to adapt to unforeseen real-time changes.

Overview of MAS in transport

Apart from these differences, both the related problems and the research presented here are allocation problems in transport modelled from an MAS perspective. For a good overview of agent based approaches to transport logistic problems reference is made to Davidsson et al. (2005).

2.3 The HLP in terms of MAS

Several characteristics typify MAS (Vlassis, 2007). In this section these characteristic will be used to classify the hub location problem. First, short definitions of the characteristics are given, after which they are applied to the subject of this research. The six characteristics as identified by Vlassis are:

1. *Homogeneity* describes the level of difference in goals, perceptions and actions between agents.

2. *Environment stability* is a measure of change of the environment that the agents can perceive.
3. *Perception* addresses the information that can be observed by agents.
4. *Control* is defined as having the authority to make decisions.
5. *Knowledge* of agents about the knowledge of other agents, deals with the level of mental notions that agents utilise.
6. *Communication* between agents. The way communication is organised in the model.

Homogeneity An express network mainly consists of depots and hubs that are connected by transport routes. It is expected that the depots and hubs will be represented by agents that are connected by the communication between them. There are two general types of agents in the model. Agents representing nodes in the network (i.e. depots or hubs) and optimisation agents. The latter group is very heterogeneous, because different optimisation agents are specialised in optimising different types of inefficiencies in the model. Hence, their behaviour differs. The premier group is considered homogeneous, since all nodes in the model are represented by the same type of agent. It should however be remarked that hubs do have a different behaviour than depots, nevertheless they are modelled within the same agent. The reason behind this decision is the fact that hubs are created by 'upgrading' a depot and hence closing a hub is followed by 'downgrading' that specific node to a depot. In case different agents would represent depots and hubs, a downgrading would involve terminating the hub agent and instantiating a depot agent. However, data gathered at the node during the time it was a hub, may still be valuable. Having one agent type competent to represent both depots and hubs enables easy data management at each node. This combined agent type therefore has stable schizophrenic behaviour, acting differently when being a hub or depot. All agents being depots have the same type of goals, perceptions and actions. Ditto, all hubs behave alike. Considering the greater portion of agents relative to all agents are depots, the model is considered to be highly homogeneous.

Environment stability Environment stability can be characterised as static or dynamic. The environment in this context is not the environment of the entire model, but described from an agent point of view. The stability is not always clear since the agents are influenced by external factors and other agents. For instance, spatial dispersion and transport cost (independent of the load on a truck, so €/km)) are stable, but the perceptions, goals, and actions of neighbouring agents can vary. For instance, the cost of transporting parcels via a certain hub can vary dependent on the volume already assigned to that hub. When these cost increase too much, the perception of a depot subscribed to that hub will change from perceiving the hub as an attractive option to an unattractive option to connect with. On the other side, depots continue to look for the cheapest connection to disperse their parcels, which reflects a stable goal. However, in multi-agent systems the environment is generally depicted as the variables that are influenced by the agents in the model. Since this model is situated in an environment that will not change during a model run, it is depicted as static.

Perception A characteristic of MAS is that data is generally distributed. It can differ spatially, temporally and semantically. Spatially, when data is only available to agents in a specific location. Temporally, when the time at which data is available to agents differs. Semantically, when different agents can have different interpretations of the same data. Information in the express network model is expected to differ spatially and temporally. Parcel information, at the start of the model, is only known at their origin, which accounts for the spatial availability. While running the model, information on the parcels will be communicated to other agents and thus flow through the model, which results in temporal difference in availability. Semantically different interpretations of the data are not expected in the model.

Control In general the system is not centrally controlled, i.e. the control is divided over all agents in the model. Only very limited control is assigned to central agents. There is a central entity to support the collection of statistics and to start and stop the model and to announce new stages[2], but no further decision control will be allocated to this agent.

Knowledge This characteristic is most relevant in BDI-agent systems (using the strong agent definition) where complex social systems are modelled. This characteristic then describes the beliefs that agents have about data of other agents or even the beliefs of other agents. The model presented in this research uses the weak agent definition and has a very open communicative structure. The agents do not act strategically selfish, but share their information at will (i.e. they only enact truth telling). Agents will ask other agents to provide information when needed.

Communication The last characteristic is the nature of communication between agents. Communication is a vital part of multi-agent models. For instance, because computations done by single agents have no meaning without sharing the results. Another reason is that data is dispersed and needs to be communicated to be utilised. The protocol (syntax and semantics) used to communicate differs between agents that are cooperative and therefore coordinate, and agents that are self-interested and hence negotiate. The agents in the presented model are cooperative and therefore coordinate their actions in an attempt to achieve the lowest overall network cost. They are able to communicate with agents within a certain distance and with agents that they are connected with. The design of the communication protocol is supported by the Prometheus methodology and described in the conceptualisation. The Prometheus methodology itself is described in the conceptualisation as well (Section 1.8).

2.4 Chapter conclusion

The second research question, '*What is the HLP?*', is answered by giving a formal description of the HLP (first introduced by O'Kelly (1986)) and its history, and by applying a MAS view on the HLP. The variant of the HLP that is studied in this research is the UMAHLP, meaning that the hubs are uncapacitated and that multiple

[2] These stages are as follows. First an initial set-up of hubs is made and then all depots subscribe to nearby hubs. Third, all parcels will have routes assigned. The fourth and final stage consists of optimising the current network.

allocation (connecting one depot to more than one hub) is allowed. This chapter showed a knowledge gap, as literature research done so far did not show any research that used multi-agent modelling to design express networks. The knowledge base as formed in this and in the previous chapter, served as input for the design phase that starts in the next chapter with gathering the requirements.

Chapter 3

Requirements specification

The previous chapters established the research goal, the research questions, the current knowledge base and the nature of the HLP. However, a thorough understanding of the nature of the researched problem is not enough to come to a successful model design. Apart from addressing the nature of the theoretical problem, the design should apply to the wishes of involved stakeholders and to a clearly defined scope. The scope specifies the boundaries of the part of reality that is modelled. This chapter first states the design requirements and follows with the scope specification.

3.1 Design requirements

Requirements specify what the model should be able to do and in which cases it can be considered a success. Since the success is determined by the involved actors they form the source of the requirements. Although usually a formal stakeholder analysis is applied to determine the requirements, it was not deemed useful for this research[1]. Extensive stakeholder analysis is foremost used to arrive at a clear system description of fuzzy problems. Since the hub location problem already has a clear formal description, because of its theoretical nature and since the involved stakeholders all have the same business-customer relation, the use of extensive stakeholder analysis is not required. Hence, a description of the involved actors is given. From those descriptions the specific requirements are derived. Subsequently, these requirements are classified into three categories that indicate the type of influence that each has on the model design.

[1] However, a swift actor analysis is performed to investigate strategic interests of involved actors to underpin the uniformity of the main goals of the actors. A description of this actor analysis is presented in Appendix A.

3.1.1 Source of requirements

The actors involved in the problem are the customers of the Express Service Providers (ESP), the ESP themselves, and the ORTEC Consulting Group (OCG) where this research is performed. A description of these three actors and their influence on the model design comes next, followed by the three categories these requirements are grouped by.

Customers Customers make use of ESP to have their packages transported from an origin to a destination. This makes up the first and most basic requirement: flow conservation. This means that all parcels that are offered should be able to be transported by the network of the ESP. Secondly, customers want their parcels to be delivered before a set time. The delivery time agreed between the customers and the ESP is called service. The obligation of the ESP set by this agreement is called the service commitment. Apart from being a wish for the customer, it is of great importance to the ESP as well, since delivering a parcel later than the agreed delivery time imposes a penalty that the ESP should pay to the customer. Moreover, it worsens the reputation of the specific ESP.

Customer requirements:

- *The model should create a network that is able to transport all offered parcels (flow conservation).*
- *The model should create a network containing routes that can deliver each parcel before their service time expires (service commitment).*

ESP The goal of ESP is not only to transport parcels, but to make a profit out of it at the same time. This can either be achieved by charging their customers a higher fee or by reducing the cost arising from their transport network. Since higher fees reduces the competitiveness, their preference is to reduce the network cost. This forms the first requirement of the ESP and actually is the overall objective of the model. The existing network of the ESP is most probably not as efficient as it could be, since it was not designed to handle the current volume distribution. Instead it has gradually grown with increasing demand. Therefore, their second requirement is that the model should be able to start optimisation not only with their current network structure of depots, hubs and the routes between those, but also with just the current depots and their volume distributions. A last requirement that the ESP impose concerns the used transport modalities. Their networks are designed around road transport and their main backup modality in case service commitments can't be made is air travel. As such, these two modalities will be considered with identical importance. The network aims to provide a cost effective road network and uses air travel in case the road network can not meet the requirements. This sums up the requirements following from the ESP.

ESP requirements:

- *The model should create a network with the lowest achievable network cost.*

- *The model should be able to start optimisation with both the existing network and just the depot location and corresponding volume distributions.*
- *The model may only route parcels via road or air transport.*

OCG The third source of requirements is the OCG. They have additional demands to make sure that the model is not only technically suited to the case, but that it can be used in a business setting as well. Their requirements focus on the type of cases the model can be used for and the time needed to analyse these cases. To suffice those needs three requirements are added. First of all, the model will be functionally designed to be able to handle an unlimited amount of nodes. This means that the structure of the model will not be restricted by a certain network size. The OCG will be able to run models covering a state, country, a continent or possibly the entire world. Although increasing model sizes might lead to technical problems on the computers used -like running out of internal memory- these technical difficulties are not considered in the design. Secondly, efforts are made to keep the model runs within an acceptable run time. What is considered acceptable is not easily quantifiable, since it depends on how users experience the run time. Therefore, users are interviewed to learn how they value the run time.

OCG requirements:

- *The model should be able to cope with any number of depots, hubs and parcels.*
- *The model should have an acceptable run time.*

The following section will classify the requirements in a format that will help structuring the design process, by using three requirement categories.

3.1.2 Requirement categories

All these model requirements influence the design process in different ways. Some influence the model logic or the structure of the algorithm(s) while others limit the possibilities that are allowed to be considered. To make this clear, the requirements are divided in three different categories: objectives, constraints and functional requirements.

Objectives goals of the model.

Objectives represent a goal of the model and are specified by a factor and a direction. For instance, in soccer the objective is to win by scoring goals. The number of goals is the factor and these need to be maximised (direction). The design can use any means that fit inside the constraints to achieve the objective as best as possible.

Constraints limitations that influence the objective.

Constraints are defined by a factor and a limit (instead of a direction). For instance, continuing the soccer example, the playing field is limited by lines; contact with the ball is limited to the bodies of the players and even the use of their bodies is limited since they may not use their arms or hands. The solution to scoring a goal may therefore not contain any strategy that involves crossing the lines of the playing field, using anything else than the players bodies or using any of their arms or hands.

Functional requirements conditions that influence the success of the design, but do not influence the objective of the model.

Functional requirements are any goals or constraints that influence the success of the design, but have no influence on the main model objective. Going back to the soccer example, foul play (e.g. tackling without playing the ball) is discouraged by the issuing of yellow and red cards. Fair play is not needed to achieve the objective (scoring goals), but it is very important to secure the health of the players. Hence, for any sport today, promoting fair play is an important functional requirement. The next sections enumerate the identified design requirement in these three categories.

Objective

The model has the following objective:

1. *The model should create a network with the lowest achievable network cost.*

Constraints

The model has to satisfy the following constraints:

1. *The model should create a network that is able to transport all offered parcels (flow conservation).*
2. *The model should create a network containing routes that can deliver each parcel before their service time expires (service commitment).*
3. *The model may only route parcels via road or air transport.*

Functional requirements

The model has to satisfy the following functional requirements:

1. *The model should be able to cope with any number of depots, hubs and parcels*[2].

[2] Technical issues may arise when increasing the number of nodes in the model. For instance, as a result of running out of memory on the host that runs the model. This is out of scope. During the design merely the functional ability to cope with any number of nodes is provided.

2. *The model should have an acceptable run time*[3].

This concludes the requirements specification. The next section describes the scope of the model design.

3.2 Scope

The goal of defining the scope is to specifically delineate the part of reality that will be modelled. Since the subject is to design a cost efficient network the scope should define what functional part of the network is included in the design. This will be set forth in the next section. Furthermore, in reality express networks only cover a certain geographical space. The geographical scoping of this research follows the network scoping. A third category is speciality scoping. It describes how the design deals with the level of detail in different parts of the model. The final section provides any additional delineations.

3.2.1 Network scope

No PUD The network scope describes the part of an express network that is investigated in this research. The general sequence in the routes of parcels is as follows (as explained in Section 1.4): the route of a parcel starts with collection at its origin, after which it is brought to a nearby depot, then it moves to a nearby hub, from where it is transported to another hub close to its destination, it then moves to a depot even closer and finally it is delivered at its destination. The model has the inter-depot travel as scope. Hence, collection at origins to get parcels to a depot and distribution from depots to the destinations is out of the scope of this model.

3.2.2 Geography

Geography Geographical scoping is subject in almost any problem definition. In this case however, the model will have no geographical boundaries since its purpose is to deal with a generalised mathematical problem. A design requirement is to have the model be able to deal with any number of depots, hubs and parcels. However, increasing the datasets will result in an increase in needed run time for the model. Although attempts are made to minimise the run time, it is not a main focus.

3.2.3 Speciality

Speciality In contrast to geographical boundaries or the network scoping in the two preceding paragraphs that define how broad the problem is modelled, specialities define the

[3] This constraint has no fixed time limit, since the acceptable time limit depends on the achievements of the model. User experiences will determine whether run time is at an acceptable level. Less run time is however always better, so in case it will not reduce model functionality or violate any other constraints effort will be made to reduce the run time.

research of the different components of the model. For instance, when dealing with non-cooperative agents that have to agree on a transaction, a straightforward heuristic could be used or research could be done in the game-theoretic field to create a well defined set of strategic mechanisms to deal with such transactions. The second option is a more in-depth researched alternative. Although it would be best to have components are researched in detail, there are two reasons to differ from such an approach. First of all, the time frame of the research does not allow for a full in-depth research of each component of the model. Second, in case it is known or highly expected that a certain component is not significantly influential to the model outcome, the effort of researching the best available solution does not outweigh the improvement of performance of the model. The description of the model design in Section 4.2 and Appendix B shows the detail in which the different components are modelled.

3.2.4 General delineations

The scope is further specified by the following delineations.

Volume distribution

- The volumes of parcels in the model are expected to represent an averaged day in an average week. More specifically, for an average week all weekdays are averaged into a single averaged day. Since changing the current network involves closing and opening hubs, which involves high cost, the network should stay in place for at least ten years to write off the investment. Therefore, the volume of this averaged day is assumed to have a representative distribution and is off set to increase the volume corresponding with the expected total amount of volume in ten years. The implication for the model design of using an off set averaged day is that it defines the time horizon of planning that needs to be computed by the model to a single route schema with an time span of one day.

Unlimited volume variety

- The model is functionally able to handle any kind of volume of parcels (in terms of distribution and total volume). However, the volume distribution is not varied during a model run.

One vehicle type

- The model assumes one vehicle type. The size and speed of this vehicle type can be set to any value, but the model is limited to a single type of vehicle. Allowing multiple vehicle types would greatly increase modelling complexity, because of the varying capacities, vehicle speeds, transport cost and repositioning cost.

Explicit transport cost

- The transport cost in the model are explicitly determined by examining the exact amount of trucks required. This provides an approximation of the transport cost that is of the most accurate kind currently applied in HLP research (Kimms, 2006).

No milk runs

- Connections between depots and hubs are always direct. It is more efficient to apply milk runs[4], but that is a tactical problem and not suited this for strategic model.

[4] See Figure 4.21 and corresponding section for an explanation of the milk run concept

Air network

- The model is aimed at computing an efficient road network. Although the existence of the air network is not ignored, it is not the aim of the model to improve the latter network. A cost penalty is assumed for air travel and in cases these cost are less than the cost of a route by road, the parcel is assigned to air transport. No further route needs to be determined in that case. However, in reality it is almost always cheaper to travel by road and parcels almost exclusively travel by air in case a direct road route won't be fast enough to make the cut-off times at the destination.

3.2.5 Significance of delineations

This section sets forth the consequence of changing individual delineations to illustrate their significance. Each general delineation is reviewed in the same order as they were mentioned in the previous section.

Volume distribution

- The volume distributions are based on an averaged day in an average week (the weekdays of an average week are averaged into a single day). It could be argued that not an average day, but an extremely busy day with very high volumes should be used, because the network should be able to deal with such volumes. The key lies in what saves the most cost. There are many days that are near the average and relatively few that are near an extreme day. Hence, this way the network is optimised to have low cost for as most days in a year as possible. Furthermore, although it is optimised for an average day, the network can handle more volume. More trucks can be hired to deal with increased transport volumes and during busy periods (for instance around Christmas) different service times are offered to deal with longer sorting times at hubs.

Unlimited volume variety

- The restriction on the volume variety is that it is not able to perform a sensitivity analysis on the used volume distributions. This is not considered due to the time frame of this research. However, it is strongly advised to perform a sensitivity analysis on the volume distribution to test the robustness of the solution. Although out of scope, sensitivity analyses can be performed by creating many different volume distribution datasets outside of the model and perform model runs on each different dataset.

One vehicle type

- The model uses one vehicle type to reduce complexity. Adding another vehicle type would complicate checking whether a vehicle is full, because the capacity may differ. Furthermore, a double driving time matrix is needed since the vehicle type may have different speeds. In addition, different vehicle types have different cost. Lastly, there would now be two vehicle types that need to be repositioned to have the right number of vehicles at each node at the end of day. These are only a few of the consequences of using one additional vehicle type. Adding even more vehicle types would increase the decision complexity exponentially.

Explicit transport cost

- The transport cost is determined as detailed as possible. A higher level approach would be to use a discount factor for transport between hubs, building on the assumption that trucks driving between hubs are fuller because of higher transport flows. Determining the explicit number of trucks would provide more security on the exact transport cost. Furthermore, it provides the possibility for detailed cost savings by trying to reduce the number of trucks by re-routing the volume.

No milk runs

- Milk runs would probably greatly enhance the solution. It is expected that major cost reductions can be achieved by use of milk runs. However, it eliminates flexibility in the transport between depots and hubs since it couples the transport of one depot to a hub to the transports of multiple other depots to that hub. A change in only one of those will effect all others. Furthermore, extra detailed information on travel times and cut-off times at the depots is needed, to calculate optimal milk runs. This level of detail is more suited for tactical analysis. It is advised to use the strategic level outcome of the model in this research and then improve the routes trough a tactical study including milk run possibilities.

Air network

- Including the air network in the scope of the model would increase its complexity. However, if feasible, it would enhance the reliability of the cost of transport by air. Currently, the model assumes the same cost penalty for all parcels unrelated to their origin or destination. Since travelling by air is in general much more expensive than travelling by road, this does not negatively impact the network configuration. On the other hand, the total network cost would be more realistic when actual induced cost are calculated for each air transported parcel specifically.

3.3 Chapter conclusion

The third research question, '*What are the requirements of the design?*', is answered in this research with use of a brief actor analysis. Based on the goals and needs of express service providers, their customers and the OCG, the objective, constraints and functional requirements for the UMAHLP were identified; the model design should comply to these. The next chapter elaborates in detail on the model design.

Chapter 4

Model design

With the model objective, its constraints and functional requirements established, the model can be designed. This chapter describes this model design process. It starts with specifying the inputs and outputs of the model. Then a quick insight in the layout of the model is given with a high level description. Finally, the last section elaborates in detail on the design of the model.

4.1 System diagram

The system diagram (Figure 4.1) shows the model boundary in context with the inputs and outputs of the model. First these inputs and outputs are described, followed by a general description of the model setup.

4.1.1 Inputs and outputs

The input is divided into two categories: Environment and Instruments. Environment data is input which cannot be influenced by the user. Variables that can be influenced are instruments. The variables of interest form the output generated by the model, they encompass the technical information that helps ESP to create an efficient network.

Environment The environment variables are inputs that are considered fixed during a study. For instance, an ESP gives the assignment to analyse a certain country. The current depot locations of that country, their volume distribution, and the driving time between them are the fixed inputs of the model. Changing these three parameters would detract the problem from reality. In other words, they are directly derived from the environment. Additional optional inputs are hub locations and routes of the current network. Providing either the hub locations alone or both the hub locations and routes will skip the first or both the first and second phase in the model respectively. These phases are described in the next section (Section 4.1.2).

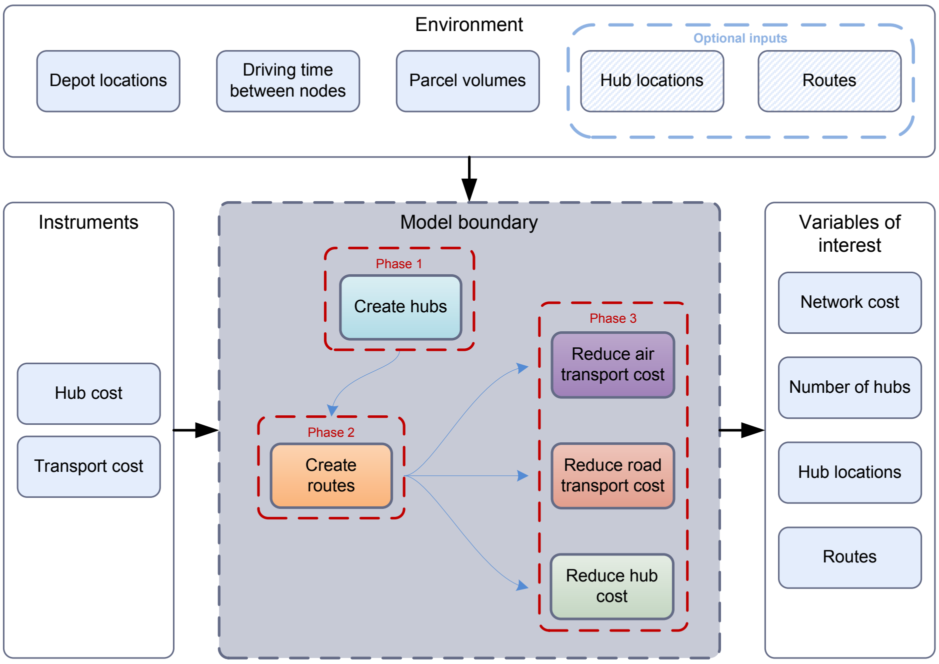


Figure 4.1: System diagram

Instruments In contrast, instruments are inputs that the user can change to influence the model results. The two instruments are the hub cost[1] and transport cost[2]. Their ratio directly influences the ratio of hubs versus travel distance in the network that the model will create. Increasing hub cost will favour having less hub touches (the average number of times a parcel flows through a hub) and therefore leads to less consolidation, because it becomes cheaper to use more trucks than to go via an extra hub. Increasing transport cost will favour hubs over trucks, so the balance between hubs and trucks will go more towards having less trucks. This result stems from the fact that more hub touches allows for better consolidation, leading to fuller trucks, hence less trucks are needed. It is the responsibility of the model user to find a good ratio between the hub cost and transport cost, using a combination of experience and the model outputs.

Variables of interest The outputs of the model are depicted as the variables of interest in Figure 4.1. The most important variable is the network cost, since it is the main objective of the model to minimize this variable. The number of hubs is a main characteristic of the type of network determined by the model. The hub locations and the routes together define the network configuration established by the model. Together they define a cost effective network configuration that will answer to the needs of the ESP.

[1] Fixed and variable.

[2] Assumed to only have variable cost. The variable cost are fixed per km, so influences like increasing fuel cost with increasing load is considered to have too little influence. An average fuel cost per km is therefore incorporated in the variable transport cost.

4.1.2 General model description

The model itself is divided into three separate phases, as depicted within the model boundaries in Figure 4.1. During the first phase, the depot locations, driving times and parcel volumes are used to setup a rough configuration of the hub locations. The second phase uses this configuration as input together with the hub cost and transport cost to determine the routes that the parcels will be transported by. Although the first and second phase are designed to make smart decisions on hubs and routes, it is unlikely that they will result in an optimal configuration at once. Since hub locations influence the possible routes and resulting routes provide clues about better hub configuration, the best decision can be made when information is available about both. Hence, the greatest opportunities lie after the first two phases. In the third phase three separate sources of cost are targeted to improve the network, by changing routes and closing and opening hubs. Figure 4.2 provides a high level technical overview of the sequence of steps in the model[3]. The three main phases it consists of are briefly described below.

```
        do
Phase 1 Create hubs;
Phase 2 Create routes;
Phase 3 while stop condition ≠ true;
                Reduce air transport cost;
                Reduce road transport cost;
                Reduce hub cost;
        end
```

Figure 4.2: Model algorithm

Phase 1 During the first phase, hub locations will be determined based on the volume distributions of the depots. The major assumption being that if there is a lot of volume to be transported between two regions, it is beneficial to have a hub in the centre of both regions. This allows for consolidation in the first region, distribution in the second and vice versa. This will result in full trucks driving between these hubs. The process is as follows. First, each depot will create a region around itself as central depot, where all other depots within a certain driving time are members of that region. It will collect the volume distributions of all depots within its region. Using these volume distributions, the volume back and forth between all regions can be determined. Now the region combination with the highest volume to be transported is selected and the central depots of both regions are upgraded to a hub. Next, all region combinations with a central depot close to this/these primary hub(s) are eliminated from the list. This process of choosing a region combination, creating the hubs and eliminating unsuited region combinations will iterate until no region combinations are left. This results in a network configuration with depots and hubs, that is ready to have routes fitted to it.

Phase 2 The routes are created in a precise and detailed way as described in the next Section

[3] This algorithm does not define a stop condition. Since it is impossible to know whether the optimal solution is reached, the stop condition cannot be defined by cost. Most probably it will be defined by a time period, but that should follow from thorough testing after full implementation of the model.

4.2. The high level process described here will however give an indication of this phase. At first, all depots subscribe themselves to the (e.g.) three hubs closest to them. This way the hubs gain knowledge on the depots that can be reached directly from them. Next, the depots group their parcels by destination. A dispatch request is created for each group. These requests contain all the details about the parcels necessary for the hubs or other depots to decide on the route this group could or should take. The dispatch requests are sent to the hubs the depot is subscribed to. Then the hub checks whether the parcels arrive at the hub before their service time. If so, it finds the hubs that the destination depot is subscribed to. It adds information on the cost and time added to handle the parcels to the dispatch request and sends the dispatch request to the just identified hubs. These hubs check the service time again and add cost and time information to the dispatch request. Then they check if the parcel can reach the destination depot directly from them within the service time. It then adds details concerning this last part of the route to the dispatch request and sends the dispatch request, now a dispatch proposal, back to the hub that it received the request from. This origin hub receives proposals of all hubs it sent the dispatch request to. It selects the cheapest feasible solution and sends that proposal back to the origin depot. The origin depot will receive a dispatch proposal from all hubs it is subscribed to and selects the cheapest option. To endorse the chosen route, it will send out a dispatch confirmation to the hub that is the first stop on the selected route. A dispatch confirmation contains the selected route and instigates the nodes on the route to update that local KPIs to incorporate the details of the confirmed parcels to be routed through that node. Other ways of determining a route, like directly transporting volume from the origin depot to the destination depot or via one or three hub touches instead of two are described in the System design (Section 4.2. The second phase ends when a route is determined for all parcels.

Phase 3 The third phase starts with a feasible network. Either using an existing network as input, or using the network that resulted from the previous phases. It will enhance the network by addressing three sources of cost: air transport cost, road transport cost and hub cost as visualised in Figure 4.2. Parcels are selected to be transported by air in case there were no routes discovered in Phase 2 that allowed the parcel (or group of parcels) to be delivered before the service time. In the great majority of cases, it is cheaper to transport the parcel by road. Therefore, a module of logic will look for parts in the network that contain groups of depots close to each other that have a relatively high percentage of parcels that need to be transported via air. It then calculates the cost of moving the closest hub a bit closer, assuming that this will reduce the travel time of the road route, in an attempt to reduce the air volume. Note that although the 'high air' depots will benefit, other depots will most probably have increased distances to that hub, invoking higher cost. In case this new configuration would lead to less overall cost, the hub will be moved. Otherwise, this option is neglected. Another way of reducing air travel, is looking for routes with only one hub touch. More about this latter option is described in the System design. The second cost source is road transport. Attempts are made to reduce this by bundling parallel sectors. An example of parallel sectors is for instance when there are routes between a hub in Amsterdam and Madrid, and between Köln and Barcelona. For a great part of the total distance, these two routes will use the same roads. Combining the loads of the two routes somewhere in Belgium and splitting them in the south of France or the north of Spain, could result in less cost by increased consolidation. These routes are only combined in case it is certain that they will lead to cost savings. The third source of cost is the hub cost. Hubs are subject to an attempt to close them in case they have either very low volume rates or when the parcels that go through the hub all have high travel slack. In the first case, the low volume indicates that the hub might be superfluous and a calculation is done to find out if the network cost drops in case it is closed. In the second case the hub does not necessarily have low volume rates, but the parcels that flow through the hub are early when considering their delivery time. Travel slack means that the parcel can be delayed for a while and still make its service. Hence, there is room to re-route them through a different hub and safely close the former hub. These three cost saving techniques are not considered to cover every possible cost saving possibility, hence further research is invited to add modules to this optimisation phase.

Twofold model Since the model first creates an initial routing solution in phases 1 and 2, after which it will start optimising in phase 3, it could be seen as a twofold model. The advantage of this separation is that it will make it easier to use the model. For instance, when the model will be used to see how the current network in a country could be improved, only the optimising part is needed. When the best solution should be found without taking legacy into account, both parts will be used. Another advantage is the modular set-up it creates. It is easier to improve, redevelop or replace a part of the optimisation strategy when it is not spread out in many parts throughout the model. The details on the three main phases in the model and how this design has taken shape is described in the next section.

4.2 System design

The research goal together with the requirements as stated in Chapter 3 define what the model should accomplish. The Prometheus methodology indicates that after defining these, the design phase can start by determining how the research goal can be achieved while meeting all design requirements. Following Prometheus this will be done with use of the system goal overview. The system goal overview is the highest level of aggregation of model design in the Prometheus methodology. It contains the entire layout of functionality in the model and therefore provides an accessible and complete view of the model design. However, the system goal overview is used differently in this research. Therefore an explanation of how Prometheus defines this stage is given first. This is followed by giving the reason for differing from this method, stating the used method, and explaining how the used method fits the Prometheus methodology. After that the entire model design is described with the altered system goal overview.

Prometheus utilises the goal overview to specify the main goals in more detail Padgham & Winikoff (2004). The starting point is a description of several sentences about what the system should be able to do. Several goals are extracted and further specified by asking the question how these goals can be accomplished. The aim of this question is twofold. Firstly, it will help identify different types of goals and secondly it will help specify known goals. As a result, one has at least as many goal trees as initial goals. Then similar goals are coalesced thus connecting their goal trees. Hence, the end result is a network of goals. In the next step the goals are ordered by grouping them into chunks of behaviour called roles[4]. This approach suits a situation in which the main goals are not all known or ambiguous. However, this research has one clear goal that the model should reach: create a feasible, cost-effective depot-hub topology for express networks. While staying close to the original Prometheus approach, asking the how question while starting at a single goal, will create a hierarchical goal tree instead of a network of goals. Secondly, it will sooner result in means instead of subgoals. This is the consequence of aiming not for the identification of different types of goals, but merely specifying the currently known goal. Hence, the used approach has more resemblance with a means-ends objectives network approach as described by Keeney (1992).

A means-ends objectives network is natural combination of goals and means in one diagram. Going from the top down, the same question is asked as in a goal tree: How can this objective be achieved? Additionally, going from the bottom upwards in the diagram should answer the question why a certain mean is used or why a goal should be achieved. This approach is more strict in the sense that the diagram should suffice to both questions. However, it allows to have both means and goals in one diagram. In most cases an object in the diagram will be both a mean and a goal, but the lower the specification level the more the objectives will be formulated as means instead of goals. The resulting tree will give a complete view of what the model will do to achieve upper level goals. The advantage of this approach is that it will make the step to roles easier as in most cases the objectives are already grouped and a goal in a higher level of the tree can be translated straight into a role. Furthermore, it poses no problem for the Prometheus Design Tool (PDT), because the structure of a goal tree and a means-ends objective network are the same. Hence, no functionality of the PDT, like consistency checking, will be jeopardised. Apart from being an efficient stepping stone to further analysis, the means-ends objective network gives is well suited to explain the contents of the model by. The next section will elaborate on the model design by use of the means-ends objective network.

4.2.1 Top level

The top level of the means-ends objectives network (hereafter referred to as 'tree') is presented in figure 4.3[5]. It presents the five modules that together accomplish the top goal of creating a feasible, cost-effective depot-hub topology for an express network. The first two modules are *Create hubs* and *Create routes* that make up Phase 1 and Phase 2 respectively. The three following cost reducing modules *Reduce air transport cost*, *Reduce road transport cost*, and *Reduce hub cost* make up the third

[4] In older literature on Prometheus roles are referred to as functionalities.

[5] Since the entire tree is too big to display as a whole, it is divided in several parts. Each block in a figure that is connected to a block in another figure is labelled to provide a better overview. For instance, in Figure 4.4 the block *Set hub windows* is labelled 3a3. The labels correspond with the figure names. Hence, the breakdown of this block can be found in Figure 4.8: Means-ends objectives network: Level 3a3.

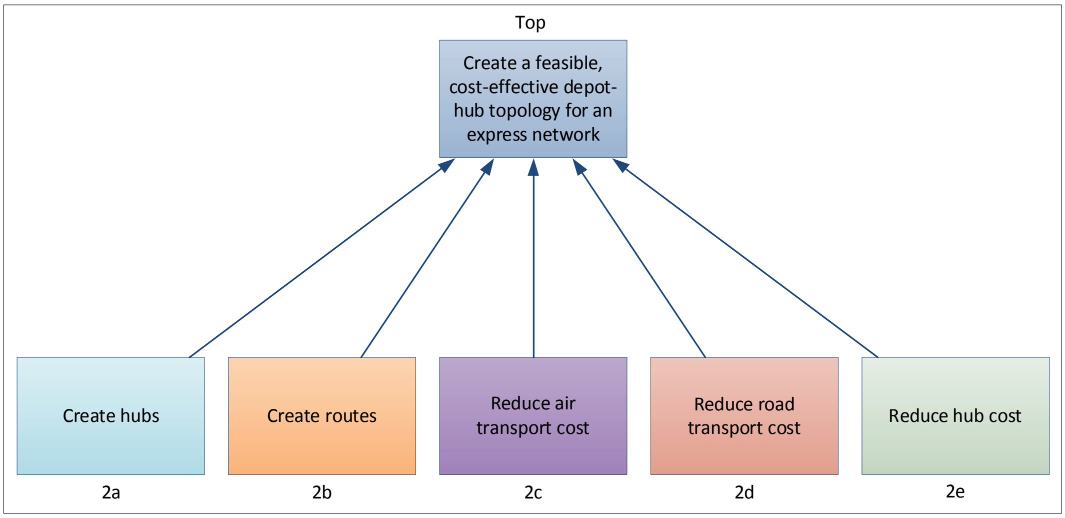


Figure 4.3: Means-ends objectives network: Top level

phase. Before describing these five modules, the goal of the model is restated in a technical sense to introduce terms that are used in the remainder of this chapter.

A network consists of two general parts: nodes and edges. A node is a point of intersection in a network. An edge is a line or segment joining two nodes. In express networks the nodes are depots (d) and hubs (h) and the edges are segments between the depots and hubs along which routes can be planned and along which vehicles will drive from node to node. So to get a network we need to determine where the nodes are and how they are connected by edges. The locations of the nodes are known in this model, since depot locations are a given and these depots are the potential hub locations as well. The model has to determine which depots to upgrade into hubs and determine via which nodes all origin depot (d^o) to destination depot (d^d) routes will go. Hence, the first step in the model is to *Create hubs*, which is described next.

4.2.2 Create hubs

There are three main factors influencing the placements of hubs. First of all, the more hubs you have, the greater the possibility for consolidation of volume. Second, the less hubs you have, the greater the distance to drive from depots to hubs. Thus more distance will be covered with non-full trucks. Third, hubs cost money. The more hubs, the greater the cost[6]. So a balance must be found between the advantages (less transport cost) and the disadvantages (increasing hub cost).

This is all part of the goal *Create hubs based on destination volume* that is the first of three means of creating hubs. The second mean focuses on secluded depots, which

[6] This is due to the economies of scale, both in variable cost and fixed cost. Variable costs depend on the volume that is handled at the hub. The more volume is handled, the bigger and more efficient sorting machines that can be bought which have a lower cost per parcel. Fixed costs increase with size, but relative to a single parcel they drop. You only need limited size of management for a single hub, which does not grow linearly with increasing volume. The same goes for the cost of the required building. It is cheaper to build one 10.000 m^2 building than two of 5.000 m^2.

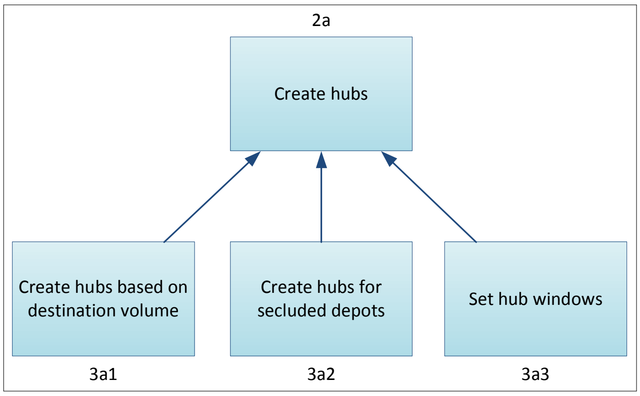


Figure 4.4: Means-ends objectives network: Level 2a

are depots that are far removed from the hubs created in the previous mean. This indicates a relatively high possibility for being forced to transport a higher share of their volume via air, since making the service time via road will be harder because it takes longer to get to the first hub. *Create hubs for secluded depots* aims to fill such gaps if left by *Create hubs based on destination volume* and can hence be looked at as a backup, or enhancement on the latter module. The third block is *Set hub windows* that is a preparation for the next phase (*Create routes*). It sets time windows within which trucks coming from the depots may arrive and within which inter-hub travel (travel between the hubs) may take place. Consequently, the end time of the hub window marks the departure time for trucks transporting parcels from the hubs to the destination depots. These three means of creating hubs are described in respective order.

Create hubs based on destination volume

Areas of depots that have a lot of parcel volume to be transported to the same destination area could benefit of having a hub nearby their collective origin (origin hub or h^o) as well as at the destination (destination hub or h^d). At the origin area the volume of the different depots can be consolidated and at the destination area the volume destined for specific depots can be combined. The overall logic tackling this challenge is visualised in Figure 4.5 and further explained in this paragraph.

As said, the origin hub - destination hub ($h^o h^d$) combinations are determined based on the need of a group of depots that are close to each other to transport a certain volume to destinations that are also close to each other (*Create hubs based on destination volume*). Region (r) is the name for a group of depots. The group of depots in region r is denoted as D_r. Each group of depots has one central depot denoted with $\bar{d}$. Depots belong to a region in case they are within a maximum allowed driving time (displayed by t^{max}) of the central depot $\bar{d}$ belonging to that region. This is summarised by the following formulas that together define the *Create a region per depot* objective:

$$d \in D \tag{4.1}$$

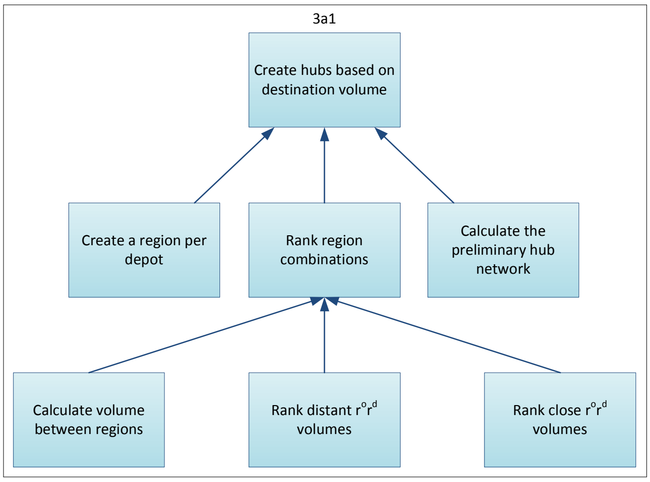


Figure 4.5: Means-ends objectives network: Level 3a1

$$r \in R \tag{4.2}$$

$$d, \bar{d} \in D_r \subset D \Longleftrightarrow t_{d\bar{d}} \leq t^{max} \qquad \forall r \in R \tag{4.3}$$

The regions are not just adjacent to each other, since this may cause inefficiencies along the borders. For instance, in case the border between two regions goes through a relatively high concentration of volume. To eliminate that problem there are as many regions as there are depots. Each depot is the central depot of one region, but can (and most probably will) be part of (many) other regions.

To be able to *Rank region combinations*, it is needed to *Calculate volume between regions*. To find the volumes that have to be transported between regions, it is of importance to define an origin and destination region (named r^o and r^d respectively). The collection of depots in r^o is denoted as D_{r^o} and the collection of depots in r^d as D_{r^d}. A single depot in D_{r^o} is denoted with d^o and in case it is the central depot of the origin region with $\bar{d}^o$. Along the same reasoning are d^d and $\bar{d}^d$ depots in D_{r^d}:

$$r^o, r^d \in R \tag{4.4}$$

$$d^o, \bar{d}^o \in D_{r^o} \subset D \tag{4.5}$$

$$d^d, \bar{d}^d \in D_{r^d} \subset D \tag{4.6}$$

For all combinations of regions $(r^o r^d)$ the total volume that flows from the depots in the origin region to the depots in the destination region plus the volume flowing in the opposite direction will be determined:

$$v_{r^o r^d} = \sum_{d^o} \sum_{d^d} (v_{d^o d^d} + v_{d^d d^o}) \tag{4.7}$$

Not all region combinations will be treated the same. One can imagine that in case two regions are far apart, there is a need for a hub in r^o for collection and a hub in r^d for distribution. However, in the case that two regions are close to each other it is more efficient to have one hub in between the two regions. These two situations are visualised in Figure 4.6.

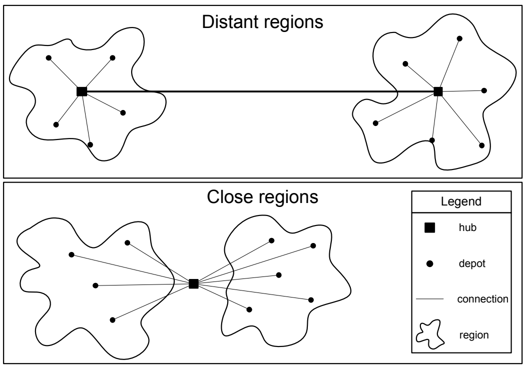


Figure 4.6: Distant regions versus close regions

The implication of this approach is that it is necessary to determine a driving time that serves as a distinction between the two categories. This driving time is measured between the central depots of the specific combination of regions. A region combination is categorised as distant in case the driving time is longer than the minimal set time t^{min}:

$$\forall r^o, r^d | t_{\bar{d}^o \bar{d}^d} > t^{min}, \quad \bar{d}^o \in D_{r^o}, \bar{d}^d \in D_{r^d} \tag{4.8}$$

A region combination is categorised as close in case the driving time is shorter than the minimal set time t^{min}:

$$\forall r^o, r^d | t_{\bar{d}^o \bar{d}^d} \leq t^{min}, \quad \bar{d}^o \in D_{r^o}, \bar{d}^d \in D_{r^d} \tag{4.9}$$

When all regions are evaluated there will be two lists of interesting region combinations ($v_{r^o r^d}$). These will be sorted on volume to complete the goals *Rank distant* $r^o r^d$ *volumes* and *Rank close* $r^o r^d$ *volumes.*

The final step is to *Calculate the preliminary hub network.* It is not trivial to calculate the preliminary hub network. The problem has similarities with the gravity model (Tinbergen, 1962), the set covering problem, and the weighted maximum set packing problem[7] (Schrijver, 2003). All are often solved with a greedy algorithm (Kwon, 2005; Chvatal, 1979). Consequently, a greedy algorithm is utilised to determine a solution. It starts with choosing the $r^o r^d$ combination that has the highest combined volume and adds the node(s) belonging to the region combination[8] to the solution. Then it eliminates all $r^o r^d$ combinations of which at least one of the corresponding

[7] The latter two are both NP-complete as proved by Karp (1972), which means that the required time to solve the problem increases very quickly as the size of the problem increases.

[8] For distant regions these nodes are the $\bar{d}^o$ and $\bar{d}^d$, for close regions this is one d centrally in between the $\bar{d}^o$ and $\bar{d}^d$.

node(s) is closer to the nodes that are in the current solution than a pre-set driving time t^{close}. Next it selects the $r^o r^d$ combination with the highest combined volume out of the set of remaining $r^o r^d$ combinations and adds the node(s) corresponding that combination to the current solution. The algorithm then iterates through these last two steps until the set of possible $r^o r^d$ combinations is empty. This means that no $r^o r^d$ combination is left of which the node(s) are either in the current solution or further away than t^{close} travel time from all nodes in the current solution. In case this solution leaves depots relatively secluded from hub access, the next step aims to make adjustments accordingly.

Create hubs for secluded depots

The second step in *Create hubs* is to *Create hubs for secluded depots*. During the first step, it is quite possible that a (group of) depot(s) is quite far away from a hub. These secluded depots might lead to high network costs if no hubs are created in their vicinity. This step is included to research this issue.

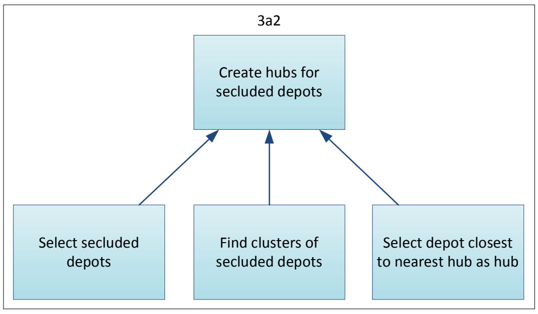


Figure 4.7: Means-ends objectives network: Level 3a2

Depots further away from a hub than a given driving time are identified as a realisation of the goal *Select secluded depots*. The secluded depots are then asked to look for nearby peers to *Find clusters of secluded depots*. In case their combined volume to transport is higher than a certain threshold, the depot in this cluster of secluded depots that is nearest to the current nearest hub, is upgraded to hub (*Select cluster depot closest to nearest hub as hub*). This can never result in the new hub being too close to an existing hub since only secluded depots are in the cluster and being secluded is based on driving time to nearest hub being higher than the before mentioned threshold. This functionality is especially helpful when there is an 'island' of depots. With a possible secluded depot problems fixed, the network is almost set for creating routes. One final prerequisite for creating routes is described in the next section.

Set hub windows

Parcels are transported from depots to hubs. At the hubs they are cross-docked, meaning that all incoming parcel shipments are re-sorted and further transported in different[9] trucks. If a parcel goes through multiple hubs this will happen twice. This process obviously needs coordination. First of all and most importantly, when there is exactly one truckload of parcels from h_a to h_b it is most cost efficient to drive one truck once. This truck cannot leave before all parcels for this transport came in at h_a and are sorted. Second of all, there should be enough time to travel between hubs and cross-dock again. To aid this process, time windows are set on the hubs[10]. The third and last step in *Create hubs* is therefore *Set hub windows*.

Hub windows determine the balance between three different types of routes:

1. the time that can be spent travelling from origin depots to hubs;
2. the time that can be spent travelling between hubs;
3. the time that can be spent travelling from hubs to destination depots.

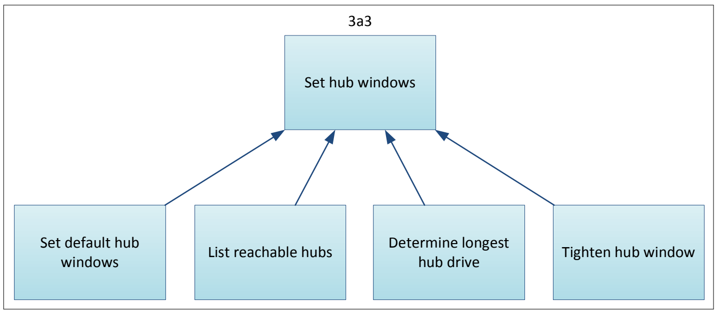


Figure 4.8: Means-ends objectives network: Level 3a3

The start of the window sets the time at which all trucks coming from origin depots must be at the hub. The end of the window sets the time at which all trucks may leave the hubs for transport to the destination depots. The resulting duration of the window sets the maximum travel time between hubs. Note that at the latter the sorting times at the origin hub and destination hub should be subtracted. The windows are set by an editable parameter and tightened where possible. *Set default hub windows* is followed by a check at each hub. Each hub is asked to *List reachable hubs*, meaning to list all hubs to which $t_{h^o h^d} \leq$ hub window duration. The travel durations of all hubs that are listed are stored in the same list. The longest driving time to the listed hubs is determined (*Determine longest driving time*) and when this time is shorter than the hub window duration, the hub window is tightened (*Tighten hub window*). The window is tightened by adding half of the duration difference to the start time of the window and subtract the other half of the end of the window. The reason for this tightening is that spare time can be used to reach further depots both at the depot-hub part as well as at the hub-depot part of the routes. It is

[9] Parcels could be transported with the same truck as with which they came in, but generally it will be a different truck. The model supports both situations described here and therefore creates no scope limitation. The phrase 'different truck' is used here to clarify the explanation of cross-docking.

[10] Note that depots have time windows as well, since all collection and sorting must be done before trucks can leave to the hubs. This is right on the edge of the scope of this model. Only the cut-off time at the depot will be used. This is the time at which the trucks may leave the depot.

distributed equally since both sides are considered equally important, there is no reason to assume one side will create more benefit than the other. When all hubs are created and time windows are set, routes can be created.

4.2.3 Create routes

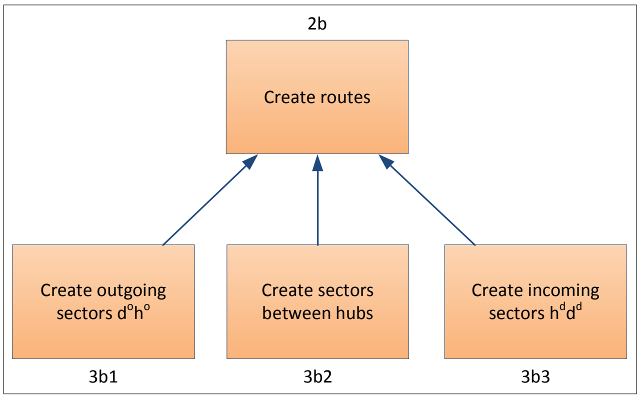


Figure 4.9: Means-ends objectives network: Level 2b

When all the nodes are set, routes can be calculated. For every origin depot (d^o) to destination depot (d^d), resulting in combination d^od^d, a route must be determined[11]. There can be three different parts of travel on a route: between the d^o and a hub, between hubs, and between a hub and d^d. This section of the model is therefore worked out corresponding to these three parts, with the goals: *Create d^oh^o sectors*, *Create sectors between hubs*, and *Create h^dd^d sectors*[12] as shown in Figure 4.9. The creation of a route for a parcel is dependent of communication from d^o via h^o and h^d to d^d and back. Most of the d^oh^o sector creation is done at the d^o, most of inter hub travel determination is done at the h^o, and the calculations on the incoming sectors is done partly at the h^d and partly at the d^d. Where possible, this is specified in the explanation below.

Create outgoing sectors d^oh^o

The creation of sectors between the origin depots and origin hubs is visualised in Figure 4.10. It starts with *Get dispatch proposals to h^o*. A dispatch proposal is a collection of data regarding the route that a group of parcels which share origin and destination, want to or will travel. At the start of the model, a dispatch request merely contains the origin, the destination and the volume of the group of parcels.

[11] An d^od^d combination exists of all volume that needs to be transported from one depot (the origin depot) to another depot (the destination depot)

[12] h^o and h^d do not necessarily need to be different physical hubs, but are named as such to give context on which part of the route is the subject. Say a certain parcel travels from d^o to d^d via hub A and hub B respectively. Then hub A is the h^o and hub B is the h^d. For another parcel which happens to travels the exact opposite way, hub B is the h^o and hub A is the h^d. The same thing applies to d^o and d^d. So h^o, h^d, d^o, and d^d are merely functional denotations and a node in the network can be any, some or all of those physically.

Based on the information in the request, the hub will decide on what action it will take. To get a dispatch request from a depot to a hub, the following must take place:

1. *Find nearest hubs.* The depot should look for a pre-set number of hubs with the least driving time.
2. *Subscribe to nearest hubs.* The depot will ask the nearest hubs for registration.
3. *Await subscription confirmation.* The depot waits until each hub has confirmed the subscription.
4. *Send subscription completion message.* The routing mechanism can only start after all depots have subscribed to their nearest hubs. This calls for some central coordination. All depots will update a central agent when they are done subscribing.
5. *Check for full truck load (at d^o)*[13]. The depot checks if their is enough volume to a certain destination to travel directly to that depot. Note that even in the case that all volume can be transported directly, the subscription to nearest hubs as described in previous steps is required, because of the routing of incoming parcels.
6. *Await go for dispatch proposals.* The depots wait until the central agent sends a signal that all depots have subscribed to their nearest depots and hence dispatch requests may now be sent by all depots.
7. *Send dispatch requests to origin hubs.* The depot creates a dispatch request for the entire volume of every $d^o d^d$ combination and sends all dispatch requests to every hub the depot is subscribed to.

When all dispatch requests are out to the hubs, the depot can wait for the responses. *Handle responses of h^o* is next and consists of two parts. First the depot should *Select proposal per dispatch request.* Every h^o corresponds one proposal to the d^o (stored in the dispatch request), which then chooses the one involving the lowest cost. Once a decision is made, the depot will *Send confirmation to next node (from d^o).* This goal is set up generally since later it will show that hubs will do the same. The goal contains working out what the next node on the route is, followed by sending the confirmation on to that node. This sequence will repeat itself until all nodes involved are updated. This way it is known at every node what the entire route of the parcels is. The main advantage is the possibility to identify possible improvements at every node. Second, when for example at an h^o a dispatch request comes in from a depot and the request is for parcels that go to a destination for which a route was already confirmed earlier, then the hub can propose the same route to the d^o immediately. This can potentially save a lot of calculation time.

As soon as all dispatch request are handled and confirmations on each chosen proposal are sent out to the h^o's, the key performance indicators (KPIs)[14] for the sector between

[13] Check for full truck load sits between *Send subscription completion message* and *Await go for dispatch proposals*, because this check can be done in parallel of waiting for all depots to have finished subscribing to their nearest hubs.

[14] Like total volume, number of required trucks, cost per kg per km, etc.

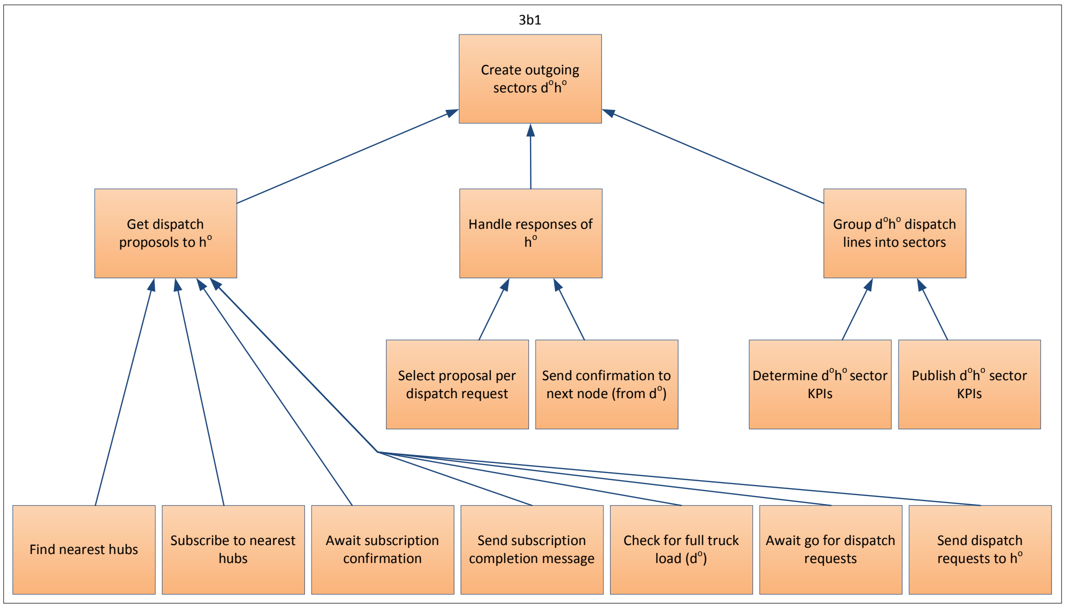


Figure 4.10: Means-ends objectives network: Level 3b1

the d^o of subject and its h^o's can be determined. This is part of the goal *Group d^oh^o dispatch lines into sectors*. The part of a route belonging to a single dispatch request is called a dispatch line. For transport between nodes and corresponding KPIs it is more interesting to look at the combined volume between nodes. Therefore all these dispatch lines need to be added to each other to form a group. Such a group containing all the dispatch lines between two specific nodes is called a sector. Hence, the volume corresponding with a sector is all volume that needs to be transported from the origin node of the sector to the destination node of the sector. Thus, a sector has a direction. To establish the KPIs, they will be first calculated by the corresponding d^o (*Determine sector KPIs*) which afterwards will publish these KPIs to the destination node belonging to each sector (*Publish d^oh^o sector KPIs)*. This concludes the *Create d^oh^o sectors*. Next, the inter hub travel is described.

Create sectors between hubs

Create sectors between hubs (Figure 4.11) is broken down into four sub goals: *Handle dispatch request of d^o*, *Handle dispatch request of h^o*, *Handle dispatch confirmation*, and *Group h^oh^d dispatch lines into sectors*. These will be described in respective order.

All actions belonging to the goal *Handle dispatch request of d^o* take place at h^o's. This goal is further specified in Figure 4.12 and consists of the following goals:

1. *Check direct driving time*. In case driving directly from the h^o to the d^d already takes more time than allowed by the cut-off time at the d^d, the dispatch request is rejected to go via the h^o and air travel is proposed.

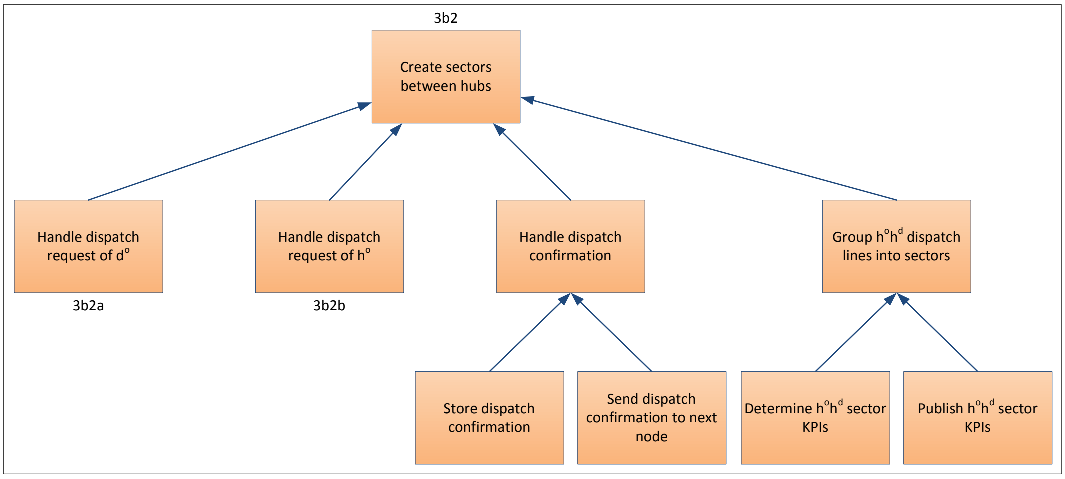


Figure 4.11: Means-ends objectives network: Level 3b2

2. *Check d^d within SL*[15]. It is very much possible that the d^o and d^d are close to each other and only one hub touch is needed. Otherwise stated, there is no expected cost benefit to pass by another hub since the possible extra consolidation is not considered to weigh out the additional transport cost. So if the d^d is in the subscription list of the hub, which indicates the d^d is relatively close by, the cost will be calculated and the $d^oh^od^d$ route will be proposed.

3. *Check for parcels with same destination.* Similar to the check at the depot. The hub will check whether there are other parcels for the same destination. If found, the route of the already confirmed parcels will be advised as route for the pending dispatch request.

4. *Get hub-hub dispatch proposals.* This concerns correspondence with other hubs to get to a proposal and is worked out below this enumeration.

5. *Select cheapest hub-hub connection.* Of all returned proposals the cheapest will be selected, but first the costs need to be calculated. This is done in two steps:

   (a) *Determine h^oh^d costs.* Local data[16] will be checked to see if the costs of this h^oh^d connection are already known[17]. If not, the cost of that connection will be calculated and stored in the dispatch request as well as locally.

   (b) *Determine total h^od^d costs.* The h^oh^d costs are added to the h^dd^d costs that are already in the dispatch request[18]. The dispatch request with the lowest h^od^d cost is selected. In case the outcome of *Select cheapest hub-hub connection* was that the parcels should go directly from the h^o to the d^d, these costs are determined and stored in the dispatch request

[15] SL = Subscription List.
[16] Meaning data at this node.
[17] This is done for every proposal that did not suggest air travel.
[18] See Figure 4.14 and corresponding explanation to find out how the h^dd^d costs came into the dispatch request.

and this communicated back to the d^o. In case the route contains an intermediate hub as well (so the route is $d^o h^o h^{im} h^d d^d$) this will already be in the dispatch proposal that came back and can be added to the costs of the first part of the route.

6. *Send dispatch proposal to d^o.* The selected dispatch request is proposed to the d^o.

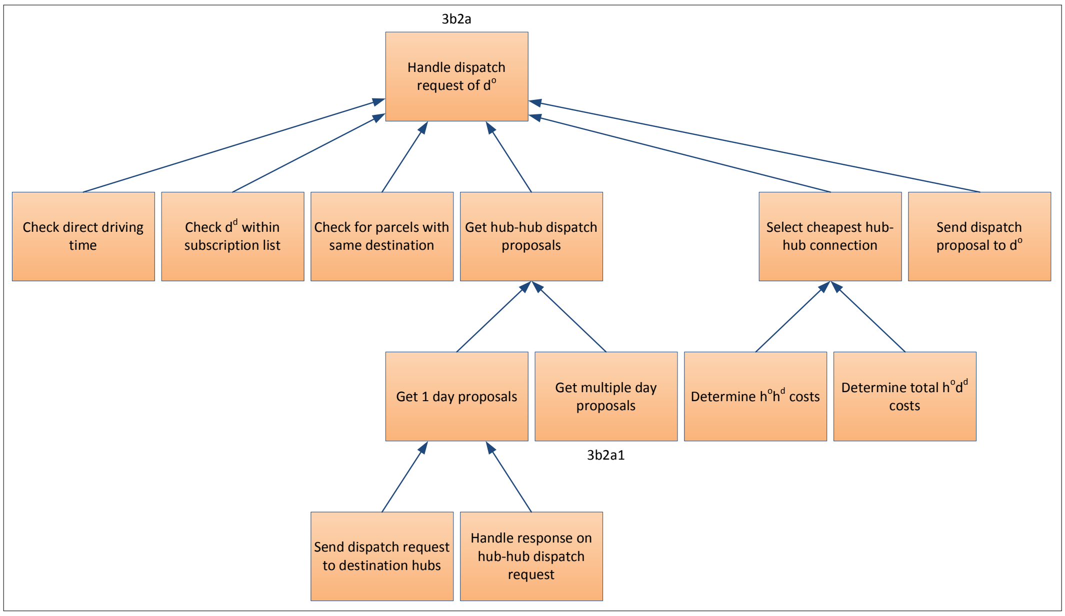


Figure 4.12: Means-ends objectives network: Level 3b2a

Get hub-hub dispatch proposals makes a distinction between dispatch requests based on their corresponding service requirement. Parcels that need to be transferred in one day are of the *Get 1 day proposals* category. They are sent through to the h^d's and their responses are handled (*Send dispatch request to h^d* and *Handle response on hub-hub dispatch request* respectively). In case a responded proposal suggests air travel, that response is overwritten with the advice to travel directly from the h^o to the d^d. This is always possible, because of the *Check direct driving time* step. Parcels that do not have the 1 day service requirements but instead have a multiple day service requirement fall in the *Get multiple day proposals* category, which is described below and displayed in Figure 4.13.

Since a multiple day service requirement allows for more driving time, the parcels in this category are handled differently (Figure 4.13). They have the possibility to touch an extra hub, which could potentially lead to fuller trucks and lower transport cost[19]. This extra hub is called an intermediate hub (h^{im}) and is selected as follows.

First the h^o determines all possible h^{im}'s for this dispatch request (*List all possible h^{im}'s*). These are hubs that suffice the following conditions:

[19] Which would mean that the cost savings of having fuller trucks outweigh the higher cost as a consequence of the longer distance that the parcels need to travel.

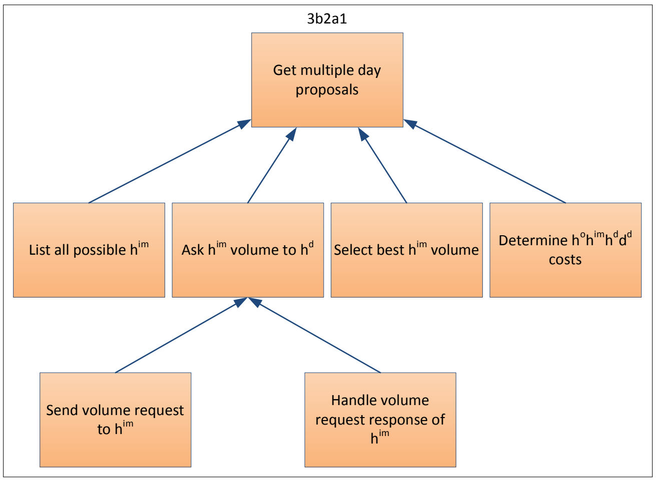


Figure 4.13: Means-ends objectives network: Level 3b2a1

1. $t_{h^o h^{im}} + t_{h^{im} h^d} \leq t^{max\ detour}$
2. $t_{h^o h^{im}} \leq t_{h^o h^d}$
3. $t_{h^{im} h^d} \leq t_{h^o h^d}$
4. $h^{im} \neq h^o, h^d$

The first condition provides the possibility to limit the resulting detour time. The second condition states that the driving time from the h^o to the intermediate hub should be less than the driving time from the h^o to the h^d. The third condition is similar to the second, but applies to the $h^{im}h^d$ part. Together Conditions 2 and 3 make sure that all potential h^{im}'s are in between the h^o and h^d. The last condition makes sure that the h^o and h^d are not marked as h^{im}'s for this dispatch request. All hubs that satisfy these conditions are put on the list of potential h^{im}'s in the corresponding dispatch request.

Next the h^o will ask all potential h^{im}'s for their volume to the h^d corresponding with the dispatch request for which an h^{im} is wanted (*Send volume request to* h^{im} and *Handle volume request response of* h^{im} respectively). The h^o will then *Select best* h^{im} based on the best combined volume. It is considered best when the combined volume of the h^o and h^{im} to the h^d is largest[20]. The last step that remains is to *Determine* $h^o h^{im} h^d d^d$ *cost* and store this in the dispatch request. This concludes the one day proposals (Figure 4.12) and multi-day proposals (Figure 4.13). What follows are the last two goals of Figure 4.12.

With all the one day and multi-day proposal costs determined, the cheapest of those proposals for each dispatch request can be selected (*Select cheapest hub-hub connection*) followed by *Send dispatch proposal to* d^o. This completes the description of *Handle dispatch request of* d^o (see Figure 4.11). Next is the breakdown of *Handle*

[20] It does not look at cost savings, since almost certainly not all dispatch requests are handled yet. Therefore going with the biggest flow is optimal for the time being and intermediate hub flows can be reconsidered later.

dispatch request of h^o of which all procedures take place at the h^d's, following the dispatch request sent from the h^o as just described.

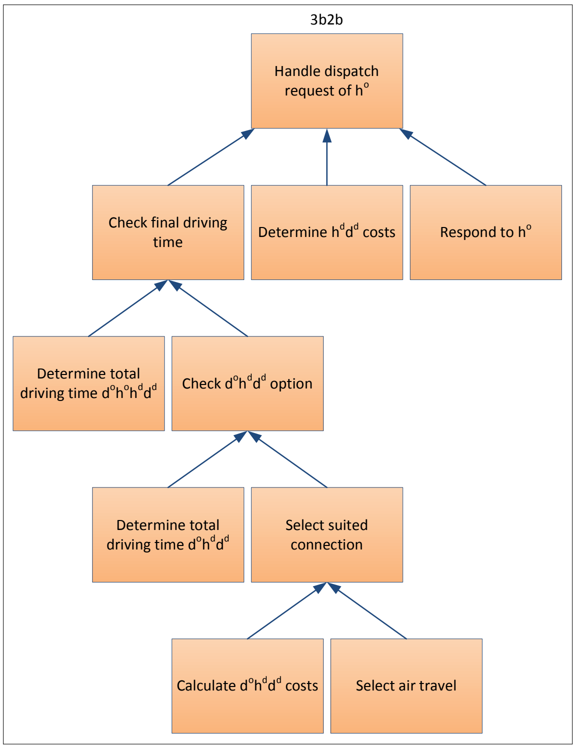


Figure 4.14: Means-ends objectives network: Level 3b2b

When a request of an h^o is received by a h^d, the hub will *Check final driving time* (see Figure 4.14). First by *Determine total driving time* $d^oh^oh^dd^d$. If this is an achievable route, then the *Check final driving time* goal is achieved. And in case this total driving time exceeds the service requirement of the dispatch request an alternative must be found, so the hub will *Check* $d^oh^dd^d$ *option*. Meaning, it will check a one hub touch option, going straight from the d^o to the h^d followed by stretch to the d^d (*Determine total driving time* $d^oh^dd^d$). When this route is achievable the cost of transport from the d^o to the h^d will be calculated (*Calculate* d^oh^d *costs*). In case that route is unachievable as well, the hub will *Select air travel*. Once it is clear what route will be proposed, the cost for the last stretch of the route will be determined *Determine* h^dd^d *cost*[21]. Finally, the proposal is responded to the requesting h^o.

Going back to Figure 4.11, *Handle dispatch request of* d^o and *Handle dispatch request of* h^o are now explained. Besides handling dispatch requests, the hubs also have to be capable to handle dispatch confirmations and determine KPIs of inter hub connections. These goal are achieved as follows:

1. *Handle dispatch confirmation*. This will happen at h^o's, h^d's and d^d's.

[21] Unless air travel is the only option. The cost of air travel will be determined by the d^o. Since the d^o is the only one with knowledge of all possible routes, once it has received all proposal it only has to calculate the air travel cost in case no other route is feasible.

(a) *Store dispatch confirmation.* Store the dispatch confirmation and update local variable (e.g. total volume that passes through hub).

(b) *Send dispatch confirmation to next node.* In case this takes place at a d^d, it will affirm the confirmation to the corresponding d^o.

2. *Group $h^o h^d$ dispatch lines into sectors.*

(a) *Determine $h^o h^d$ sector KPIs.* This is only done by h^o's and is triggered by the signal of the d^o's in its subscription list that they are done with determining routes for their volume. Hence, only changes made for *Reduce total cost of the network* (Figure 4.3) of the routes and request for flows where the specific hub will act as intermediate hub (see Figure 4.13), will create the need to recalculate the sector KPIs.

(b) *Publish $h^o h^d$ sector KPIs.* Publish the KPIs to the h^d of the concerning sector.

This concludes *Create sectors between hubs.*

Create ingoing sectors $h^d d^d$

The final part of the creation of routes is *Create $h^d d^d$ sectors* as visualised in Figure 4.15. This takes place at the h^d's at the moment that all hubs have flagged that the depots in their subscription list have finished with determining routes for all their parcels. The same process occurs here as it does in Figure 4.10. Since first the $h^d d^d$ sector KPIs are calculated after which they are stored locally and published to the h^o's[22].

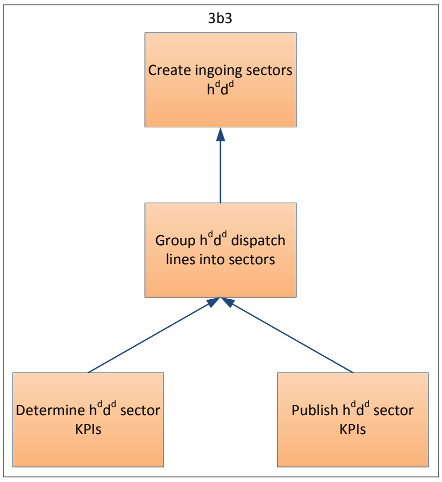


Figure 4.15: Means-ends objectives network: Level 3b3

This concludes the first two phases of the model. Hubs are created and routes are determined for all parcels. The output of this section is therefore a network that is

[22] Based on the dispatch confirmation, the h^d is able to publish the KPIs of each sector to only the h^o's via which the routes go that make use of this specific sector

able to transport parcels within the requirements. However, chances are that many improvements could be made to reduce the cost of the network while still satisfying all requirements. This is the focus of the next three sections.

4.2.4 Reduce air transport cost

The model contains two mechanisms to *Reduce air transport cost* (see Figure 4.16):

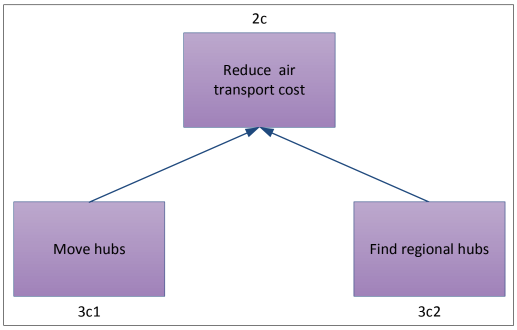


Figure 4.16: Means-ends objectives network: Level 2c

1. *Move hubs* (see Figure 4.17). A reason for a depot to have a lot of air travel is that it cannot make the cutoff time at the h^o. By moving the closest hub within the range of that depot, this obstacle for road transport disappears.
2. *Find regional hubs* (see Figure 4.19). Another way of making sure the cut-off time can be met, is using only one hub touch. Then only the end of the hub window counts as deadline to reach that specific hub, thus providing more time to make the cut-off. In the *Create routes* logic there is already support for using one hub touch although the hubs that are considered are limited to only the h^o and h^d belonging to a parcel. The regional hubs (h^r) can be any hub that will allow a parcel to make the cut-off time at that h^r and at the d^d.

Move hubs

The optimisation logic of moving hubs to get a more efficient network is visualised in Figure 4.17 and consists of the following steps:

1. *Check hub coverage*. If only a few other hubs are close by, a possible move of the hub h^m would have a higher chance of secondary inefficiency. Obviously, moving a hub does not only affect the depots that could not reach it before, but also affects the depots that currently have routes passing through the hub that is moved. A move most probably causes an increase in travel time for parcels at a number of depots. If this increase is too big these parcels will not make the cut-off time any more. The more hubs that were in the vicinity of the moved hub, the higher the chance that these disadvantaged depots can find

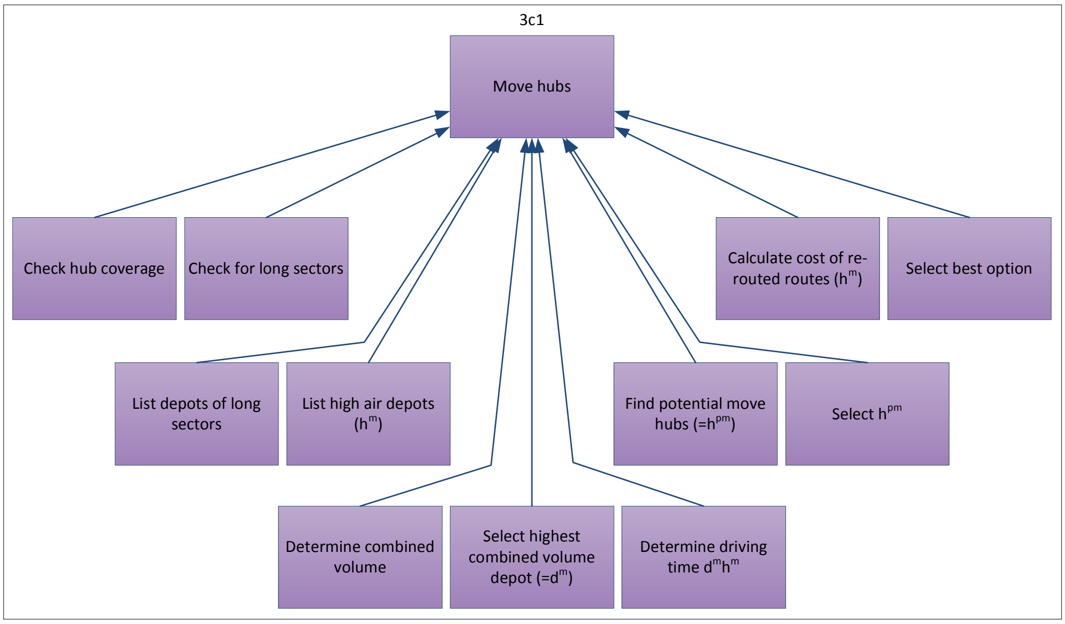


Figure 4.17: Means-ends objectives network: Level 3c1

suitable alternative routes that pass through those other hubs. The number of hubs in the vicinity of the h^m is given by $y_{h^m h_j}$.

$$y_{h^m} = \sum_{h_j} y_{h^m h_j} \quad \begin{cases} y_{h^m h_j} = 0 & \text{if } t_{h^m h_j} > t^{max}, \quad h^m \neq h_j \\ y_{h^m h_j} = 1 & \text{if } t_{h^m h_j} \leq t^{max}, \quad h^m \neq h_j \end{cases} \tag{4.10}$$

2. *Check for long sectors.* Long sectors between the hub and the depots it is connected to, indicate the direction in which the hub should be moved. Any sector of a length more than the set threshold is indicated as long.

3. *List depots of long sectors.* The depots that belong to the long sectors will be nominated to check whether the hub should be moved towards them. These depots are denoted with D_{ls}. A depot that is connected to a hub h_i is denoted with d^{h_i}.

$$D_{ls} \subset D \tag{4.11}$$

$$\forall d^{h_i} | t_{h^m d^{h_i}} > t^{ls} \tag{4.12}$$

4. *List high air depots (h^m).* All depots that have high air volume and are within a minimal driving time of the h^m. Moving the nominated hub towards these could result in less air. The set of depots with a lot of parcels routed by air is denoted with D_{air}.

$$D_{air} \subset D \tag{4.13}$$

$$\forall d^o | v_{air} > v^{min}, t_{h^m d^o} > t^{min} \tag{4.14}$$

The two combined (D_{ls} and D_{air}) leads to the list with all depots that will be asked to determine the combined volume in the region of which they are the central depot. This set of depots is denoted with D_{h^m}.

$$D_{h^m} = D_{ls} \cup D_{air} \tag{4.15}$$

5. *Determine combined volume.* Each depot in the list will ask all other depots within a certain driving time (which marks the region r^m) to respond their air volume and the volume of their sector to the h^m in case that sector to the h^m is longer than a certain other driving time. These two volumes arise from the two different types of cost savings that can be realised by moving a hub. First, moving a hub could result in a depot making the cut-off time of that hub, which enables it to route via that hub instead of air. Second, consolidation takes place at hubs. Hence, trucks that travel between hubs are expected to be fuller on average than the trucks between depots and hubs. Reducing the sector lengths between depots and hubs could potentially save transport cost. Not only because of mileage, but because fuller trucks result in less €/kg/km. The set of depots belonging to the region r^m is denoted with D_{r^m} and the depots in that set with d^{r^m}.

$$d^{r^m} \in D_{r^m} \subset D \tag{4.16}$$

$$v_{r^m h^m} = \sum_{d}^{r^m} v_{d^{r^m} h^m} \tag{4.17}$$

6. *Select highest combined volume depot (d^m).* Once the list of depots is populated with the volumes of the cluster surrounding each depot, the depot with the highest cluster volume is selected. In case the h^m will be moved, it will be moved in the direction of the selected depot (marked as d^m). This is the central depot ($\bar{d}^{r^m}$)of the region r^m of which the depots D_{r^m} have the highest combined volume:

$$\bar{d}^{r^m} \in D_{r^m} \subset D \tag{4.18}$$

$$d^m = \bar{d}^{r^m} | \max(v_{r^m h^m}) \tag{4.19}$$

7. *Determine driving time $d^m h^m$.* Moving a hub consists of closing the hub of subject and opening another. So other potential hub locations should be found in the general direction of the d^m. This means that they should be near the line between the h^m and the d^m. The total driving time $t_{d^m h^m}$ is determined and used to define an area to locate potential hubs (h^{pm}).

8. *Find potential move hubs (h^{pm}).* The potential hubs (h^{pm}) are all nodes located at most $k^1 * t_{d^m h^m}$ from d^m and at most $k^2 * t_{d^m h^m}$ from h^m. The area in which these nodes are located is hatched in Figure 4.18.

$$\forall d_j | t_{d^m d_j} < k^1 * t_{d^m h^m}, t_{h^m d_j} < k^2 * t_{d^m h^m} \tag{4.20}$$

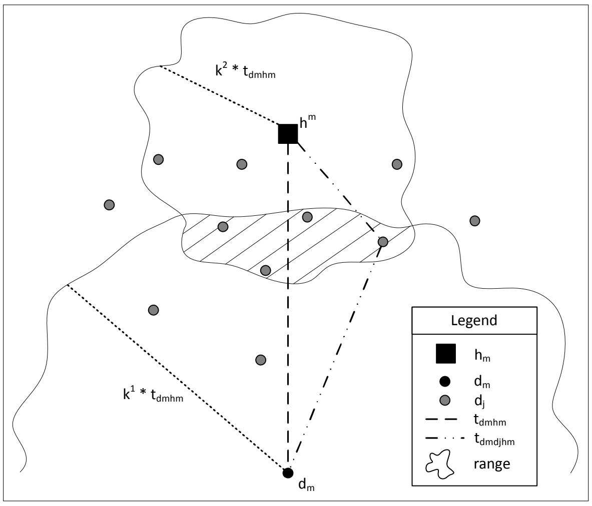


Figure 4.18: Move hubs

9. *Select h^{pm}.* Choose the three depots that have the minimal driving time $t_{d^m h^{pm} h^m}$, to limit the amount of possibilities[23] and to get the depot as close to the $d^m h^m$ line as possible.

10. *Calculate costs of re-routed routes* . Calculate the cost difference of re-routing all routes of all depots involved. These are the depots with long sectors and depots with high air volume (so D_{h^m}). Plus all routes of that used to go through the moved hub and come from depots not on the list. This means that for the depots on the list, all routes are redone and for depots of which a part of the parcels would go to the h^m and are not on the list, only the routes that were going through the h^m are recalculated. It is assumed that it is not beneficial to recalculate all of their routes, since they already found better options than going through the h^m before.

11. *Select best option.* The option with the least total amount of cost is chosen. This means either leaving the h^m and effectively making no change, or choosing to close the h^m and open one of the h^{pm}.

Find regional hubs

The optimisation logic of finding regional hubs to reduce the air volume is visualised in Figure 4.19 and consists of the following steps:

1. *Locate all possible h^r.* The potential regional hubs are all hubs that can be visited while still making the cut-off at the destination. More specifically, the travel times plus the processing time at the hub is less than the available time to make the service commitment(t^s). The processing time at the hub is denoted with t^p_h.

$$\forall h^r | t_{d^o h^r} + t^p_{h^r} + t_{h^r d^d} \leq t^s, \quad h^r \neq h^o, h^r \neq h^d \tag{4.21}$$

[23] In respect to the calculation time

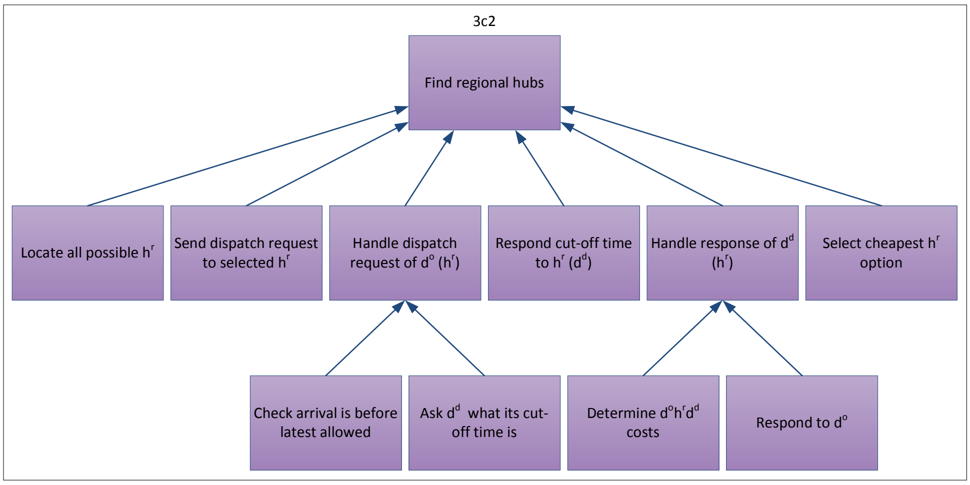


Figure 4.19: Means-ends objectives network: Level 3c2

2. *Send dispatch request to selected* h^r (action at d^o). Once the potential regional hubs are known, the d^o in question sends a dispatch request to each h^r.

3. *Handle dispatch request of* d^o *(action at* h^r*).* Two checks are done by the regional hub upon receiving a dispatch request for one hub touch transport:

   (a) *Check arrival is before latest allowed.* The h^r has, as any other hub, a hub window. It will check whether the parcel can make the latest allowed arrival time.

   (b) *Ask* d^d *what its cut-off time is.* If the hub window at the h^r does not cause a problem, the d^d corresponding to the dispatch request is asked for its cut-off time[24].

4. *Respond cut-off time to* h^r *(action at* d^d*).* The d^d will respond its cut-off times to the h^r[25].

5. *Handle response of* d^d *(action at* h^r*).* In case the cut-off time at the d^d poses no problems as well the following will be done at the h^r.

   (a) *Determine* $d^o h^r d^d$ *costs.* Calculate the transport and processing cost of the entire route.

   (b) *Respond to* d^o. Respond these cost to the d^o.

6. *Select cheapest* h^r *option* (action at d^o). Out of the responses of all h^r the cheapest is selected and a confirmation is sent to the corresponding h^r.

This concludes the *Reduce air transport cost* section (see Figure 4.16). Next, *Reduce road transport cost* will be described (see Figure 4.3).

[24] Unless the cut-off time is already known at the h^r. In which case no correspondence is needed.
[25] The d^d will respond all its cut-off times, so for each service. This way, if another request comes in that requests a route to the same d^d for the same or another service, no communication is needed.

4.2.5 Reduce road transport cost

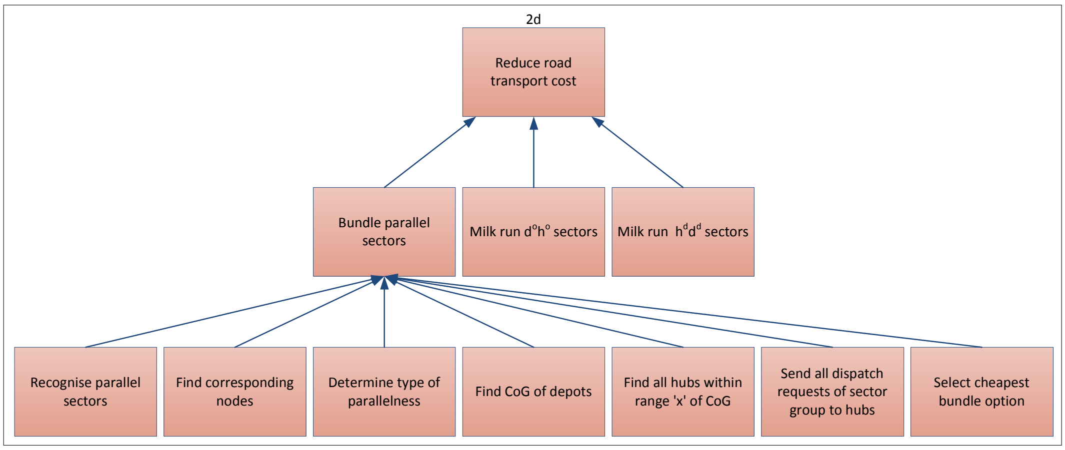


Figure 4.20: Means-ends objectives network: Level 2d

The only way to reduce cost per km is to influence the load factor of the trucks. Since the absolute cost of a truck for a given distance is the same when being either full or empty, the costs relative to the transported load are reduced with increasing weight. Three ways to get *less cost per km* by increasing the load factor of the used trucks are possible (see Figure 4.20). These will be described in respective order:

1. *Bundle parallel sectors.* Focuses on longer sectors, which can be d^oh^o, h^oh^d, or h^dd^d.
2. *Milk run d^oh^o sectors.* Focuses on sectors between depots and hubs.
3. *Milk run h^dd^d sectors.* Focuses on sectors between hubs and depots.

Bundle parallel sectors

If the inter hub travel is a very long sector say from Spain to the Netherlands, one can imagine that with the techniques in the model as described before, the h^o would be in Spain and the h^d in the Netherlands. This makes sense. However, there are cases in which a better solution could be found. Say for instance, there are many transports from Spain to the Netherlands. These transports originate in many locations in Spain and move to one or more locations in the Netherlands. Most probably, a lot of the trucks going from Spain to the Netherlands are travelling along the same highways in France and Belgium. It might be beneficial to combine these parallel sectors, combine the freight and possibly use one or more less trucks.[26] *Bundle parallel sectors* is realised in the following way, illustrated in Figure 4.20:

1. *Recognise parallel sectors.* The first step is to identify parallel sectors. The sectors should be relatively close to each other and be parallel for a minimal

[26] Note that this example is completely illustrative and does not necessarily represent a real-life situation.

distance. They need to be close to each other to limit the detour to the extra consolidation points. Since there will always be a detour, the sectors need to be parallel for a minimal distance to outweigh the disadvantage of the detour with the advantage of the extra consolidation. When parallel sectors are recognised they are grouped.

2. *Find corresponding nodes.* To be able to make an attempt to change the current routes by combining a group of sectors, communication with the related hubs and depots should take place. Hence, the next step is to identify the nodes at the ends of all sectors.

3. *Determine type of parallelness.* There are two types of parallelness, either the sectors go from multiple nodes on one side to a single node on the other, or they go from multiple nodes on one side to multiple nodes on the other. The model is able to improve both situations and treats them slightly different. In the multiple-to-one case an attempt is made to insert a hub into all sectors as close to the multiple nodes as possible. In the multiple-to-multiple case an attempt is made to insert a hub into all sectors on both sides of the sectors.

4. *Find CoG*[27] *of depots.* To consolidate as much volume as soon as possible the best place to join the sectors is close to the centre of gravity (weighing volume) of the group of nodes, so first this centre has to be determined.

5. *Find all hubs within range 'x' of CoG.* To find the best (potential) hub all nodes within a certain range of the CoG are selected. The possibility exists that this selection includes one or more already open hubs. In case this is true, all other nodes are not considered any more. The already open hub is chosen to have the consolidation take place, or when multiple open hubs remain these will be selected for the next step. In case no open hubs are amongst the selection the one closest to the CoG is selected.

6. *Send all dispatch requests of sector group to hubs.* Regardless of how many nodes remain as possible location for bundling the sectors, the process in this step is the same. Each remaining location receives all dispatch requests related to the group of sectors. They will check the possibility to route all volume and calculate the cost of doing so.

7. *Select cheapest bundle option.* The last step is to select the cheapest option. This is either choosing the cheapest extra existing hub, extra new hub or not bundling at all.

Milk run sectors

Both *Milk run* $d^o h^o$ *sectors* and *Milk run* $h^d d^d$ *sectors* are not incorporated in the model. First an explanation of the milk run concept is given after which the reason for not incorporating milk runs in the model is given. Figure 4.21 illustrates the milk run concept.

[27] CoG = Centre of Gravity.

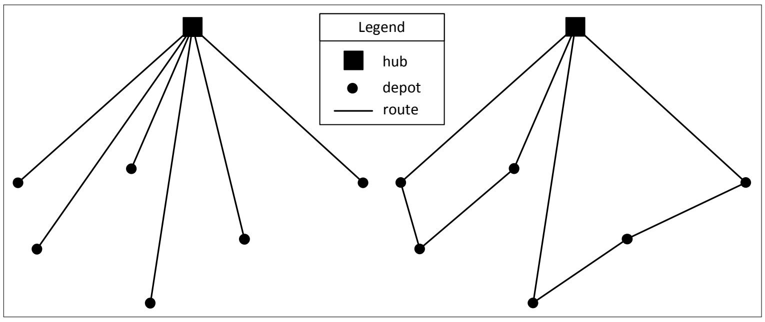


Figure 4.21: Milk run concept

In the situation on the left of the figure all depots have a direct route to the hub. This is not efficient since in most cases the loads that are going to the hub are less than a full truck load. This could be improved by letting trucks do a picking round at multiple depots and delivering these all at once at the hub. A further improvement could be combining the delivery of multi-day service parcels with picking up the parcels that are sent today. The situation on the right in the figure shows an example of combining the pick up of parcels at multiple depots. A round or 'run' like this is called a milk run, referring to milk man in the past that used to pick up empty milk bottles and deliver full bottles simultaneously. Although this strategy will induce a great amount of savings, it is not included in the model. The level of detail needed to calculate a good round is quite high and hence the calculations are on a tactical level while the rest of the model is on the strategic level. Milk runs are generally recalculated on a weekly or monthly basis, because they are moderately sensitive to changes in volume at the depots. The hub model is meant to create a network that suits a longer term like 5 to 10 years. Creating milk runs would therefore imply a long term stability that is not accurate.

4.2.6 Reduce hub cost

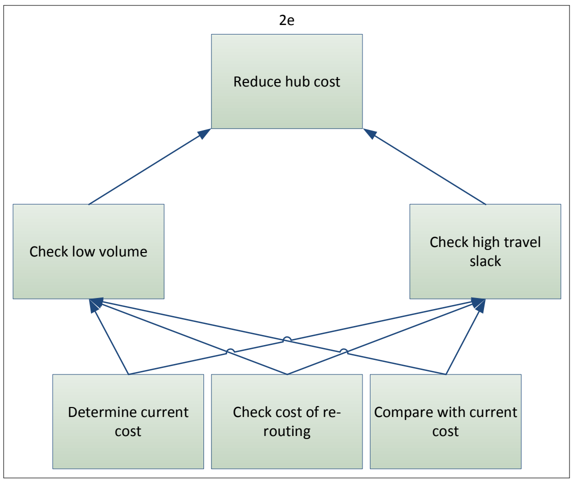


Figure 4.22: Means-ends objectives network: Level 2e

The last way of reducing the cost of the network described in this model is by reducing hub cost (see Figure 4.22). The hub cost (fixed and variable) can be reduced by economies of scale. Reducing hubs results in higher volumes on the hubs that are left, which reduces their marginal cost (cost per parcel). However, closing a hub requires all volume passing through it to find a new route. That route is usually longer than the old, previously optimised one; invoking extra transport cost. A hub should only be closed in case the cost savings outweigh the newly induced cost. Two conditions nominate a hub for possible closure: when only very few parcels flow through it, and when all parcels that go through the hub have a lot of slack time. Therefore while looking for hubs to close, two goals are handled *Check low volume* and *Check high travel slack*.

The most obvious reason to close a hub is because it handles relatively few parcels. The possibility exists that the few parcels that do flow through this hub, can only make their service via road when using this hub. Therefore, it cannot be closed without checking the feasibility of re-routing the parcels that are currently processed at the hub. Another reason to close a hub is because this will result in larger remaining hubs. Economies of scale will in turn provide lower total hub costs. Even in case a lot of volume is going through the hub under investigation (denoted as h^c), it is still presumed beneficial to close it in case the entire volume (or a great part of it) is able to make its service commitments by road via different routes. This means that these parcels should have some slack time in their routes that can be used to travel the extra distance. High travel slack refers to the share of the volume that has more slack time than a minimal threshold. In case a great deal of the volume at the hub has equal or more slack than the threshold, it is marked as high travel slack hub.

These two different reasons of attempting to close a hub instigate the same closing procedure that is as follows:

1. *Determine current cost.* The total cost of the current routes that go through the hub is determined.

2. *Determine cost of re-routing.* The total cost of the newly made routes is determined.

3. *Compare with current cost.* The cheapest option is chosen. In case it invokes closing the hubs, all depots having parcel volume move through that hub are informed. New routes are already available, because they were needed to determine the re-routing cost. So the involved d^o can immediately send confirmations of the new routes to the hubs via which the volume is re-routed.

This concludes the general description of the model design.

4.3 Chapter conclusion

This chapter has answered the fourth research question, '*What does a multi-agent express network model look like?*', by presenting the high level architecture of the designed model. The model consists of three main phases. In the first phase volume distributions at the depots are used as an indication as to where hubs could be placed. The initial hub setup that results from this phase serves as input for the second phase. In the second phase routes are created in an efficient and quick way to result in an express network that satisfies the constraints and functional requirements mentioned in Chapter 3. In the third phase the current network is altered by changing routes and hubs to achieve the main objective of the model: create a network with the lowest achievable network cost. Optionally, a current real-life network can be given to utilise the third phase standalone. The design can as such be seen as a twofold model, providing the possibility to use it as a whole or utilising a separate phase. Because the time span of this research was limited, the decision was made to implement only one part of the design in a proof of concept; this is presented in the next section.

Chapter 5

Proof of concept

When the model design is complete, the next step is to implement the design into a tool which assists in creating an efficient express network. As such, the quality of the model design is demonstrated. Unfortunately, the time line of this research does not allow for a full implementation of the design. Therefore, a part of the model that can be isolated from the design and is meaningful as a standalone unit is worked out in detail and programmed to serve as a proof of concept. Based on these prerequisites the *Create hubs based on destination volume* functionality as depicted in Figure 4.5 is selected. This chapter starts with a more detailed explanation of this selection and defines the name of the tool: POHST (Preliminary Organisation of Hub Setup Tool). Next, the detailed design of the proof of concept is presented. Subsequently, an illustration is made of the way the detailed model design is translated into code.

5.1 Concept selection

To demonstrate the quality of the model design a tool is made that incorporates the model logic. Implementing the entire model design into a computational model does not fit the time frame of this research. Hence, part of the design is selected to create a proof of concept. This has to meet the following selection criteria:

- manageable to implement in the research time frame
- isolable from the rest of the model design
- meaningful as a stand-alone unit

On a high level, considering the phases described in Paragraph 4.1.2 (Phase 1 handles creating hubs, Phase 2 creating routes and Phase 3 reducing cost by changing hubs and/or routes), Phase 1 is the only feasible phase of the model design which can be selected. Phase 2 focuses on creating routes. This phase cannot be split up into smaller parts, because the phase is too interconnected. Implementing this phase as a whole would not fit the time frame. Phase 3 needs the logic of Phase 2 to be implemented, which inherently renders it unmanageable to fit the time frame. Similar to Phase 2, Phase 1 cannot be modelled as a whole either. However, Phase 1 itself is not too interconnected.

Phase 1 consists of three sections: creating hubs based on destination volume, creating hubs for secluded depots and setting hub windows. The last section of setting hub windows is merely a preparation for Phase 2. As a result, it is not meaningful as a stand-alone unit. Creating hubs for secluded depots actually builds on the logic of the first section. Therefore, it is not isolable from the rest of the design, nor considerably meaningful as a stand-alone unit. That leaves Level 3a2: *Create hubs based on destination volume*. This section meets all the requirements. It is manageable to implement in the research time frame. It is isolable from the rest of the model design, because it does not need any other logic to be implemented. Finally, it is quite meaningful as a stand-alone unit, because it encompasses the most important part of Phase 1: creating hubs. The other two sections are merely expansions of this first section. In conclusion, Level 3a2: *Create hubs based on destination volume* is selected to be implemented as proof of concept. This proof of concept is called POHST: Preliminary Organisation of Hub Setup Tool and is described in the following section.

5.2 POHST

This section will first give a description of the inputs, outputs and model boundaries of the proof of concept using the system diagram. Then the sequence of events in the POHST model logic will be explained in the following section.

5.2.1 System diagram

The information needed to run the model and the variables of interests it produces are visualised in the system diagram of the proof of concept (Figure 5.1). The inputs are divided into two categories: environment and instruments. The environment represents the information that cannot be influenced by the user. Depot locations, parcel volumes and the driving time between nodes are facts that are part of reality. The instruments are the model inputs that the user is able to influence, in this case the parameters of the model (t^{max}, t^{min} and t^{close}). The user can run the model repeatedly with differing parameter settings to create insight into the network through the altering model outputs. These outputs are the number of hubs and the locations of these hubs. Next to these outputs, the detailed textual output of the model is potentially interesting. Amongst other information, it shows the order in which the hubs were selected, what region combinations they originated from and the type those region combinations are (close or distant).

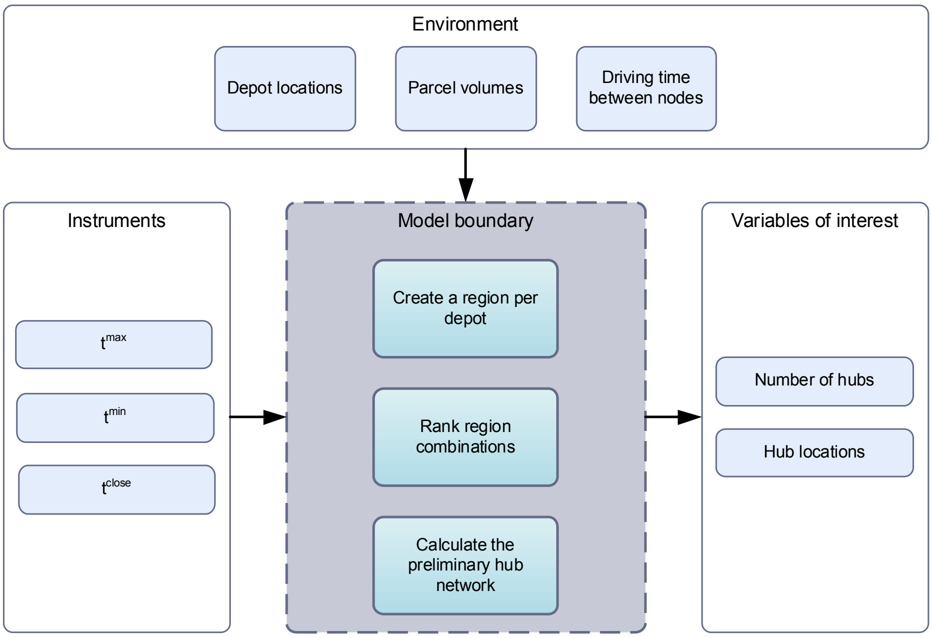


Figure 5.1: System diagram of the proof of concept

5.2.2 Description of POHST

The *Create hubs based on destination volume* functionality, as its name suggests, entails the creation of hubs in areas of which the volume distributions[1] have favourable conditions. Figure 5.2 illustrates the steps taken by POHST to achieve this.

As explained in Section 4.2.2, the process starts with the creation of regions as shown in Figure 5.2 step 1. Each depot creates a region with itself as central depot by defining region to consist of all depots within a radius of t^{max} driving time from itself[2]. After defining its region the depot will send its region definition to all other depots in preparation of step 3. Then, each central depot asks the depots in its own region to provide their volume distributions (step 2 in Figure 5.2). The central depot then aggregates the individual volume distributions into a volume distribution that represents the volume of the entire region going to all other depots. Each central depot can now combine their region volume distribution (from step 2) with the information from all region definitions (step 1) to upgrade the volume distribution from depicting region to depot volumes into region to region volumes. These region to region volume distributions contain one way volumes. A final upgrade should be made to get to a full matrix of back and forth volumes between the regions. To achieve this, all depots send their region to region volume distribution to the Hub_Setup Agent. There all received region volume distributions are summed to get the volume distribution with all back and forth volume between regions. This concludes step 3.

Step 1

Step 2

Step 3

[1] The volume distribution at a depot is the amount of parcels (e.g. in kg or m^3) that need to be transported from this depot to all other depots. It is called a distribution because the volume is specified per destination depot.

[2] The radius is depicted by a circle in Figure 5.2. Note that, since the radius is defined by driving time instead of a set distance it is dependent on infrastructure and hence will most often consist of a

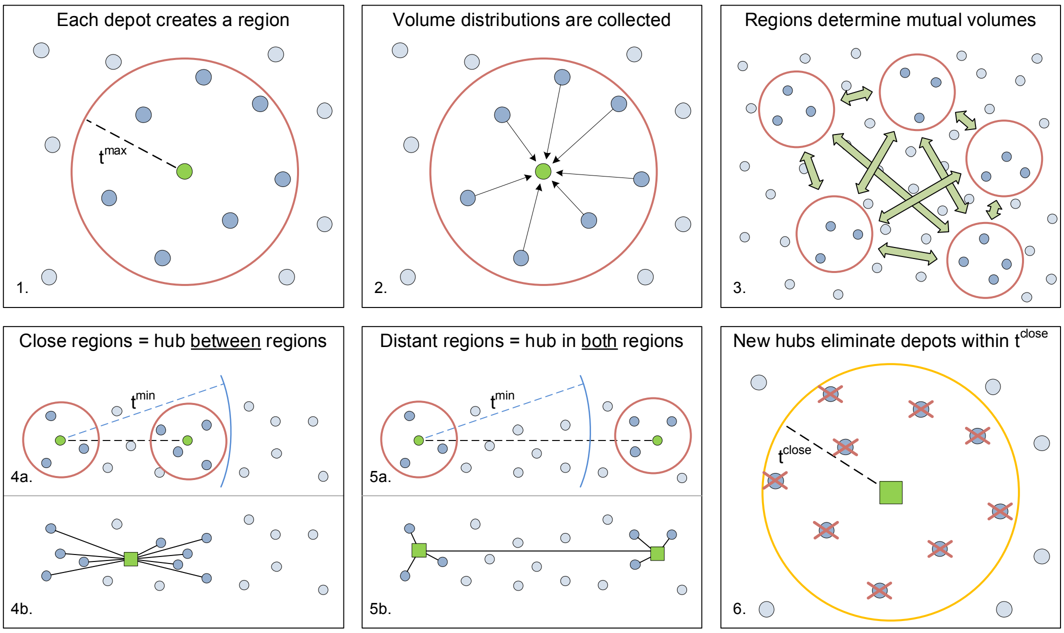


Figure 5.2: POHST model logic in six steps

Step 4 & 5 Next, the Hub_Setup Agent ranks each combination of two regions (now referred to as a region combination) to be close (Type 1) or distant (Type 2) as illustrated in Figure 5.2 step 4 and 5 respectively. It does this by comparing the travel time between the two central depots of a region combination with t^{min}. In case the travel time is less than t^{min} it is marked as a Type 1 combination (step 4a). If chosen during optimisation this region combination will result is one hub that lies in the middle between the two central depots of the region combination (step 4b). In case the travel time is more that t^{min}, the region combination is marked as a Type 2 combination (step 5a). If this is selected during optimisation it will result in two hubs by upgrading both central depots into a hub (step 5b). After marking the region combinations the optimisation starts by choosing the region combination with the highest volume. Depending on the combination type this will either result in one or two new hubs.

Step 6 To make sure two hubs are always a minimal distant apart, all depots near (near is the minimal travel time defined by t^{close}) the newly created hub(s) should be eliminated as potential hub (Figure 5.2: step 6). This is not trivial since the optimisation works with region combinations and not with individual depots. Therefore, for each depot that is close to the new hub(s) region is identified of which this depot is the central depot. This creates a list of regions to be excluded. Next, all region combinations containing at least one region on that list are eliminated from the set of viable region combinations. Once elimination is finished a check is done whether any region combinations are left on the list and if so the optimisation iterates by selecting the region combination that currently has the highest volume. Selecting high-volume region combinations, creating hubs and removing invalid region combinations is repeated until all region combinations are eliminated. When no region combinations

less clean defined circle as depicted here.

are left, the optimisation is finished and the model terminates.

The detailed design diagrams following from the detailed design steps in Prometheus are presented in Appendix C with corresponding description. This concludes the description of the proof of concept POHST. The next section explain how the design is translated into code.

5.3 Design implementation

Throughout the report the subject of ontology is mentioned. Especially Section 1.5 elaborates on the topic. It explains the way in which the real-life situation is translated to a model through increasingly specific methods and approaches. Section 1.5 ends on the design level with agent modelling. However, a final step is made from the design to actual JACK/Java code. This section will illustrate this final step by comparing a part of the design with the implemented code.

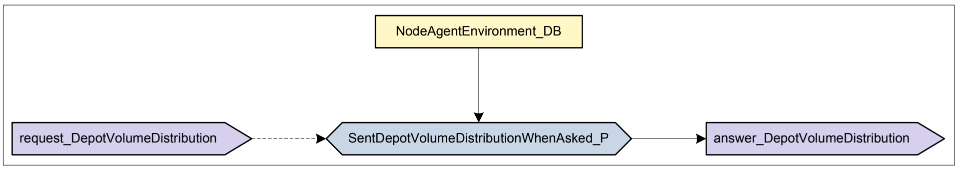


Figure 5.3: Example: plan

The capability *SentVolumeDistributionWhenAsked_C* is shown in Figure 5.3. It entails the functionality of receiving a request for its volume distribution, handling that request with the *SentVolumeDistributionWhenAsked_P* plan which reads the volume distribution from the *NodeAgentEnvironment_DB* and sends the answer to the requesting agent. The implementation of this plan will be illustrated and compared with the design, since it contains the majority of the logic within the capability. The implemented JACK code is shown in figure 5.4.

The package as shown in line 1 shows the archetype of the artefact declared in this class. Line 3 and 4 show the import statements of data and events, which enables this plan to interacts with these type of artefacts. Hence, a line connecting two types of artefacts in the design, always means that in the code the archetype on one end of the line has to be imported by the artefact at the other end. Line 10 declares the type of artefact, in this case a plan by using the *extends* statement. The following three lines of code (12-14) are typical JACK language statement. As the design shows a message[3] is received and sent by the plan which corresponds with line 12 and 13. Line 14 declares the use of the database by the plan as depicted in the design.

Next two methods *relevant* (line 16) and *context* (line 21) are used by JACK to determine whether this plan should be executed when receiving a event of the type declared in line 12. These methods are needed in case multiple plans may be used for the same kind of message. To determine whether a plan is relevant in a certain situation the *relevant* method (line 16) is tested. This method contain a boolean

[3] Messages are a type of event, just as actions and percepts. See Appendix B.1.

```
1 package au.edu.rmit.cs.plans;
2
3 import au.edu.rmit.cs.data.*;
4 import au.edu.rmit.cs.events.*;
5
6 /**
7 Sent own volume distribution which is stored in the NodeAgentEnvironment_DB to the
8 requesting (central) region depot.
9 */
10 public plan SentDepotVolumeDistributionWhenAsked_P extends Plan {
11
12     #handles event request_DepotVolumeDistribution request_depotvolumedistribution_h;
13     #sends event answer_DepotVolumeDistribution answer_depotvolumedistribution_s;
14     #reads data NodeAgentEnvironment_DB NodeAgentEnvironment_DB_dat;
15
16     static boolean relevant(request_DepotVolumeDistribution ev)
17     {
18         return true;
19     }
20
21     context()
22     {
23         // Trigger: Event "request_DepotVolumeDistribution"
24
25         true;
26     }
27
28     #reasoning method
29     body()
30     {
31         /**setup information needed at creation of message*/
32         int j = NodeAgentEnvironment_DB_dat.getSizeVolumeDistributionOfSingleDepot();
33         int [] k = NodeAgentEnvironment_DB_dat.readVolumeDistributionOfSingleDepot();
34         String l = this.getAgent().getBasename();
35
36         /**create message*/
37         answer_DepotVolumeDistribution q = answer_depotvolumedistribution_s.signal(l, j, k);
38         @send(request_depotvolumedistribution_h.nameInvokingAgent, q);
39     }
40 }
```

Figure 5.4: Example: coding of a plan

type statement that can be tested. Since the Node agent only has this plan to handle this type of message, the *relevant* method is not needed to make a choice between applicable plans and is therefore defined to be true. For all plans that pass the *relevant* test, the context method is tested to select the one that fits the desires of the agent best. Context methods only need to be specified when using BDI agents. Since the weak definition of agents is used in this research all context methods are declared to be true.

The *reasoning method* or *body* (line 29) contains the actual code to execute if this plan is selected to handle the event. It starts with gathering information to be sent from the database. First the number of depots is determined at line 32, followed by reading the volume to each of those depots. Information gathering ends with storing the name of the agent executing the plan. Next the message to be sent is created. Line 37 calls the *answer_DepotVolumeDistribution* class, in which this type of message is declared, to create the message object using the information just gathered. Finally, this message is sent by the plan at line 38 as indicated in the design by the outgoing arrow. This concludes the description of the proof of concept.

5.4 Chapter conclusion

The main part of the first phase of the designed model is implemented as proof of concept, thereby answering the question: *How can the conceptualisation be translated to a tool*? It concerns determining an initial setup of hubs based on volume distribution patterns. It is based on the assumption that in case a relatively high amount of volume needs to be transported between regions, it is beneficial to have hubs at both locations. A detailed design of this process was made and revealed two great advantages of using the Prometheus methodology. First, the very high detail design is very close to actual programming, making the last step from design to code as easy as possible. Second, the Prometheus Design Tool is able to generate a code framework out of the detailed design. As such, it guards the structure of the model consequently aiding in the verification. The full code of the proof of concept can be requested from Taco Wijnsma[4] or Martijn Warnier[5]. With the proof of concept established, the next chapter can shows the actual use of the tool.

[4] tgwijnsma@gmail.com
[5] m.e.warnier@tudelft.nl

Chapter 6

POHST application

The application of the proof of concept POHST is where it can show its added value as strategic analysis tool in express networks. This chapter aims to demonstrate how POHST can be used in the preliminary phase of network research aiding in data gathering and the generation of start solutions. To illustrate these functions two data sets are analysed. Both data sets are described in Section 6.2 and are fully presented in Appendix D. The chapter starts with an introduction of using POHST as strategic analysis tool. This includes a practical description of the GUI and the different parameter settings. Then the actual experimentation using the two datasets is shown. Part of the results are shown in Appendix D. The next chapter will describe the user experience of the tool via the expert validation.

6.1 POHST as strategic analysis tool

This research is performed at the OCG[1] where strategic analyses are done on express networks. Their projects start with data gathering and swift high level analysis to gain insight in the properties of the region under investigation. In this section a case is made for the assistance of this process with POHST in two major ways: to help early data gathering and analysis and to create start solutions for other models utilised by the OCG.

As shown in Chapter 5, POHST generates preliminary hub setups. By generating several network configurations with use of the model parameters, different start solutions can be created. Start solutions[2] are used as input for optimisation models. Furthermore, these same functionalities can be utilised in swiftly analysing the

[1] OCG stands for the ORTEC Consulting Group as introduced earlier in Chapter 1.

[2] Start solutions are difference starting values for optimisation model. Multiple start solutions are used to enhance the change that one of them will reach the global optimum before getting stuck in a local optimum. Start solutions that are closer to the global optimum can help reaching the global optimum faster during optimisation. However, random start solutions are often used to ensure starting optimisation from very different places in the solution space.

robustness of these hub setups. Some areas might be more sensitive to parameter variations than others. Indicating it would be beneficial to gather more detailed data from these areas. To explain how to use POHST the next sections describe the GUI and explain the influence of the parameters on the model output.

6.1.1 GUI

The GUI provides easy access to the tool and helps to gain insight in the used data through clear visualisation of the geographical dispersion of depots and the hub setup process (i.e. by animating the order in which hubs are created). In addition, it requires little data to run and the calculation time of the tool is minimal as Appendix D will show. The GUI consists of two main tabs. The tab *Databases* provides the environment data selection interface. The depot volume distributions and driving times between nodes need to be entered here. In case visualisation is used, the set of latitudes and longitudes need to be selected as well.

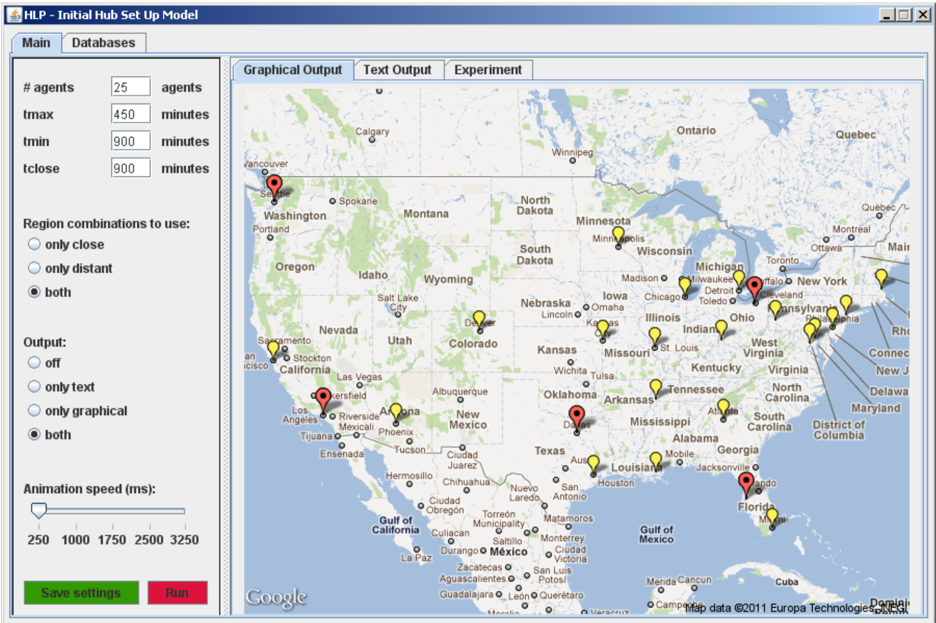


Figure 6.1: Graphical User Interface of the proof of concept

The *Main* tab consists of a sidebar and three additional tabs named Graphical output, Text output and Experimentation. The sidebar starts with stating the parameters. The number of agents needs to be entered here to let the model run correctly. It may not be varied and exists only because of the structure of the code behind the GUI. Next, the three model parameters are stated. As explained in previous chapters t^{max} determines the radius of the regions, t^{min} defines the radius by which region combinations are classified as close or distant and t^{close} determines the elimination radius used at the end of each iteration. The parameters will be discussed in short in the next section and their influenced will be treated in detail in Section 6.2. Following the parameters a radiobutton group offers the possibility of letting the algorithm choose only close regions, only distant regions or the best of both. The next radiobutton group shows the output options. POHST provides the possibility to show the model outputs by visualisation through a connection with Google Maps and by showing textual output. In case the option *off* is selected the textual output pane will only create some statistics on the model run, instead of providing a detailed insight in steps taken by the agents in the model. After the output the option is provided to slow down animation further than default. Since model runs are quite fast a delay is activated after every optimisation iteration in case visualisation is requested. This gives more time to see the different hubs pop up when selected by the algorithm. The sidebar ends with a save settings button that has to be used before one is able to start the model by using the run button.

The graphical output tab shows the geographical area spanned by the inserted geographical locations of the nodes. The area shown is spanned through automatically determining a zoom level that can show all depot locations. The depot locations are displayed using small yellow markers. As soon as a depot is upgraded to a hub during a model run, its small yellow marker will be replaced with a larger red marker. The text output tab contains detailed information on the progress of the model run with details provided by the agents themselves. The experiment tab provides the possibility to execute small parameter sweeps while storing the statistical outcomes on the local hard disk. For each parameter the number of times it should be varied can be entered and the amount of minutes it has to be increased with between runs can be set. For instance setting the t^{max} parameter at 50 minutes and indicating in the experiment tab that it should be varied 15 times with 10 minutes there will be 16 model runs with t^{max} at 50, 60, 70, ..., 190 and finally 200. The statistical output of each run will be stored locally enabling further data analysis in other statistical tools. More details on the statistical output is given in Section 6.2. The parameters are discussed further in following section.

6.1.2 Parameter setting

This section contains an explanation of the explicit and implicit meaning of the parameters. Their influence is illustrated with the CAB and TN data in the next section.

Defining regions with t^{max}

explicit: All depots within this travel time (defined in minutes) of a central depot A are marked as belonging to the region of which A is the central depot. A being an index for all depots.

implicit: reducing this parameter favours depots that are in compact high volume density areas to become a hub. Increasing the parameter enables depots that are located in regions where volume density is on average quite high but more dispersed to become hubs as well. Put differently, reducing

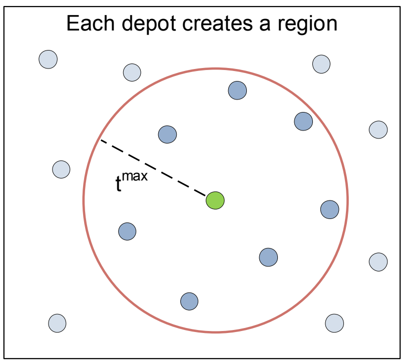


Figure 6.2: Parameter t^{max}

the value of this parameter excludes depots in areas of the latter category increasingly (relatively favouring the former).

Classifying regions with t^{min}

explicit: Region combinations of which the central depots are closer to each other than this specified travel time (in minutes), are marked as close (type 1) others as distant (type 2). Selected region combinations of type 2 will result in upgrading a depot to a hub in both involved regions. Selected region combination of type 1 will result in upgrading a depot that lies between the two regions to a hub.

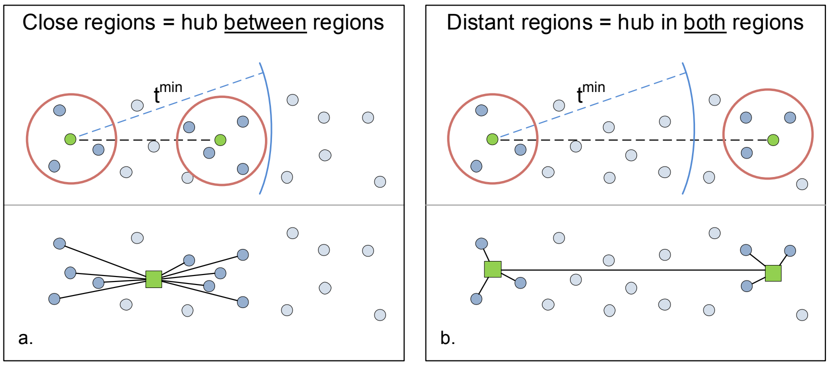


Figure 6.3: Parameter t^{min}

implicit: This parameter influences the importance of distances from depots to hubs in a quite direct manner. A type 2 combination will result in hubs being closer to their destination and having more interhub travel than type 1 combinations. A higher or lower value is not by definition better or worse and will not necessarily lead to more or less total travel time. However, it does influence the relative ratio between depot-hub travel and hub-hub travel.

Eliminating region combinations with t^{close}

explicit: Travel time in minutes for which all nodes closer to the existing hubs than this specified travel time will be eliminated from the possible set of hubs (more specifically the region combinations corresponding that node are eliminated). If the node to eliminate is already a hub, it will be left uninfluenced.

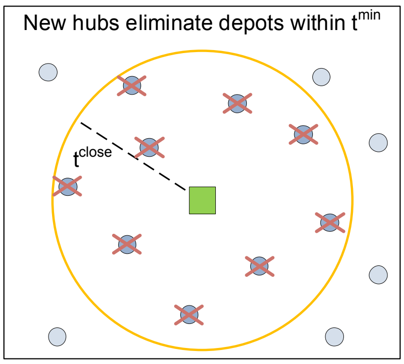


Figure 6.4: Parameter t^{close}

implicit: t^{close} directly influences the number of hubs that the model will create. Having more hubs increases hub costs and reduces the total amount of travel, so transport cost decrease. This is an implicit influence, since the tool does not calculate the costs. However, the ratio of hub costs versus transport costs will definitely differ as stated when the cost will be made explicit.

6.2 Experimentation

The experimentation exemplifies the use of POHST in analysing network datasets. The two datasets used are introduced first. An explanation of the default parameters for each dataset follows. Then the scoring system used to quantify the model outcomes is introduced. Finally, the actual experimentation is given, highlighting a number of exemplary results.

6.2.1 Data sets

Two data sets are used for the experimentation. First is the Civil Aeronautics Board (CAB) data which was originally used by O'Kelly (1987). It is based on movements of airline passengers between 25 major U.S. cities in 1970 and is used extensively in literature since its introduction (e.g. Klincewicz (1992); Jaillet et al. (1996); B. Y. Kara & Tansel (2001); Topcuoglu et al. (2005)). Its exposure in literature makes it especially interesting and is the reason for using it in this research. Although it presents a network and can be used to test the POHST, it does not represent a typical express network. Therefore an additional dataset is used.

The second dataset was introduced by Çetiner et al. (2010) where it was used to analyse the Turkish postal services (PTT). This Turkish Network (TN) data contains 81 major Turkish cities[3]. With 81 nodes this dataset can be considered to be very large. Many literature on the CAB dataset for instance shows that its usage is regularly limited to using 10 nodes instead of the full 25 (See for instance Jaillet et al. (1996) and B. Y. Kara & Tansel (2001)).

6.2.2 Default parameters

To be able to research these datasets with POHST default parameter settings are defined. First, these settings for the TN data are described, followed by the reasoning behind the parameter settings for the CAB data.

TN data

The goal of POHST is to calculate a network design that gives an optimal start towards a fully designed express network. Since the TN data is more alike a real express network, it is described first. The default settings for the parameters are chosen in such a way that the service commitment and flow conservation requirements are met. Parcels with the highest service have at most 12 hours to get from their origin depot to their destination depot[4]. By default (but not restricted to) the POHST assumes depot-hub-hub-depot routes. In this case, hubs should be at most 9 hours apart. This follows from the European working time directive that states that

[3] The geocoding data for both datasets was collected manually with a combined use of Google Maps (http://www.maps.google.com) and the names presented in the original datasets.

[4] This service obviously varies with varying destinations. The OCG found that using 12 hours as the highest service, is a reliable assumption. As such, this research assumes the same

truck drivers need to rest after 9 hours of work. Keeping hubs within these 9 hours of driving time from each other, ensures that parcels can be transported between the hubs without delays. By subtracting these 9 hours from the original 12 hours, 3 hours are left for both the origin depot to origin hub as well as the destination hub to destination depot parts of the route. Consequently, 90 minutes are available for each part of depot-hub travel. Hence, depots should most ideally lie within 90 minutes driving time from at least one hub.

This 90 minutes range (defined for express networks and thus the TN data as Maximal Travel Time to Closest Hub or t^{mttch}) is therefore a natural setting for the t^{max} parameter. t^{max} defines the range of regions and a region logically should contain depots that can reach the central depot (and hence the potential hub) in time to make their service commitments. Hence, t^{max} is defined as follows:

$$t^{max} = t^{mttch} \tag{6.1}$$

Furthermore, two adjacent hubs should not be further away from each other than twice t^{mttch}. This is to avoid situations where a depot is situated between two hubs it cannot reach. During the elimination of potential hubs, this should be considered. Hence, t^{close} is logically defined to be at most twice t^{mttch}. Additionally, a lower bound for t^{close} is defined as well. This originates from the same reason as why t^{close} is used in the first place, namely to avoid that hubs get too close to each other. The lower bound of t^{close} is set at t^{max} to avoid hubs entering each others regions. This concludes for the TN data (and other express network data) that the definition of t^{close} is most logically (though not necessarily):

$$t^{mttch} \leq t^{close} \leq 2 * t^{mttch} \tag{6.2}$$

Note that t^{close} is used at the end of an iteration after choosing one or two hubs to avoid new hubs getting too close to these hubs in the next iteration. To make sure hubs do not get too close to each other while creating hubs, t^{min} is used. It serves the same purpose as t^{close} although in a different way; at a different place in the model logic. Following their similar function t^{min} is defined as t^{close}:

$$t^{mttch} \leq t^{min} \leq 2 * t^{mttch} \tag{6.3}$$

Following the equations and with t^{mttch} at 90 minutes and t^{close} and t^{min} at their highest bounds to reduce hub cost, t^{max} equals 90 minutes, t^{close} equals 180 minutes and consequently t^{min} equals 180 minutes as well.

CAB data

Since the CAB data does not originate from an express network it cannot be assumed that it is sensible to approach it with the same settings. A quick model run confirmed this since it resulted in 23 of the original depots to be upgraded to hubs. Although the absolute values used as standard TN settings are not sensible to use on the CAB data, the established relation between the parameters is. To derive standard settings for the CAB data, the outcomes in related literature are kept in mind. Klincewicz (1992); Topcuoglu et al. (2005); O'Kelly (1987) all use the full CAB dataset and present solutions in which 1 to 5 hubs were created. Most of their solution are created 3 or 4 hubs. Between these latter two, the arbitrary choice is made to tailor the parameters to result in a solution with 4 hubs. This means that t^{max} equals 450 minutes, t^{close} equals 900 minutes and consequently t^{min} equals 900 minutes as well.

6.2.3 Quantative scoring

The only quantitative data that POHST produces that can immediately be used for statistical analysis is the number of hubs in the solution. Although it also states which hubs are chosen, this data is not usable for analysis yet. Therefore a scoring system is designed that aims to include support for all important influences of hub selection. Most experiments are done while varying the parameters. After every model run, points are rewarded to the depots that are upgraded to a hub. When adding the score for the depots over all model runs, it becomes clear which depots are upgraded the most, since these will have the highest scores. Having a high overall score indicates the robustness of the related hub.

Since the number of hubs selected can vary between runs and each run should be valued the same, the total score that is divided between the opened hubs each run is always 1. Consequently, a hub is rewarded a larger share of this one point when it is part of a solution with less hubs. The reason behind this decision is that a hub is considered to be more significant when it is part of a solution with for instance 5 hubs than when it is part of a solution with 30 hubs.

A last property incorporated into the scoring system is the order in which the hubs are chosen. A hub that is chosen first will get a higher score than the one that is chosen second. The second hub in its turn gets a higher score than the third and so on. The consequent scores decrease linearly. To construct a linear scale that sums up to one, formula 6.4 is used to calculate the score (S) for a single hub (x). It uses the total number of hubs (n) and the order (a) in which they were created (i.e. for the hub that is created first $a = 1$ and for the third hub $a = 3$).

$$S(x) = \frac{1}{\left(\frac{n(n+1)}{2}\right)} * (n - (a - 1)) \tag{6.4}$$

6.2.4 Experiments

Several experiments were performed using the CAB and TN data. The parameters were varied to show their influence on the model output. In addition, two structural tests were executed. One varied region settings; comparing solutions using only close, only distant or both region combinations. The other test actually reversed the algorithm. As such, the tool did not select the region combinations with the highest volume first, but instead selected those with the lowest volume first. The full experimentation is given in Appendix D. Here, the most interesting results are depicted and corresponding conclusions are drawn.

Varying t^{max}

Both the CAB and the TN data (depicted here in 6.5) showed that varying t^{max} leads to quite differing outcomes. Many different depots are upgraded to hubs and they all have about the same score. The amount of depots upgraded did not vary much; either 16 or 17 hubs were created in each run. As expected, the number of hubs created in both datasets was not sensitive for t^{max}; the parameter involved in eliminating depots around hubs (t^{close}) is expected create more variations in the number of hubs created. On the other hand, which hubs are selected is very sensitive to t^{max}. This is inherent to the nature of t^{max}. When it increases, the radius of a region is increased. That leads to more depots being included in a region. As such, the total amount of volume within every region changes, which leads to a rearrangement of the ranking of region combinations (different region combinations are now ranked as having the highest volume). This in turn leads to different hubs being opened, which induces such a high variation in scores per depot.

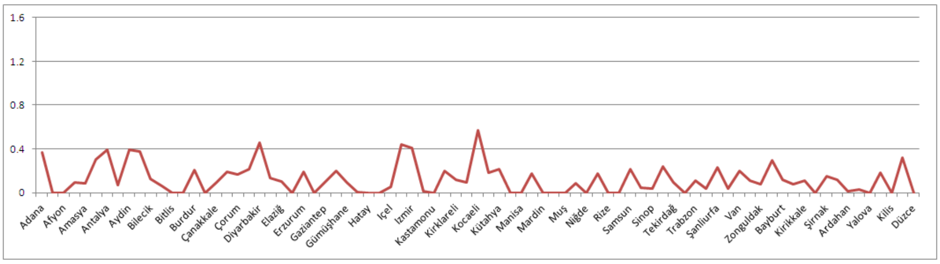


Figure 6.5: TN data: score per hub varying t^{max} from 18 till 180

Varying t^{min}

Figure 6.6 shows the score of the TN nodes while varying t^{min}. Similar two the CAB data, the opened hubs are much more robust (i.e. their scores are much higher, since these hubs make up almost every outcome). On the other hand, an average of 16 hubs opened is still quite high. This indicates that t^{min} foremost influences which hubs are chosen and in which order and as t^{max} it does not influence the number of opened hubs as much.

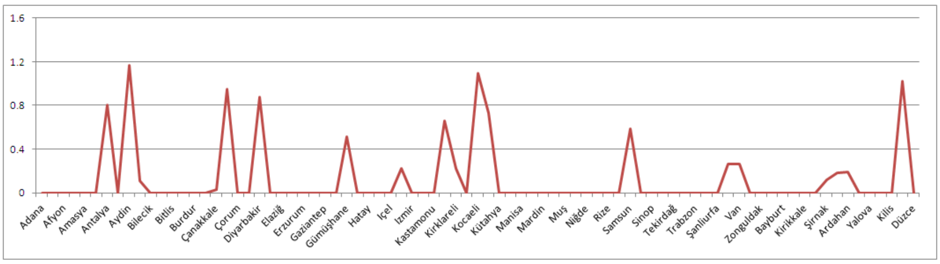


Figure 6.6: TN data: score per hub varying t^{min} from 36 till 360

This makes sense, because increasing t^{min} causes more type 1 region combinations. As such, only one hub is created instead of two. Thus, depots are only eliminated around one hub, which induces more iterations before all region combinations are eliminated[5]. This is why the same amount of hubs are opened in each run.

Varying t^{close}

Finally, varying t^{close} shows the most distinct outcomes of the three parameter variations. This is depicted for the CAB data in Figure 6.7. The distinct outcomes are due to the high variability in the number of hubs. In the CAB data an average of 6.1 hubs are created with an very high standard deviation of 6.05. In the first run, with t^{close} at 180 minutes, the number of hubs opened is 21 (out of a total of 25 nodes). This number decreases roughly exponentially until it stabilizes on opening just 2 hubs in the last three runs. Since each run is awarded the same amount of points (1), the last runs have a much greater impact on the total score. As such, Cleveland, Los Angeles and Tampa clearly stand out. This clearly shows the sensitivity of the data to variations in t^{close}.

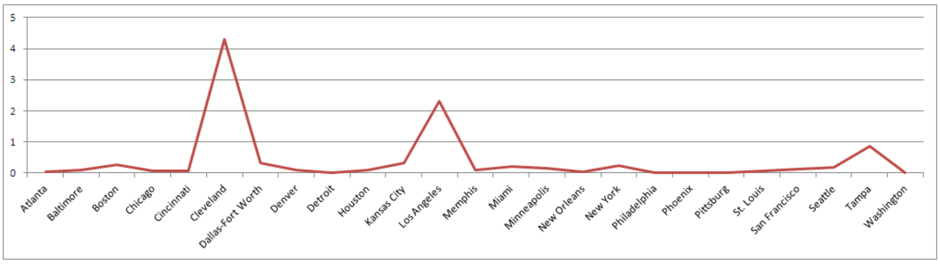


Figure 6.7: CAB data: score per hub varying t^{close} from 180 till 1800

Calculation time

The calculation time is tracked for every model run. For the experiments on the CAB data as described in previous section the average run time was 0.88 seconds. It is very stable with a standard deviation of only 0.034 seconds. The shortest run time lasted 0.83 seconds and the longest run time was 1.00 seconds. For the TN data during the same type of experiments, the average run time was 1.45 seconds. Again, the run time was incredibly stable with a standard deviation of 0.037 seconds. The shortest run time was 1.414 seconds and the longest run time was 1.488 seconds. Both average run times are very fast considering normal use[6]. These run times allow users to test many different scenarios while leaving sufficient time to analyse the outcomes.

[5] In most cases more iterations are induced, but this is not necessarily the case. Type 2 region combinations might also lead to the opening of just one hub. This happens when the other depot was already opened as a hub. The situation in terms of the amount of new hubs and the amount of eliminated region combinations then equals a situation where Type 1 region combinations are found

[6] Besides normal use, it is interesting to compare the run time with other methods presented in literature. Although such literature was found, the model should first be fully implemented to be able to perform a valid comparison. Hence, such comparison is not included in this report.

Reverse algorithm

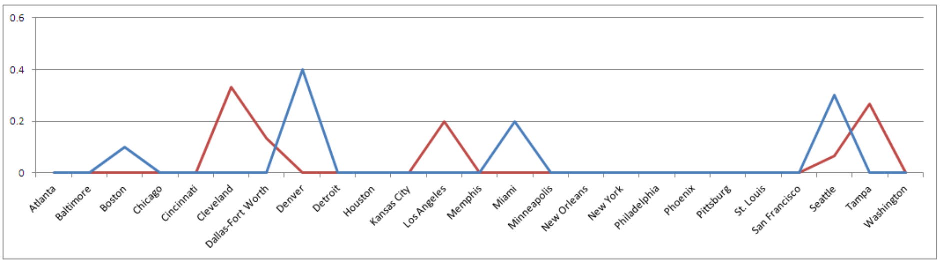


Figure 6.8: CAB data: score per hub normal versus reversed algorithm

As Figure 6.8 shows, the number of opened hubs is not very sensitive to the algorithm setting. Roughly the same number of hubs is maintained by the fixed parameters of t^{max}, t^{min} and t^{close}. On the other hand, the hub locations do clearly differ in the two approaches. When the region combinations with the lowest volume to be transported between them are selected first, this obviously leads to different hubs to be chosen first. Interesting to see in the CAB data is that Seattle is chosen with both algorithm settings, although much earlier when using the reversed algorithm. This indicates the low parcel volume in Seattle, but also the isolated area around Seattle.

Region settings

Differing between distant and close region combination has a significant impact on which nodes are upgraded to hubs, as can be seen in Figure 6.9. The run using both region combinations (indicated by the red line) is offset by +0.02 points, so it does not overlap with the other runs and remains visible. Interesting to see in the TN data (unlike the CAB data) is that the model run using both region combinations is exactly the same as the model run using only distant region combinations. This indicates that the destinations of most parcels in the Turkish network are not close to their origin.

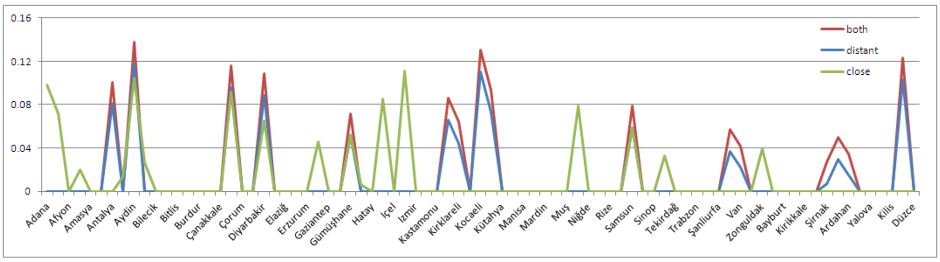


Figure 6.9: TN data: selections per hub using both region types vs. only close and only distant

6.3 Chapter conclusion

This chapter answered the research question, '*What is the added value of the tool?*', with use of three different topics. First the benefits of representing the data through a map in the graphical user interface was explained. It provides a spatial insight in the distributions of hubs in the network. Secondly, the tool provides an accessible way of performing multiple model runs, with changing parameters. Parameters that can be varied are: the radius of regions, the border between close and distant region categories, and the range in which potential hubs are eliminated when a new hub is established. The impact of the three parameters was investigated through data analysis using two data sets that originate from literature. These showed that the number of hubs is very sensitive for the parameter that determines the range of elimination of potential hubs but barely for the other two. On the other hand, the arrangement of the chosen hubs is very sensitive to the radius of regions and the border definition, but less for the elimination of potential hubs. During the experiments, the tool acted as expected; though an official confirmation of the design and implementation of the model results from the verification and validation that is presented in the next section.

Chapter 7

Verification and validation

The previous chapter demonstrated the application of POHST. It showed its capabilities as tool in the preliminary phase of network research. However, to confirm the quality of the tool verification and validation are necessary. This chapter describes both. First, different verification methods are discussed, including expert verification, and their results are concluded. Next, the expert validation[1] is given, which includes expert recommendations to improve the tool. Finally, the user experience of POHST is briefly described.

7.1 Verification

The verification entails securing that the programmed model behaves as intended by the model design. Different verification methods are described in literature, but as Kleijnen (1995) states, none of them are perfect. He discusses modular programming, checking intermediate simulation outputs (unit testing), checking final simulation outputs (integration testing) and animation. All of these methods are considered for use in this research. As far as the methods were applicable, they have been used in the verification. The next sections describe the different methods, whether or not they have been applied and if so, how they were applied. This paragraph concludes with the verification result.

Modular programming

Modular programming means breaking down a model into smaller modules, where each module has its own independent functionality. It is considered a general good programming practice, because it helps with locating bugs in the software. When unintended behaviour is observed, the module that caused this behaviour can be isolated, altered and tested separately to fix the problem efficiently. This research

[1] This is based on an expert session with a group of six experts of the OCG.

uses a hierarchical breakdown of the model to identify the modules. There are two types of modules on the first breakdown level, namely the agents and the messages between them. The modules in the second breakdown level are the capabilities within the agents, the internal messages between the capabilities, and the databases within the agents. The third and lowest breakdown level is represented by the plans and messages within the capabilities. During programming, much effort was put into have a working model every time a new module was completed. This allowed for early and accurate bug finding and helped to make sure that only minor fixes were needed after completing the last module. Besides providing a clear programming strategy, modular programming allows for verification of the separate modules by unit testing.

Unit testing

Unit testing is used during programming to ensure consistency. It is a method widely used by programmers and consist of testing individual modules on their behaviour. For example, each time agents communicate with each other or with themselves, the (partially finished) tool is run to test whether the message is created, sent, received, and if the message can be read by the recipient. In a similar way the functionality of all agents, capabilities, plans and databases is tested. After a module was finished, it was immediately subjected to unit testing. Corrections were made where unindented behaviour was detected.

The use of unit testing involves a very important assumption namely that of compositionality. With compositionality it is assumed that the whole of the model is the same as the sum of its parts. This is not necessarily true.Therefore, unit testing is often followed by testing the model as a whole. This is called integration testing.

Integration testing

Integration testing involves running the model over the range of inputs that the model is suited for. This is not possible for the model in question, since amount of possible combinations of the travel distances between nodes and the volume distribution of each one of them is infinite. Moreover, there is no limited amount of network configurations nor a limited amount of configuration types. Generally, the model cannot be tested for a full range of model inputs. What can be said about the correctness of the final model is that at completion of the programming process, the model did not show any unintended behaviour at the unit tests. Furthermore, the application of the model to the CAB data set and the TN dataset in Chapter 6 has shown correct behaviour at every model run executed so far. Although this is no guarantee for further use, it does add to the credibility of the model.

Animation

Animation is incorporated into the graphical user interface (GUI) of the model to serve multiple purposes. First of all, interpreting the model results is much easier with a graphical representation, which simply displays a depot turning into a hub, than by reading the textual output, which states that a node with a certain ID is upgraded to a hub. Second, the model results can also be analysed quicker and with more context, since more geographical information is presented. Both arguments make this model more accessible as a tool. Thirdly, the animation helps with the verification of the model. For instance, one can easily trace whether the hubs that are opened, actually stay open. And it is easily visible whether the locations of all nodes stay fixed. Furthermore, since the user interface allows for quick and easy alternation of the parameters and re-running the model, it helps in verifying the influence of the parameters on the model.

Expert verification

After a group session, which is introduced in more detail in the validation section, experts were asked questions considering the verification of POHST. These questions address the strengths and weaknesses of the tool, the model construction and whether or not the experts expect the tool behaves as intended by the model design. This section describes a selection of comments made by the expert. A full overview of all comments is given in Appendix E.

There is a general consensus on the ease of use of the model and the fact that it has very little runtime. The experts see merit in using the tool to find advanced starting solutions for network configurations. On the other hand, several limitations that were either applied in the tool alone or in the model design as well, were received less enthusiastic. They would have liked to see PUD included in the tool, as well as imbalances in the volume. One expert would have liked to see actual routes being determined to see if agent modelling could solve the hub location problem. Although these comments logically follow considering the question "what are the weaknesses of the model structure", these have already been considered in the scope as described in Chapter 3 or while choosing a proof of concept (Chapter 5). Restating these, PUD is out of scope since the Hub Location Problem only considers the placement of hubs and routing between the depots and these hubs. Imbalances in volumes refers to sensitivity analysis on the volume distributions that are inputted in the model. Considering imbalances in volumes has not been a topic in this research. However, POHST does allow for quickly changing the used database. If different databases with varying volume distributions are prepared POHST would support this type of analysis. However, an improvement that could be made is to automate this process in such a way that only the original database has to be inserted and the tool would automatically perform a sensitivity analysis on that data. Finally, creating routes as commented by one expert is out of scope of this proof of concept. It is incorporated in the model design as presented in Chapter 4, but not a part of POHST because the time frame of this research did not allow to include it.

Two interesting recommendations are as follows. First of all, it was recommended to include functionality in the tool to create a bubble chart of the volume distributions. This is a graphical overlay on the map showing a filled circle at each node. The size of each circle is relative to the volume at that node. It could help understanding why certain decisions are made by the tool. To illustrate this, bubble charts of the CAB data and TN data are included in Appendix E. The second recommendation was to be able to force the number of hubs that the model outcome should consist of. This is not considered to be either a possible nor desired feature of this model. The recommendation is related to another type of formulation to the HLP, namely the p-median problem (Mirchandani & Francis, 1990). In these cases the best organisation of a fixed number of hubs is determined. However, the idea behind the approach in this research is to have the number of hubs emerge from the data and parameter properties. Besides these comments and recommendations, all experts agreed that the model was well constructed and that it behaved as intended by the model design.

Verification result

Although the type of model inputs do not allow for thorough integral testing, other verification methods have shown that the model is behaving as intended by the model design. Modular programming has undoubtedly helped in preventing bugs. Unit testing and the animation have both assisted in locating any remaining unintended behaviour. Finally, expert verification confirmed the proper functionality of the tool.

7.2 Expert validation

Ideally, the model would be validated by use of quantitative analysis. However, since the time frame of this research does not allow for a full implementation of the design as presented in Chapter 4, there are no complete networks with related network cost to validate. Instead, the proof of concept produces a selection of hubs of which it is not sure yet what the corresponding network cost will be. To still be able to provide confidence in the quality of the proof of concept, an expert review session is held.

The validation is based on a group session with five experts from the OCG. These experts all have university grades in either econometrics, mathematics or mathematical engineering. Some of them work as programmers; they program and test mathematical models. Others work as analysts and use these models to analyse network characteristics. In the session, the model design was explained and the functionality of the tool was illustrated using the CAB and TN data. These demonstrations were followed by a thorough discussion on both the model structure and the usability of POHST. Following the group session, the experts were asked to answer questions referring to the verification (as used in the previous section), on the validity of the model, and reflecting on the tool. The questions relating the latter two categories along with a summary of the answers are given below. The entire expert session is depicted in Appendix E.

1. Do you consider the model to meet all the design requirements?

One expert correctly stated that the tool does not consider the entire scope of the model requirements. As such, it does not minimize network cost, nor does it meet the service requirements. However, as validation for the tool, another expert mentioned that the tool did choose the proper locations as compared with expected results.

2. Do the model outcomes change as expected when altering the parameters?

This question was asked in trifold, reflecting on each parameter. However, responses on these questions were low. Four experts responded on the question related to the effect of t^{max} all confirming the model outcomes varied as expected. Only one expert responded on the question relating t^{min} and t^{close} stating both influenced the outcomes as expected. Hence, the influence of t^{max} is considered to be confirmed. A conclusion on t^{min} and t^{close} cannot go further than stating that the single received response was positive.

3. Do you consider the HLP to be well represented by POHST?

The experts agreed that the tool represents the first phase of the HLP well. However, to truly represent the HLP, the next steps of creating routes and reducing cost are crucial.

Validation result

It is very difficult to validate the proof of concept. Due to the lack of quantitative outcomes that allow for comparison with results found in literature, the tool was validated by expert review alone. The expert validation showed that experts have confidence in the model outcomes and consider it to generate valuable start solutions. On the other hand, they also comment that the quality of these start solutions remains unsure until quantitative tests can be performed.

7.3 User experience

Although verifying and validating the proof of concept was the main topic of the group session, the experts were also asked a few questions on the user friendliness of the tool. Experts were asked to reflect on the visual attractiveness, the provided functionality and to think of situations the tool could be used in.

1. What are the strengths of the user interface?

The experts all commented on the user friendliness of the graphical user interface. It has a straightforward design that is very intuitive. Furthermore, the greatest added value is the graphical representation of the depots and hubs. It provides insight in the geographical dispersion of the nodes and provides the possibility to see the order in which the hubs are created, which aids analysis.

2. What improvements can be made to the user interface?

Several improvements are suggested. For instance changing the parameter names to be more self explanatory. This is not done yet to keep the consistency between the report and the tool, but will be considered for use of the tool after the research. Other suggestions were related to providing even more insight in the used data. For instance, as mentioned before, one expert suggested the use of bubble charts to visualise the transport flows. This would be a valuable addition to POHST, but too great of change to incorporate during this research. Appendix E contains bubble charts of the CAB data and TN data and quickly reflects on the relevance of these charts. Another suggestion on adding units to the parameters, however, was processed immediately.

3. What are the weaknesses of the application?

One additional weakness is identified that was not mentioned before. The tool would benefit from having more guidance for the user. There is no information present in POHST that explains how to use it and what the different parameters mean. This recommendation will definitely play an important role in enhancing POHST from a proof of concept towards a mature tool.

4. What situations would you expect that the tool could be used in?

The tool is considered to be most useful for quickly gaining insight in a network and to generate starting solutions. Especially generating starting solutions is mentioned multiple times by different experts. This underpins the success of the design and its implementation through the tool, since the proof of concept was built based on the *Create hubs* logic of the initial model design in Chapter 4. It was designed to result in a hub configuration that would serve as an input for the *Create routes* phase, after which the third phase of reducing cost could be performed.

User experience result

The visualisation is of great added value; enabling insight not only in the final network configuration but also showing the order that the hubs are created in. This indicates the significance of the hubs, since the hubs that are created first, result from region combinations with the largest volume to be transported. The improvements mentioned were to rename the parameters, to give the parameters units and to add bubble charts provide additional insight in the used data. This last recommendation is analysed further in Appendix E. Overall user experience of the interviewed experts shows that POHST is an easy to use tool.

7.4 Conclusion

The use of modular programming, unit testing, some integral testing and animation have definitely shown their value. These methods aided in checking and maintaining the consistency throughout the model and helped identify any remaining unintended behaviour of early versions of POHST. Experts confirmed that the model is well constructed and that it behaved as intended by the model design.

Validation of the proof of concept is very difficult. Foremost the lack of quantitative outcomes prevented the use of traditional thorough validation methods. Therefore, expert validation was applied. The experts identified the strengths and weaknesses of POHST and overall stated their confidence in the quality of the tool. Especially the use of POHST as a tool to generate start solutions was advised strongly.

The user experience of interviewed experts showed that POHST is a simple and easy to use tool, that provides valuable visualisation. The great strength of the visualisation is that it shows the order in which the hubs are created, which is an indication of their significance.

The verification and validation combined with the user experiences showed that as far as it can be proved in its current form, the proof of concept is valuable and provides high quality outcomes. As soon as POHST is developed to incorporate the entire model design, quantitative validation tests should be performed to confirm the expert validation conclusions.

With the conclusion of the verification and validation established, the overall conclusions can be drawn. The next chapter will answer all the research questions and then focus on the final question: *How well suited is multi-agent modelling to solve the hub location problem in express networks?*

Chapter 8

Conclusion and recommendations

The design showed that the HLP can definitely be solved with MAS. The application of a stand-alone part of the entire model design as proof of concept shows the potential of multi-agent modelling in the preliminary phase of network research. Based on the design and proof of concept, this chapter first lists the experienced benefits followed by the main challenge of using multi-agent modelling to solve the HLP. Next, these deliberations are concluded. The chapter ends with recommendations for further research. However, before going into the these final conclusions, the other research questions are answered to provide a swift insight into this research.

8.1 Conclusion

Question 1 This research started with analysing how MAS can be applied to express networks. It showed that agent modelling provides a natural way of representing the complexity of interacting nodes. As such, it is suited as an approach of the hub location problem
Question 2 (HLP). The HLP investigates the optimal placement of hubs in an express network. With fixed depot locations it uses information on the volume distributions of these depots to determine optimal hub locations while making sure that parcels can be sent from any location and will be delivered within the set service time. Optimal in this sense means placement of hubs in such a way that the total network cost,
Question 3 consisting of hub cost and transport cost are as low as possible. The inputs of the designed model are the depot locations, their volume distributions, the driving times between nodes, hub cost and transport cost. The outputs are the network cost, the number of hubs, the locations of these hubs and the routes that all parcels will take.

Question 4 The agent model that is designed to solve the HLP consist of three main phases. Phase 1 is responsible for creating hubs based on volume distributions. Hubs are placed in regions that have a lot of parcels to be transported between them. Phase 2 creates routes via the hubs that resulted from phase 1. Although where possible the most efficient routes are calculated, the main focus of this phase is to create routes for every parcel in the first place. During Phase 3 the main focus is cost reduction through reducing air transport cost, reducing road transport cost and reducing hub cost.

The time frame of this research did not allow for a full implementation of the designed model. Therefore, a proof is concept (named POHST) is made to show the quality *Question 5* of the design. POHST is the implementation of the first part of Phase 1 and hence creates hubs based on volume distribution. A graphical user interface is added to make a tool out of POHST. Experts have confirmed that it could be used to create *Question 6* insight in the data of express networks and to create start solutions. Although the type of model inputs did not allow for thorough integral testing, other verification *Question 7* methods have shown that the model is behaving as intended by the model design. Since the model has no quantitative output, it was not possible to validate the model using traditional validation methods. The decision is made to use expert validation *Question 8* and hence experienced programmers and analysts are asked to value POHST. Many recommendations have been given for further improvement and the consensus on the current version was that POHST proved its usability.

The design process has further revealed interesting benefits and drawbacks of using *Question 9* the agent paradigm to tackle the HLP. These experiences are set forth in the following paragraph. It should be noted that the advantages and disadvantages experienced in this research apply to this specific design. It cannot be proved that these conclusions will hold for other multi-agent models to be designed for the hub location problem. On the other hand, it definitely provides a good indication of the possibilities of MAS for network designs.

8.1.1 Experienced benefits of MAS applied to the HLP

There are several benefits of the use of multi-agent modelling for the HLP. One of the major advantages of using the agent paradigm is that it provides a natural way of modelling interactions. The decisions made by the different nodes are not incredibly complex, but these decisions all influence each other. This cascading effect leads to quite complex interactions and thus a complex model. The agent paradigm showed to be well suited to this issue. It provided an accessible way of defining and addressing design challenges. Put differently, the structure of the problem really suits the representation in agents.

Scalability of the solution is another positive side to this technique. The model design entails the definition of only two types of agents, agents that represent the nodes (both depots and hubs) and agents that contain logic to recognise situations that are eligible for optimisation. These two agent types are enough to tackle hub location problems of any size in terms of number of nodes[1].

Next, the design shows that multi-agent modelling allows for easy adjustments of agents to local circumstances. An important characteristic of the HLP is that it is never certain whether it is beneficial to change the current network configuration, for

[1] Functionally the presented design can tackle any number of nodes. However, technical problems like CPU capacity, memory capacity or practical issues like extensive run times may and at a certain point most probably will occur.

instance by closing a hub, without calculating all implications of the new situation. However, since the hub location problem is classified as NP-hard[2], it is practically impossible to calculate all the possible configurations. Therefore, indications based on analysis and experience of experts are used to estimate the implications of a certain local change. The argument made here is that a multi-agent model provides an easy and more detailed way of gathering local information to make high quality estimations.

Another advantage is the possibility of thorough statistics gathering. Where other methods often select a number of depots to become hubs and then test whether the network cost are lowered, using agents leads to decision making in a distributed way. The consequence of evolving the network configuration through local decision making is the possibility of logging these decisions and the circumstances under which these are taken. This provides a valuable knowledge base that can be used to trace the reasons behind the decisions made by the model. This is especially useful in cases where the model produces a configuration that is the most efficient one found so far, but has a configuration that is not immediately appealing to the analyst. The logging can be used to supply information explaining which benefits are resulting from the different decisions that were made.

Using multi-agent modelling furthermore facilitates distributed computing. Although not necessary for this proof of concept, increasing problem sizes require an increasing amount of computing power. On a single computer the run time can quickly rise beyond practical use. Distributed computing allows the model to be spread out over multiple computers utilising local computing power. Apart from the fact that distributed computing might be a necessity, it is a benefit in general. Since running a single scenario takes less time with more computing power at hand, more scenarios can be explored in the same amount of time.

Finally, the expert session clearly showed that the tool was appreciated as intended: as a tool which assists in the preliminary phase of network research by generating starting solutions and aiding the data gathering process. Although this argument cannot be based on quantitative facts, this qualitative appreciation does show promise for an implementation of the entire model design in a multi-agent paradigm.

8.1.2 The main challenge

Besides the experienced benefits, there are also some challenges in modelling the HLP from a multi-agent perspective. The multi-agent paradigm focusses on solving problems locally. However, the general principal in the HLP is that a change in the network should only happen if it reduces the overall cost. This has shown to result in the fact that every node should have knowledge of the entire network, which results from the tightly coupled character of an express network. A small change in one part of the network can have a number of consequences elsewhere in the network. To make the right decision using a multi-agent paradigm, the consequences in terms of the cost of all changes in the network that are caused by this decision,

[2] NP-hard stands for non-deterministic polynomial-time hard, which forms a class of problems in computational complexity theory.

need to be considered and communicated. The communication required to facilitate this information sharing may quickly explode, resulting in long run time. Storing the current network configuration in every node could fix this, but that would not cherish the multi-agent modelling values of local data views. This is an indication that multi-agent modelling might be less suited for the use in tightly coupled systems where the benefit of a decision depends on the change in a value aggregated over the entire system. However, this is based on the design presented in this research. There is no proof that it is impossible to accomplish a global optimum with decisions based on local knowledge, it is merely expected based on this research that it will be a greater challenge in tightly coupled systems with a globally aggregated variable of interest.

8.1.3 MAS to solve the HLP

Now that the advantages and disadvantages of multi-agent modelling to solve the hub location problem are given, the last of the research questions should be answered: *How well suited is multi-agent modelling to solve the hub location problem in express networks?* On one hand, the model design definitely showed that multi-agent modelling is able to solve the HLP. In addition, POHST showed that it functioned nicely in its limited scope of modelling the first step of the first phase of solving the HLP. On the other hand, without the implementation of the entire model design, it is difficult to answer this question. However, it is certain that a major challenge lies ahead when implementing the entire model design into a computational model to prevent excessive communication between agents, because this can lead to long run time. On the other hand, it is also certain that many interesting properties of agents can be further investigated and subsequently add great value to compete with and possibly defeat current HLP models.

8.2 Recommendations for further research

Although this research aimed to analyse and execute as much as possible, the time frame of this research did not allow for an in-depth research into every interesting aspect related to research problem. As such, many interesting subjects for further research remain. This section gives some recommendations to such further research.First of all, the most interesting and important recommendation is to implement the entire model design as presented in this research. A fully implemented model can then be compared with existing solutions and be judged on its added value as approach to the HLP. An especially attractive characteristic of agent modelling is the relative ease with which user experience can be incorporated in the agents. For instance, the local data viewing by agents provides the possibility to judge and improve very specific situations. A way in which experts add value is by recognising specific inefficiencies and suggesting ways to resolve those, based on experience. It would be very interesting to research the effect of incorporating more of such knowledge into the model; effectually combining solid mathematical knowledge with the softer factor of human experience.

This research has assumed a single definition of the environment variables applying to the entire model. However, it is again the local data viewing characteristic of agents that allows for different environmental influences. These could, for example, be based on geography. A subject that could be explored is combining data of several countries. Since borders impose restrictions on a national level, one can expect that joining nationally optimised networks of an express service provider in reality results in a sub-optimal solution on the level of a continent. For instance, holidays often differ between bordering countries resulting in situations where country A has a day off while country B is operational. In case their networks are intertwined, this may lead to unforeseen inefficiencies when analysis is only performed on national level. Connecting national agent models to form a continent-wide active agent model creates the possibility to account for such border crossing dissimilarities.

This research has provided and extensive technical network design, but the model output could be improved by incorporating the desires, intentions and beliefs of involved stakeholders. A final recommendation is therefore to research the feasibility of incorporating political forces in the model. Reality shows that solutions are hardly ever implemented in the exact same configuration as was advised by a study. In all cases, goals and interest of employees and managers within a company influence to what extend the proposed network configuration can be implemented as such. This would not be an easy task, but the strong definition of agents allows for such mechanisms. Since the strong definition of agents is merely an extension on the weak definition used in this research, the presented model can be extended instead of changed to provide the suggested improvement.

The stop condition of the entire model design is not determined yet, as explained in Chapter 4. Further research should determine what factors the stop condition should be defined by and what a suited stop condition would be for the proposed model. It is expected that the stop condition will be based on two main factors, time and convergence. As long as the model is converging the solution towards the optimum the model should keep running. However, run time should still be acceptable. Hence, optimisation should stop when either the acceptable run time is exceeded or the solution is not converging fast enough. Further research should look into these conditions to test and further specify them.

With the final conclusions drawn and recommendations on further research given, the final chapter can provide a reflection on the performed research. Not only will it reflect on the research process and the conclusions drawn, but it will also reflect on the challenges posed on a personal level.

Chapter 9

Reflection

This research has posed some interesting challenges. Challenges related to research questions and their answers and on a personal level. This chapter briefly reflects on both types of challenges and thus reflects on the research process and conclusions that are drawn.

To start with the general conclusion, it states that the potential of using multi-agent modelling to approach the HLP is shown and that POHST has proven to be viable as generator of start solutions. However, the most interesting question remains partly unanswered. Is MAS suited to approach the HLP? Hence, can it not only solve the problem, but do this in an efficient way as well? Although this was not mentioned in the research, because of the limited time frame to prove this, it is expected that the hub location problem could be solved even more efficiently when traditional methods of mathematical modelling are combined with agent based modelling. Traditional methods can find a globally optimal start solution with generalised global settings. Then these outcomes can be used in a multi-agent system to optimise local situations. Because agents use local data views (as opposed to a central data view), they excel at adapting to local situations. It is even possible to assign different actions to different agents, based on their local circumstances. This could tailor the solution to be better fitted to the real-life circumstances. With a hybrid system the strengths of both perspectives are exploited while minimising their weaknesses.

Secondly, the methodology posed interesting challenges, but it also helped significantly in designing my first ever multi-agent system. The main challenge of using Prometheus is the lack of practical guides that quickly teach how to use the methodology. Prometheus is designed at RMIT in Melbourne where courses are given to learn how to use it and guides serve a supportive function. However, during the research it became clear that only limited guides were available. In addition, these guides often used outdated terms and examples that did not fit the latest versions of the Prometheus design tool. This made it very challenging to find out how to use the strengths of the methodology. Fortunately, once things became clear, Prometheus helped a great deal in checking the consistency of the design, making modelling decisions, respecting the agent paradigm and verifying the model through code generation. In case one is prepared to take the time to delve into this methodology, I would definitely recommend its usage, because it will certainly raise the quality of ones design.

Another major challenge was the size of the research problem to begin with. The definition of the hub location problem is deceivingly simple. And although I was warned for the ambition level of this research, it still proved to be slightly more challenging than anticipated. Especially the tightly coupled character of the problem manifested itself in more ways than first anticipated, posing many technical design challenges. The benefit of tackling such a complex problem, is that it motivates every day to solve a complex puzzle. As with every problem, as soon as you found the solution, it becomes less interesting. The hub location problem has kept me interested all through the research and it still does.

Although the model has gone through a verification- and validation process, it is not a fully verified and validated model. The model is not completely guaranteed to work without any errors or with a 100% certainty, because this certainty can simply not be provided. However, I do believe in the quality of the design and as such, I am eager to see the results if it gets fully implemented.

It is hard to tell how much time is needed to complete the implementation of the entire model design as described in Chapter 4. Looking at the speed at which I was able to implement the model in JACK and the experience I got from it, I would say I would probably be able to fully implement the model to a working version in two to three months. However, with increasing model size it will become increasingly difficult to find and fix any bugs or other unintended behaviour. Moreover, time to write a report or manual is not included. In the end, the best way to find out, will be to actually implement the model.

References

Abdinnour-Helm, S., & Venkataramanan, M. (1998). Solution approaches to hub location problems. *Annals of Operations Research*, *78*, 31-50.

Alumur, S., & Kara, B. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, *190*(1), 1 - 21.

Beasley, J. E. (1990a). *OR-Library.* Retreived September 12, 2011 from http://people.brunel.ac.uk/~mastjjb/jeb/info.html.

Beasley, J. E. (1990b). OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, *41*(11), 1069-1072.

Benders, J. (2005). Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science*, *2*(1), 3-19.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and MultiAgent Systems*, *8*(3), 203–236.

Çetiner, S., Sepil, C., & Süral, H. (2010). Hubbing and routing in postal delivery systems. *Annals of Operations Research*, *181*, 109-124.

Chvatal, V. (1979). A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, *4*(3), 233–235.

Claes, R., Holvoet, T., & Van Gompel, J. (2010). Coordination in hierarchical pickup and delivery problems using delegate multi-agent systems. In *Proceedings of the 4th workshop on artificial transportation systems and simulation,* (p. 1-7).

Dam, K., & Winikoff, M. (2003). Comparing agent-oriented methodologies. In *Agent-oriented information systems.* Melbourne, Australia: Springer.

Davidsson, P., Henesey, L., Ramstedt, L., & Törnquist, J. (2005). Agent-based approaches to transport logistics. *Transportation Research Part C*, *13*, 255-271.

DeLoach, S. A., & Wood, M. F. (2001). Developing multiagent systems with agenttool. In *Proceedings of the 7th international workshop on intelligent agents vii. agent theories architectures and languages* (pp. 46–60). London, UK: Springer-Verlag.

DeLoach, S. A., Wood, M. F., & Sparkman, C. H. (2001). Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*, *11*, 231-258.

Enserink, B., Koppenjan, J., Thissen, W., D.P., K., & Bekebrede, G. (2003). *Analyse van complexe omgevingen.* Delft, The Netherlands: Faculty of Technology, Policy and Management. (Course manual of spm2110)

Everett, H. (1963). Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, *11*(3), 399–417.

Gascueña, J., & Fernández-Caballero, A. (2009). Agent-based modeling of a mobile robot to detect and follow humans. In A. Håkansson, N. Nguyen, R. Hartung, R. Howlett, & L. Jain (Eds.), *Agent and multi-agent systems: Technologies and applications* (Vol. 5559, p. 80-89). Springer Berlin / Heidelberg.

Glover, F. (1989). Tabu search - part I. *Informs Journal on Computing*, *1*, 190–206.

Glover, F. (1990). Tabu search - part II. *Informs Journal on Computing*, *2*, 4–32.

Hekmatfar, M., & Pishvaee, M. (2009). Hub location problem. In W. A. Mller, M. Bihn, R. Zanjirani Farahani, & M. Hekmatfar (Eds.), *Facility location* (p. 243-270). Physica-Verlag HD.

Holland, J. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* Ann Arbor, MI, USA: University of Michigan Press.

Howden, N., Ronnquist, R., Hodgson, A., & Lucas, A. (2001). JACK intelligent agents - summary of an agent infrastructure. In *Proceedings of the 5th acm international conference on autonomous agents.* Canada.

Jaillet, P., Song, G., & Gang, Y. (1996). Airline network design and hub location problems. *Location Science*, *4*(3), 195–212.

Kara, B., & Taner, M. (2011). Hub location problems: The location of interacting facilities. In H. A. Eiselt & V. Marianov (Eds.), *Foundations of location analysis* (Vol. 155, p. 273-288). Springer US.

Kara, B. Y., & Tansel, B. c. (2001). The latest arrival hub location problem. *Management Science*, *47*, 1408–1420.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, *4*(4), 373–395.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. Miller & J. Thatcher (Eds.), *Complexity of computer computations* (pp. 85–103). Plenum Press.

Keeney, R. (1992). *Value-Focused Thinking. A Path to Creative Decision Making.* Cambridge, MA, USA: Harvard University Press.

Kimms, A. (2006). Economies of scale in hub & spoke network design: we have it all wrong. In M. Morlock, C. Schwindt, N. Trautmann, & J. Zimmermann (Eds.), *Perspectives on operations research* (p. 293-317). Weisbaden, Germany: DUV.

Kleijnen, J. P. C. (1995). Verification and validation of simulation models. *European Journal of Operational Research*, *82*(1), 145-162.

Klincewicz, J. (1992). Avoiding local optima in the p-hub location problem using tabu search and grasp. *Annals of Operations Research*, *40*, 283-302.

Kwon, R. H. (2005). Data dependent worst case bounds for weighted set packing. *European Journal of Operational Research*, *167*(1), 68 - 76.

Labrou, Y., Finin, T., & Peng, Y. (1999). The current landscape of agent communication languages. *IEEE Intelligent Systems*, *14*(2), 45–52.

Luck, M., McBurney, P., & Preist, C. (2003). *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing).* Agentlink.

Mirchandani, P. B., & Francis, R. L. (Eds.). (1990). *Discrete Location Theory.* New York: Wiley Interscience.

O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science*, *20*(2), 92 - 106.

O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, *32*(3), 393–404.

ORTEC. (2010). *Company profile.* Retreived February 28, 2011 from http://www.ortec.com/about/company_profile.aspx.

ORTEC Consulting Group. (2010). *OCG within the ORTEC group.* Retreived February 28, 2011 from http://www.ortec-consulting.com/ocg/ocg_about_consulting.aspx.

Padgham, L. (2011). Personal correspondence.

Padgham, L., Thangarajah, J., & Winikoff, M. (2007). Auml protocols and code generation in the prometheus design tool. In *Proceedings of the 6th international joint conference on autonomous agents and multiagent systems.* New York, NY, USA: ACM.

Padgham, L., & Winikoff, M. (2002). Prometheus: a methodology for developing intelligent agents. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems.*

Padgham, L., & Winikoff, M. (2004). *Developing Intelligent Agent Systems: A Practical Guide.* John Wiley and Sons.

Perini, A., & Susi, A. (2004). Developing tools for agent-oriented visual modeling. In G. Lindemann, J. Denzinger, I. Tim, & R. Unland (Eds.), *Multiagent system technologies, Proceedings of the second German conference MATES'04* (pp. 169–182). Berlin, Germany: Springer-Verlag.

ReVelle, C., Eiselt, H., & Daskin, M. (2008). A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, *184*(3), 817 - 848.

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). Pearson Education.

Sardina, S. (2010). *Introduction to PDT.* University Lecture. Melbourne, Australia: RMIT University.

Schrijver, A. (2003). *Combinatorial optimization - polyhedra and efficiency.* Berlin, Germany: Springer-Verlag.

Sharma, D., & Tran, D. (2004). Dynamic scheduling using multiagent architecture. *Computer Science*, *3214*, 476-482.

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, *60*, 51–92.

Sterling, L., & Taveter, K. (2009). *The art of agent-oriented modeling.* Cambridge, MA, USA: The MIT Press.

Thangarajah, J., Padgham, L., & Winikoff, M. (2005). Prometheus design tool. In *Proceedings of the fourth international joint conference on autonomous agents and multiagent systems* (pp. 127–128). New York, NY, USA: ACM.

Tinbergen, J. (1962). *Shaping the World Economy: Suggestions for an International Economic Policy.* New York, NY, USA: The Twentieth Century Fund.

Topcuoglu, H., Corut, F., Ermis, M., & Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms. *Computers & Operations Research*, *32*(4), 967 - 984.

Valencia-Jimenez, J., & Fernandez-Caballero, A. (2006). Holonic multi-agent systems to integrate independent multi-sensor platforms in complex surveillance. In *Proceedings of the IEEE international conference on video and signal based surveillance.* Washington, DC, USA: IEEE Computer Society.

Van Essen, J. (2009). *Heuristics for the hub location and network design problem with a mixed vehicle fleet.* Master thesis, Delft institute of Applied Mathematics, Delft University of Technology. (Faculty of Electrical Engineering, Mathematics and Computer Sciences)

Vlassis, N. (2007). *A concise introduction to multiagent systems and distributed artificial intelligence* (1st ed.). Morgan and Claypool Publishers.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, *10*(2), 115–152.

Wooldridge, M., Jennings, N. R., & Kinny, D. (1999). A methodology for agent-oriented analysis and design. In *Proceedings of the third annual conference on autonomous agents* (pp. 69–76). New York, NY, USA: ACM.

Appendix A

Requirements specification

A.1 Actor analysis

The actor analysis aims to get a clear definition of the problem and the strategic interests of involved actors. An actor analysis is performed investigating Express Service Providers (ESPs) their customers and the OCG that provides consulting services to ESPs.

Actor	**Customers**
Interest	Low cost, high service
Desired goal/ situation	Same or lower cost for same or higher service
Description of current or expected situation and gap	Increasing cost, making it less attractive to mail parcels
Causes	Less volume for ESPs results in higher cost per parcel
Power/ solution possibilities	Switching ESP if they offer lower same or better service for same of lower cost.

Table A.1: Actor analysis: Customers

Actor	**Express Service Providers**
Interest	Higher market share, Higher turnover, higher profits
Desired goal/ situation	Continuity of the company
Description of current or expected situation and gap	Shrinking parcel volumes
Causes	Increasing competition, electronic mail
Power/ solution possibilities	More efficient supply chain to reduce cost and be able to compete with lower prices

Table A.2: Actor analysis: Express Service Providers

Actor	ORTEC Consulting Group
Interest	Offer high quality analysis to help solve problems of their customers
Desired goal/ situation	Keep ESPs as customers
Description of current or expected situation and gap	Lower turnover of ESP clients endangers continuity of current consulting work
Causes	Increased competition in ESP market
Power/ solution possibilities	Aiding in designing the most cost effective network feasible

Table A.3: Actor analysis: ORTEC Consulting Group

The interests, goals, problem perceptions and powers of the actors are expressed in tables A.1, A.2, and A.3. It shows that the goals of the OCG are the same as those of the ESPs. It is in the interest of the OCG that ESPs perform well, because that will increase perspective for more consulting work in the future. The ESPs have similar goals as the customer though some on a higher aggregation level. Customers want lower prices for the offered service. This is in the interest of the ESPs as well, since lower prices increases their competitiveness.

Essentially the highest risk of an express service provider is the competition of other service providers and the best weapon against them is the price that service is offered for. Hence, the main goal of ESPs and the OCG accordingly is to reduce cost of the express network.

Appendix B

Model design

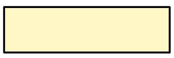
B.1 Prometheus artefacts

Action  Actions are the only way in which an agent can influence the environment.

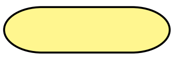
Actor  Actors are any entity outside of the system that is able to interact with the system.

Agent  Agents in the methodology represent agents following the weak agent definition as described in Section 1.2

Capability  Capabilities are a collection of functionality within an agent. Hence, they may contain messages, plans, databases, and again capabilities.

Database  Represents a data store in the model.

Event Actions, percepts and messages together are depicted as event. Events can trigger a plan.

Goal  A goal is something that an agent will try to achieve.

Message  A message is information sent from one agent to another (or in case of internal communication to itself).

Percept

A percept is information coming from out of the system into the system. It could for instance be an observation of an agent or an action of the user of a system.

Plan

A plan describes the way that a certain event has to be handled.

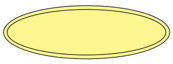
Protocol

A protocol is a conversation of messages. It specifies the agents involved, the messages sent and the order in which these message are transferred.

Role

A role is a collection of goals that represent an ability that the model needs to encompass to meet the design requirements. Often, though not necessarily, roles are directly translated to capabilities of agents.

Scenario



A scenario is a sequence of steps that are an example of a situation that can occur in reality and that should be supported by the model. The use of scenarios is not an essential step in the design process, but aids the modelling process and help identifying goals.

Appendix C

Proof of concept

Figure C.1: Detailed goal overview of proof of concept

The system overview[1] (Figure C.2) shows two types of agents: the Node_Agent and the HubSetup_Agent. The HubSetup_Agent will execute the algorithm based on the region volume distribution that the Node_Agents will produce and send to the HubSetup_Agent.

[1] Appendix B.1 shows an explanation of all Prometheus artefacts and the icons they are represented with.

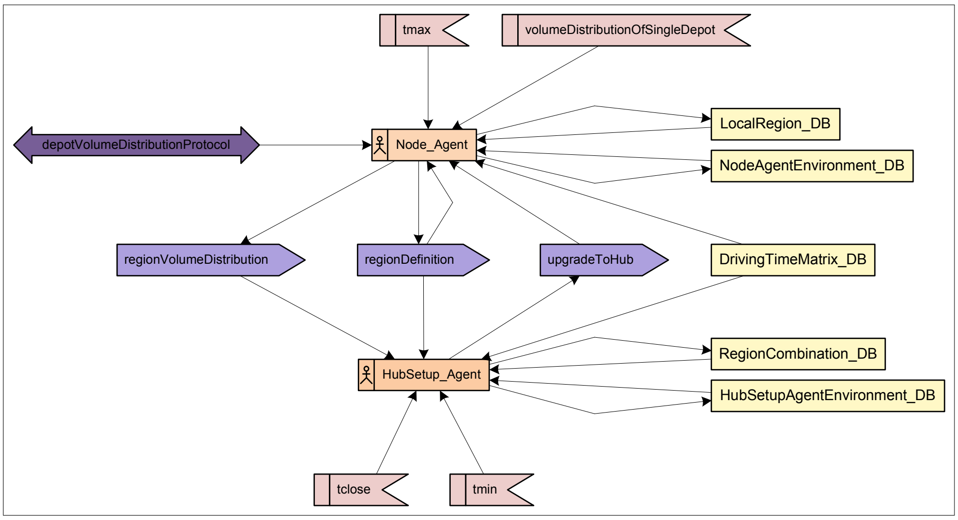


Figure C.2: System overview of communication and database use

The inputs are foremost represented by precepts that are stored locally by the agents in their environment databases. So t^{max} and volume distributions are used by the Node_Agents and t^{close} and t^{min} by the HubSetup_Agent. The driving time matrix is put into a database that is accessible by both agents (as the arrows show).

The local region database contains all data produced by the Node_Agent and the region combination database contains all data produced by the HubSetup_Agent.

The internals of the Node_Agents are illustrated in the agent overview given by Figure C.3. The capabilities match one by one with the goals defined in the before mentioned detailed goal overview. The light purple messages correspond with the messages shown in the system overview. These are messages going into and/or coming out of the agent. The darker purple messages are internal to the agent are mainly used as triggers between capabilities.

The internals of the HubSetup_Agent are shown in Figure C.4. As with the Node_Agent, the capabilities of the HubSetup_Agent also represent goals of the detailed goal overview. The HubSetup_Agents shows three separate sequences. First of all the top one, involving the *UpdatHubSetupAgentEnvironmentDB_P* plan, is used to store the model input relevant for the HubSetup_Agent in its environment database. The second sequence involves receiving and storing the region definitions of the nodes. The HubSetup_Agent will receive such a message from each depot, in which it declares what other depots belong to its region. The third sequence is started every time a region volume distribution is received and stored. In case the just received volume distribution is the final one, the sequence is continued into the optimisation part. Once optimisation is completed, the HubSetup_Agents tells the specific depots that they should upgrade to a hub.

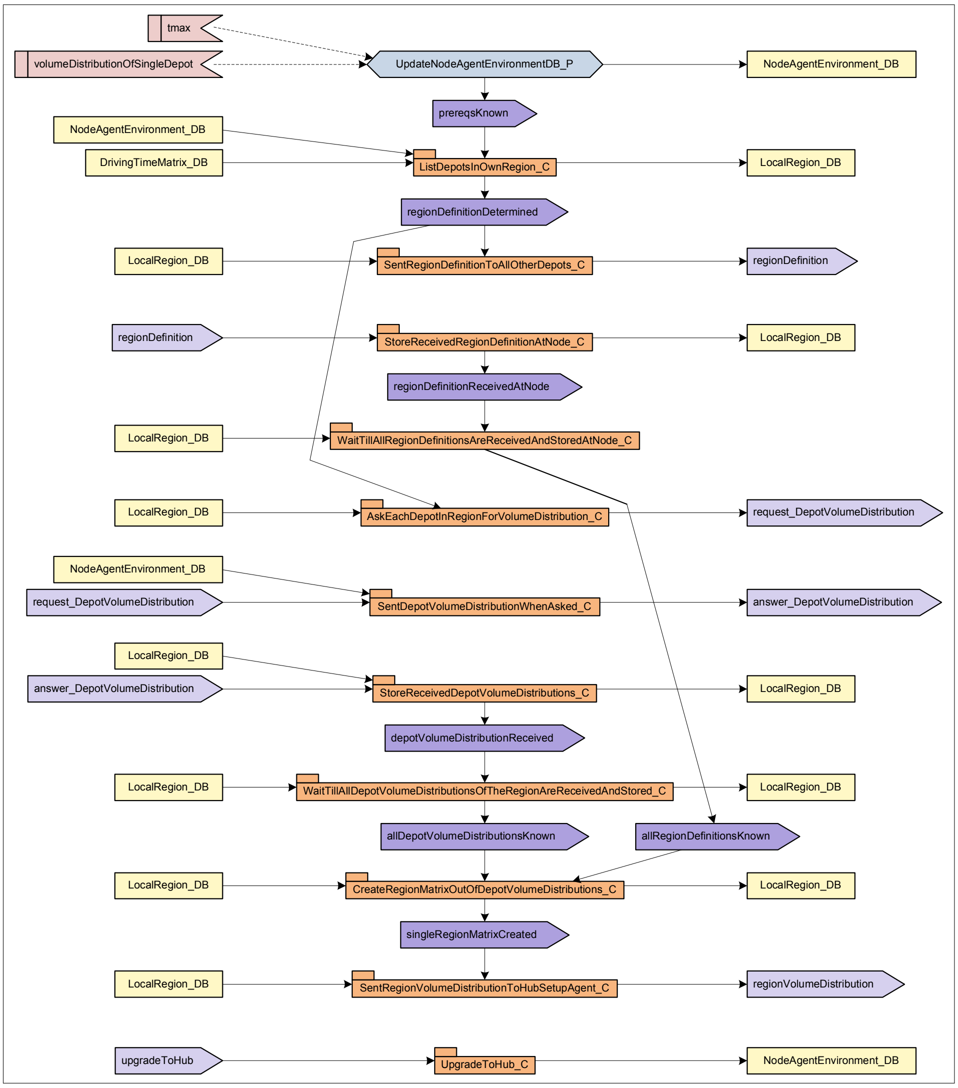


Figure C.3: Agent overview of Node agent

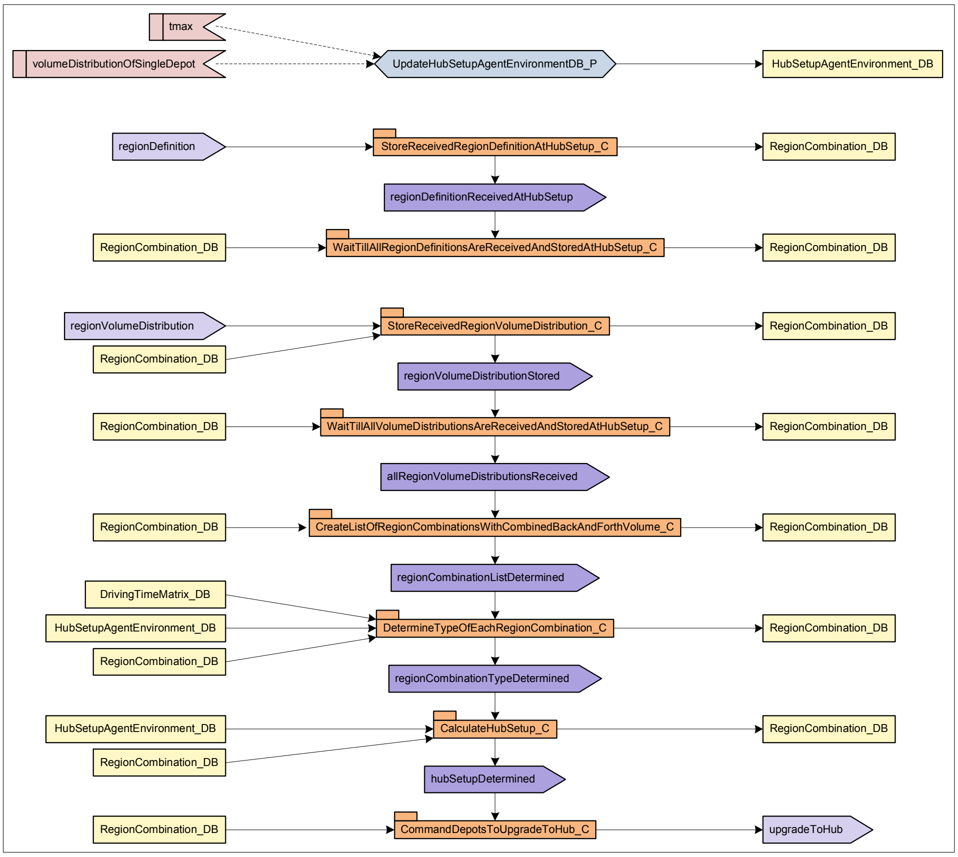


Figure C.4: Agent overview of Hub-Setup agent

Appendix D

POHST application

This chapter describes statistical outcomes of different tests and experiments that were performed with POHST. First, the parameters are varied on both datasets, followed by experiments with different structural settings. One structural experiment is to run the tool with different region settings. The default setting is to let the tool consider close as well as distant regions for opening hubs. This default outcome is compared with runs where only close, and only distant regions are eligible for selection. The other experiment type examines the influence of the greedy algorithm by analysing the different outcomes in case the algorithm is reversed. Reversing the algorithm means that instead of selecting the region combination with the highest volume, it selects the region combination with the least volume to be transported in between the regions.

D.1 Parameter testing

Parameter testing involves varying the input parameters to investigate their influence in the model outcomes. These test are calibrated around the default parameter settings as described in Section 6.2.2. For each parameter, model runs are done starting at 20% of the default value, incrementing the parameter by 20% each run until 200% of the default value is reached. These tests are first described for the CAB data, followed by the TN data.

D.1.1 CAB data

The standard settings for the CAB data are as follows: t^{max} is 450 minutes, t^{min} is 900 minutes and t^{close} is 900 minutes as well. Following the test as explained in the preceding paragraph the parameters will be tested over the following ranges:

- t^{max} will be varied between 90 and 900 minutes.

- t^{min} will be varied between 180 and 1800 minutes.
- t^{close} will be varied between 180 and 1800 minutes.

The figures are scaled exactly identical. Hence, the vertical axle shows the score from 0 to 5 and the size of the graphics are the same to make sure cities are situated exactly at the same place when comparing figures.

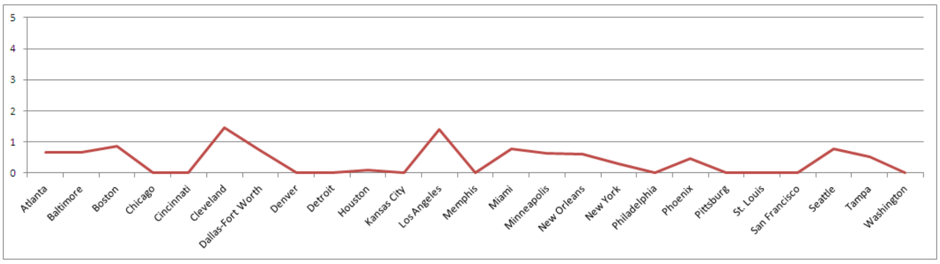


Figure D.1: CAB data: score per hub varying t^{max} from 90 till 900

Figure D.1 shows the total score of each hub summed over 10 model runs while increasing t^{max} from 90 to 900 minutes with increments of 90 minutes. Two hubs clearly outperform the others, namely Cleveland and Los Angeles. These are followed by a group of runner ups and 11 depots that are never chosen to become a hub. Although not shown in the figure, the average number of chosen hubs was 4.9 with an standard deviation of 0.57. This is a very stable outcome in terms of number of hubs. As such, this experiment shows that for the CAB data the amount of hubs created is quite stable, but the hub locations are very sensitive to t^{max}. A total of 14 different hubs are chosen over 10 runs and it shows very few clear winners.

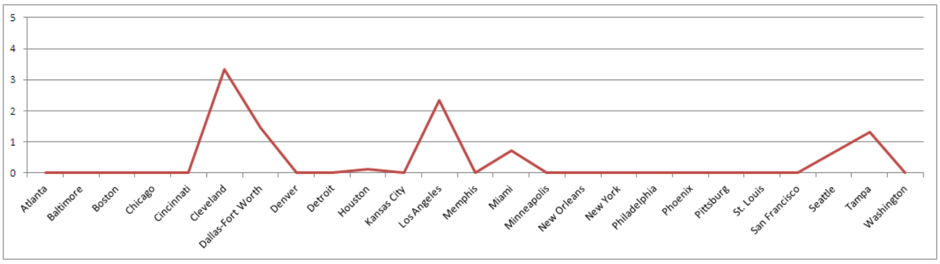


Figure D.2: CAB data: score per hub varying t^{min} from 180 till 1800

Figure D.2 clearly shows better defined winners. Cleveland again has the highest score and even a total score that is twice as high as in the t^{max} experiment. Los Angeles, Dallas-Forth Worth, Tampa and Miami are the other nodes that stand out. These five hubs make up almost every outcome[1] except for one where Houston becomes a hub at the expense of Dallas-Forth Worth). This indicates that t^{min} has no influence on the number of hubs that are chosen, but only on which hubs are chosen.

Figure D.3 shows the most distinct outcome of the three parameter variations. This is due to the high variability in the number of hubs. On average 6.1 hubs are created

[1] Hence, the average number of hubs is 5 and the standard deviation is 0.

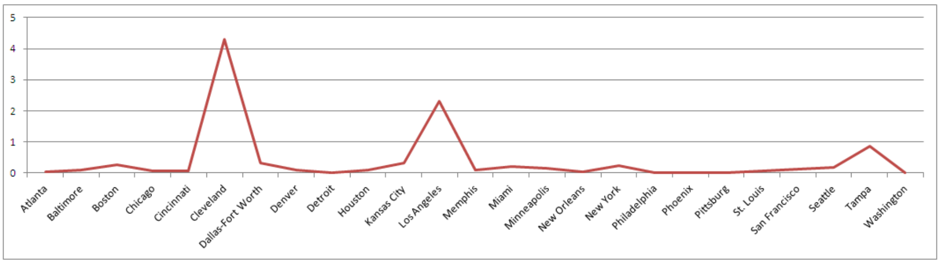


Figure D.3: CAB data: score per hub varying t^{close} from 180 till 1800

with an very high standard deviation of 6.05. The number of hubs in the first run, with t^{close} at its lowest, is 21. Through the rest of the runs this number decreases roughly exponentially until it stabilises on 2 for the last three model outcomes. This decline corresponds to the nature of t^{close}: the higher it is, the larger the radius of eliminating depots around one opened hub. Since 1 point is divided between the opened hubs at each run (as explained in Section 6.2), the last runs have a much greater impact on the total score. At the first run, the cities received on average 1/21th of a point, while at the last three runs an average of 0.5 point was awarded. As such, Cleveland, Los Angeles and Tampa clearly stand out.

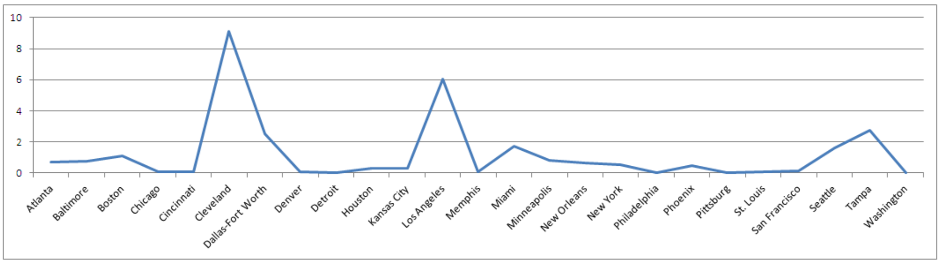


Figure D.4: CAB data: score per hub summed for previous three experiments

When summing the data of the three parameter tests, two nodes stand out as shown in Figure D.4. Clevelend and Los Angeles clearly outperform the others. This does not mean that they should be chosen as hub in every situation, but since they show to be the most constant performers they are a safe choice.

D.1.2 TN data

The standard settings for the TN data are as follows: t^{max} is 90 minutes, t^{min} is 180 minutes and t^{close} is 180 minutes as well. Following the test as explained in at the start of this section the parameters will be tested over the following ranges:

- t^{max} will be varied between 18 and 180 minutes.
- t^{min} will be varied between 36 and 360 minutes.
- t^{close} will be varied between 36 and 360 minutes.

The figures are scaled exactly identical again. The vertical axle shows the score from 0 to 1.6 and the size of the graphics are the same to make sure cities are situated exactly at the same place when comparing figures.

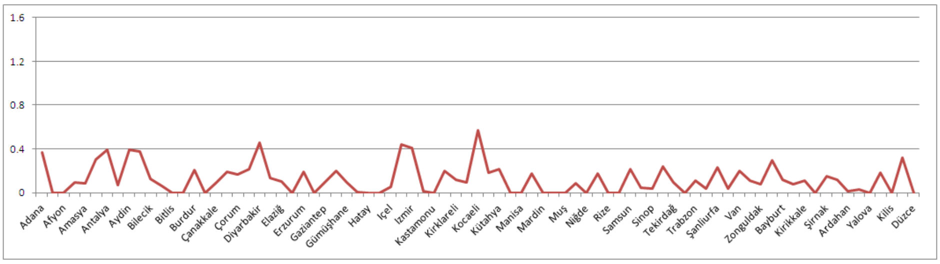


Figure D.5: TN data: score per hub varying t^{max} from 18 till 180

Just as with the CAB data, varying t^{max} gives very different outcomes and hence Figure D.5 shows that many hubs have about the same score. The average number of hubs is 16.4 with a standard deviation of only 0.7. This indicates that the number of hubs is very stable, varying only between 16 and 17 hubs. Of the 81 nodes 23 are never upgraded to a hub and the other 58 nodes divide the scores of all model runs. As before, in the TN data the number of hubs is not sensitive for t^{max} and the chosen hubs are very sensitive for t^{max}.

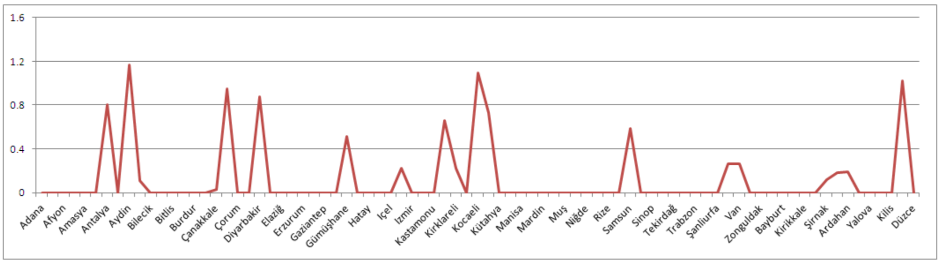


Figure D.6: TN data: score per hub varying t^{min} from 36 till 360

Figure D.6 shows the score of the nodes while varying t^{min}. Two interesting results manifest themselves. First of all, the figure shows that 62 of the 81 nodes are never upgraded to hubs and hence 19 nodes are. Secondly, of these 19 nodes there is not one that stands out clearly. The scores are spread quite evenly. With a stable number of hubs (average is 16.2 hubs with a standard deviation of 0.42) this indicates that t^{min} foremost influences the order in which the hubs are chosen.

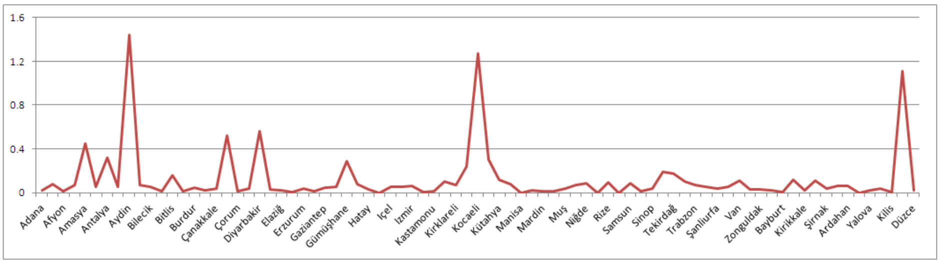


Figure D.7: TN data: score per hub varying t^{close} from 36 till 360

Varying t^{close} again shows clear winners. With the TN data three depots stand out, namely Aydin, Kocaeli, and Osmaniye. Although nearly all the nodes are chosen at least once, Aydin, Kocaeli and Osmaniye score well over all values of t^{close} and foremost when t^{close} is at the higher end of its range. Just as with the CAB data, the number of hubs starts out high at 76 hubs and declines roughly exponentially to just 6 hubs. The average number of hubs created is 24.4 and the standard deviation from that average is 23.56, which is to be expected with the large difference of number of hubs between model runs.

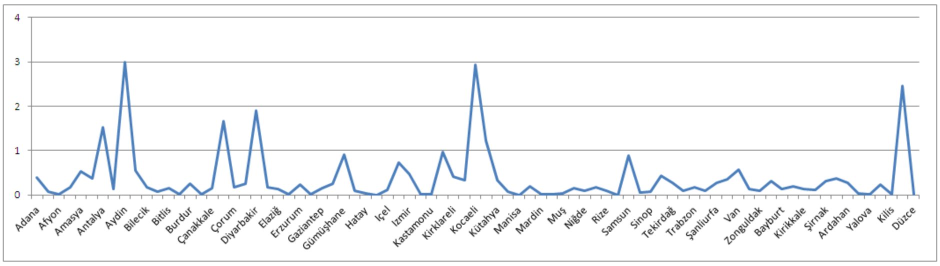


Figure D.8: TN data: score per hub summed for previous three experiments

The overall scores are presented in Figure D.8. The three high performers from the t^{close} test Aydin, Kocaeli, and Osmaniye are the three highest scoring nodes overall as well. Furthermore, three nodes are never chosen at all: Isparta, Manisa, Sakarya. This indicates that these nodes are close to strong hubs, by which they are always dominated.

D.2 Structural experiments

Two structural experiments are performed on the two datasets. First of all, the influence of the greedy algorithm is tested using a reversed algorithm. Secondly, to investigate the influence of close and distant region combinations, runs using only close or distant regions are compared with using both.

D.2.1 Reversed algorithm

In this experiment two runs are done for each dataset with the default parameter settings. The difference between the runs is that the first (indicated by red line) uses the normal greedy algorithm and that the second (indicated by blue line) has the algorithm reversed, hence choosing the region combinations with the lowest volume.

Figure D.9 shows the scores of the two different runs performed on the CAB data. The number of hubs does not differ much. The normal algorithm results in 5 hubs and the reversed algorithm in 4 hubs. This is the consequence of using the same parameter settings.

Figure D.10 is added to show that although the two runs have almost the same amount of hubs, the nodes they select to become hubs are very different. The figure shows the absolute number of times that a nodes is upgraded to a hub in both runs.

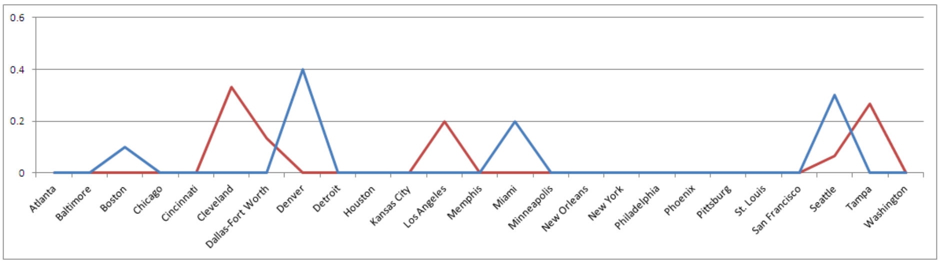


Figure D.9: CAB data: score per hub normal versus reversed algorithm

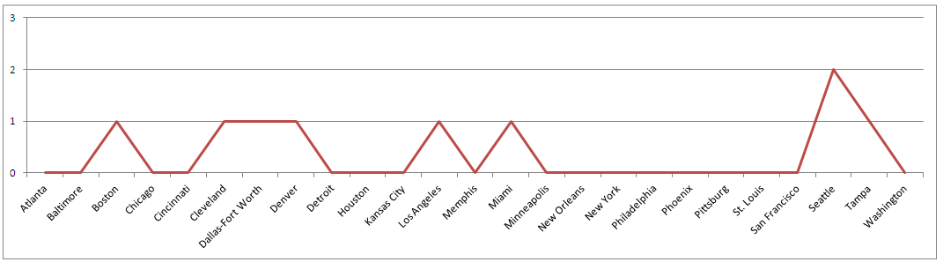


Figure D.10: CAB data: selections per hub normal versus reversed algorithm

Only Seattle is chosen in both solutions, most probably because it is isolated and therefore will not be eliminated very quickly. Apart from Seattle, all chosen hubs differ between the two runs.

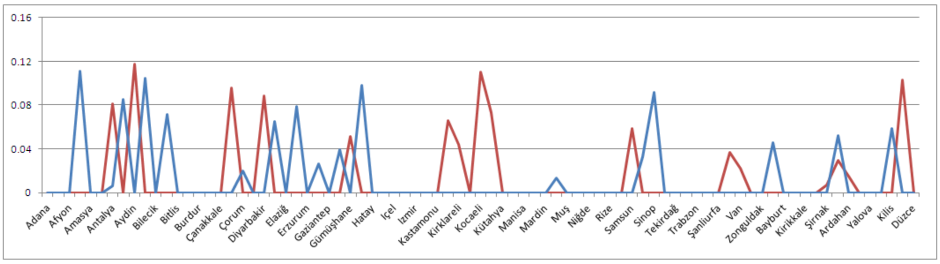


Figure D.11: TN data: score per hub normal versus reversed algorithm

The TN data shows a similar result as with the CAB data (see Figure D.11. The normal algorithm run results in 16 hubs and the reversed algorithm in 17 hubs. Again, the number of hubs is almost the same, confirming this is foremost influenced by the parameters and not by the algorithm.

The absolute number of times that nodes are upgraded to hubs in both model runs are presented in Figure D.12. Again almost all hubs are different between the two solutions. Only Antalya and Bartin are chosen in both model runs.

In conclusion, this experiment has shown that the number of hubs is not influenced much by the algorithm. The fixed parameters roughly maintain the amount of hubs in both situations. However, the hub locations differ tremendously between the two approaches. This proves that choosing region combinations on the basis of the volume to be transported between them has a very significant influence on the resulting solution.

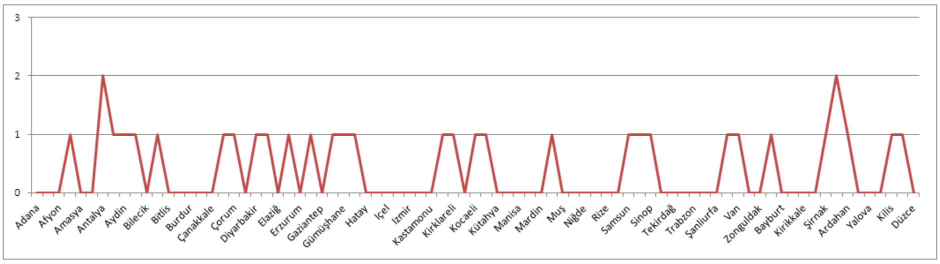


Figure D.12: TN data: selections per hub normal versus reversed algorithm

D.2.2 Varying region combinations

The second experiment investigates the influence of the use of different types of region combinations. While running POHST with default parameter settings, region combinations that are eligible to be chosen are limited to either only close or only distant regions. This does not mean that all region combinations are set to be of one type (since that would be the same as extreme value testing of t^{min}). It means that when testing with only distant region combinations all region combinations that are marked as close are eliminated before the algorithm starts. The two runs, one with only close and one with only distant region combinations, are then compared to the default.

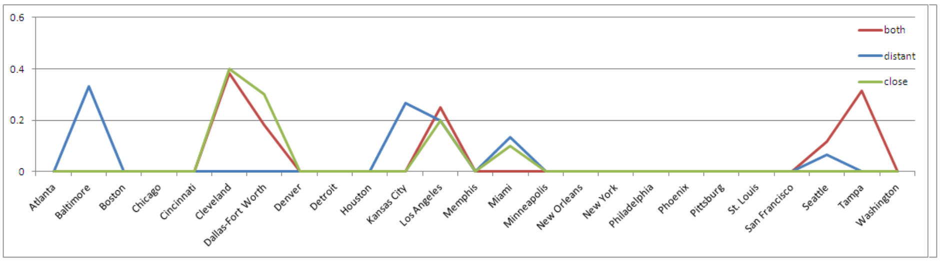


Figure D.13: CAB data: selections per hub using both region types vs. only close and only distant

Figure D.13 shows the score per hub over the three different model runs. The green line depicts the score of the run using only close regions, the blue line depicts the score of the run using only distant regions, and the red line is a run with default settings hence using both regions. The red line is offset by +0.05 points to increase the readability of the graph. This way the graph clearly shows that Los Angeles was selected in all three runs. As such, Los Angeles stands out, because it is the only depot upgraded to hub in every run. Another interesting node is Miami, which is selected in the run using only distant regions and the run using only close regions, although it is not selected when using close and distant regions combined. This confirms that the region combination types have a significant influence on the model outcomes.

Figure D.14 shows the same test with the TN data. The difference between the runs using only close or only distant region combination is evident with this dataset as well. The run using both region combination types is again offset (this time the

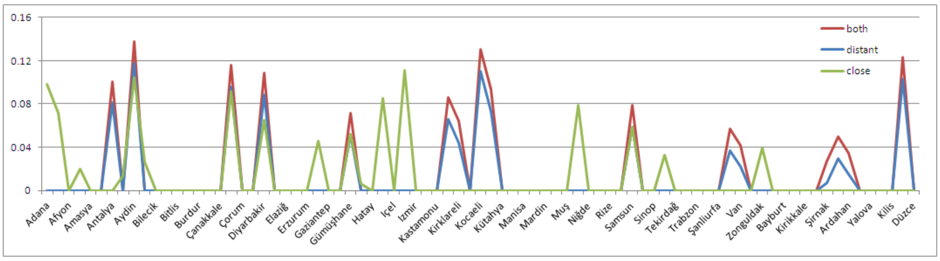


Figure D.14: TN data: selections per hub using both region types vs. only close and only distant

score is offset with +0.02 points) to prevent overlays with the other two runs. The figure clearly shows that the model run using both region combination types and the run using only distant region combinations result in identical outcomes. Through investigating the textual output it becomes clear that in the default run, only distant region combinations are chosen. This indicates that the destinations of most parcels are not close to their origin. Hence, they need to be transported over a larger distance than t^{min} travel time, resulting in hubs arranged accordingly.

In conclusion this experiment confirmed that differing between distant and close regions has a significant impact on which nodes are upgraded to hubs.

D.3 Civil Aeronautics Board Data

The Civil Aeronautics Board (CAB) data was originally introduced by O'Kelly (1987). The data is not fully presented in that paper and is therefore retrieved from the online OR-Library (Beasley, 1990a) which in turn was originally described in Beasley (1990b). The original dataset is interpreted to suit this research. The original data consists of two values for each node combination. The number of units of flow between nodes i and j is depicted with W_{ij}. The units of flow in Okelly's original paper were airplane passengers. To suit the HLP the number of passengers is interpreted as the number of (equally sized and weighted) parcels. The second value, C_{ij}, originally depicted the transportation cost of a unit of flow between nodes i and j. For this research these cost are rounded to the nearest integer and interpreted as minutes of travel time between the corresponding nodes. The consequence of using the exact data is twofold. First of all it means that travel times do not exactly correspond with the real-life case and that parcel volumes cannot be considered representative of real American supply and demand. Secondly, it however does provides the opportunity to compare results with earlier research that used the CAB data, as soon as the model includes at least Phase 2 of the design as well, such that network cost can be determined.

First the table with the driving times between the nodes is presented (Table D.1), followed by the volume distributions between the nodes (Table D.2). Thirdly, the table with geocode information is shown (Table D.3). The geocodes are not original but personally derived by looking up the city names mentioned by O'Kelly (1987) in

Google Maps[2].

[2] http://maps.google.com

City name	*Node id*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Atlanta	1	0	577	943	598	374	560	709	1208	604	695	681	1937	332	593	909	426	756	673	1590	527	483	2141	2184	408	541
Baltimore	2	577	0	370	613	429	313	1196	1502	406	1242	960	2318	787	950	939	1000	179	96	2000	211	736	2456	2340	844	36
Boston	3	946	370	0	858	750	556	1541	1765	621	1603	1251	2600	1137	1267	1125	1368	190	274	2299	494	1043	2703	2504	1189	406
Chicago	4	598	613	858	0	255	311	790	907	237	932	406	1742	486	1187	346	830	720	675	1447	404	256	1854	1733	1006	592
Cincinnati	5	374	429	750	255	0	226	794	1080	239	880	533	1890	402	947	599	700	578	512	1571	256	307	2036	1967	775	399
Cleveland	6	560	313	556	311	226	0	1010	1217	94	1105	695	2047	627	1085	626	922	409	366	1743	105	491	2165	2027	933	299
Dallas-Fort Worth	7	709	1196	1541	790	794	1010	0	664	983	221	448	1250	411	1098	852	424	1363	1289	895	1049	538	1494	1687	912	1162
Denver	8	1208	1502	1765	907	1080	1217	664	0	1144	875	552	842	880	1715	694	1067	1626	1575	593	1302	781	956	1025	1519	1475
Detroit	9	604	406	621	237	239	94	983	1144	0	1095	637	1979	620	1152	535	936	490	453	1682	199	450	2087	1936	992	393
Houston	10	695	1242	1603	932	880	1105	221	875	1095	0	642	1376	477	964	1046	305	1417	1338	1017	1125	677	1650	1891	795	1206
Kansas City	11	681	960	1251	406	533	695	448	552	637	642	0	1358	379	1236	405	674	1097	1039	1049	768	229	1506	1504	1039	932
Los Angeles	12	1937	2318	2600	1742	1890	2047	1250	842	1979	1376	1358	0	1608	2336	1531	1662	2453	2397	358	2126	1582	362	987	2158	2289
Memphis	13	332	787	1137	486	402	627	411	880	620	477	379	1608	0	858	701	348	956	880	1266	651	255	1809	1873	661	751
Miami	14	593	950	1267	1187	947	1085	1098	1715	1152	964	1236	2336	858	0	1501	676	1098	1022	1978	1015	1066	2591	2726	198	923
Minneapolis	15	909	939	1125	346	599	626	852	694	535	1046	405	1531	701	1501	0	1040	1018	988	1281	728	450	1590	1401	1311	922
New Orleans	16	426	1000	1368	830	700	922	424	1067	936	305	674	1662	348	676	1040	0	1178	1096	1304	919	602	1917	2090	496	963
New York	17	756	179	190	720	578	409	1363	1626	490	1417	1097	2453	956	1098	1018	1178	0	84	2144	329	881	2574	2415	1008	216
Philadelphia	18	673	96	274	675	512	366	1289	1575	453	1338	1039	2397	880	1022	988	1096	84	0	2082	273	818	2527	2389	927	133
Phoenix	19	1590	2000	2299	1447	1571	1743	895	593	1682	1017	1049	358	1266	1978	1281	1304	2144	2082	0	1815	1264	662	1129	1800	1969
Pittsburgh	20	527	211	494	404	256	105	1049	1302	199	1125	768	2126	651	1015	728	919	329	273	1815	0	552	2253	2129	875	195
St. Louis	21	483	736	1043	256	307	491	538	781	450	677	229	1582	255	1066	450	602	881	818	1264	552	0	1736	1712	872	707
San Francisco	22	2141	2456	2703	1854	2036	2165	1494	956	2087	1650	1506	362	1809	2591	1590	1917	2574	2527	662	2253	1736	0	695	2405	2430
Seattle	23	2184	2340	2504	1733	1967	2027	1687	1025	1936	1891	1504	987	1873	2726	1401	2090	2415	2389	1129	2129	1712	695	0	2528	2322
Tampa	24	408	844	1189	1006	775	933	912	1519	992	795	1039	2158	661	198	1311	496	1008	927	1800	875	872	2405	2528	0	814
Washington	25	541	36	406	592	399	299	1162	1475	393	1206	932	2289	751	923	922	963	216	133	1969	195	707	2430	2322	814	0

Table D.1: CAB Data: Driving time matrix

City name	*Node id*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Atlanta	1	0	6469	7629	20036	4690	6194	11688	2243	8857	7248	3559	9221	10099	22866	3388	9986	46618	11639	1380	5261	5985	6731	2704	12250	16132
Baltimore	2	6469	0	12999	13692	3322	5576	3878	3202	6699	4198	2454	7975	1186	7443	1162	5105	24817	6532	806	8184	3896	7333	3719	2015	565
Boston	3	7629	12999	0	35135	5956	14121	5951	5768	16578	4242	3365	22254	1841	23665	6517	3541	205088	37669	2885	13200	7116	17165	4284	8085	51895
Chicago	4	20036	13692	35135	0	19094	35119	21423	27342	51341	15826	28537	65387	12980	44097	51525	14354	172895	37305	15418	26221	42303	35303	13618	17580	40708
Cincinnati	5	4690	3322	5956	19094	0	7284	3102	1562	7180	1917	2253	5951	1890	7097	2009	1340	25303	6031	1041	4128	5452	3344	1067	4608	7050
Cleveland	6	6194	5576	14121	35119	7284	0	5023	3512	10419	3543	2752	14412	2043	15642	5014	2016	62034	15385	2957	5035	7482	6758	2191	6599	14181
Dallas-Fort Worth	7	11688	3878	5951	21423	3102	5023	0	11557	6479	34261	10134	27350	6929	7961	4678	13511	29801	7549	5550	3089	9958	14110	4911	2722	10802
Denver	8	2243	3202	5768	27342	1562	3512	11557	0	5615	7095	10753	30362	1783	3437	8897	2509	23273	5160	8750	2583	7288	17481	7930	1278	8447
Detroit	9	8857	6699	16578	51341	7180	10419	6479	5615	0	4448	5076	22463	4783	24609	9969	4224	79945	20001	4291	10604	11925	13091	4172	12891	19500
Houston	10	7248	4198	4242	15826	1917	3543	34261	7095	4448	0	4370	17267	3929	8602	2753	20013	28080	5971	2131	3579	6809	8455	2868	2336	5616
Kansas City	11	3559	2454	3365	28537	2253	2752	10134	10753	5076	4370	0	15287	3083	4092	7701	2809	17291	4462	3239	2309	16003	8381	3033	1755	7266
Los Angeles	12	9221	7975	22254	65387	5951	14412	27350	30362	22463	17267	15287	0	5454	15011	17714	10037	105507	20040	31780	10822	16450	92083	32908	3865	24583
Memphis	13	10099	1186	1841	12980	1890	2043	6929	1783	4783	3929	3083	5454	0	3251	1126	5926	10653	3062	759	1255	6173	2974	1056	1504	4588
Miami	14	22866	7443	23665	44097	7097	15642	7961	3437	24609	8602	4092	15011	3251	0	5550	9473	169397	25073	1170	14272	8543	8064	1840	20618	20937
Minneapolis	15	3388	1162	6517	51525	2009	5014	4678	8897	9969	2753	7701	17714	1126	5550	0	2152	26816	6931	4947	2676	8033	12692	6157	3065	12044
New Orleans	16	9986	5105	3541	14354	1340	2016	13511	2509	4224	20013	2809	10037	5926	9473	2152	0	21806	4519	886	1742	4782	6453	2022	3546	5065
New York	17	46618	24817	205088	172895	25303	62034	29801	23273	79945	28080	17291	105507	10653	169397	26816	21806	0	9040	11139	63153	34092	70935	14957	28398	166694
Philadelphia	18	11639	6532	37669	37305	6031	15385	7549	5160	20001	5971	4462	20040	3062	25073	6931	4519	9040	0	2802	30224	7982	14964	4589	6227	12359
Phoenix	19	1380	806	2885	15418	1041	2957	5550	8750	4291	2131	3239	31780	759	1170	4947	886	11139	2802	0	1869	3716	11510	3519	569	3520
Pittsburgh	20	5261	8184	13200	26221	4128	5035	3089	2583	10604	3579	2309	10822	1255	14272	2676	1742	63153	30224	1869	0	5020	6610	2139	5431	13541
St. Louis	21	5985	3896	7116	42303	5452	7482	9958	7288	11925	6809	16003	16450	6173	8543	8033	4782	34092	7982	3716	5020	0	9942	3276	3820	11799
San Francisco	22	6731	7333	17165	35303	3344	6758	14110	17481	13091	8455	8381	92083	2974	8064	12692	6453	70935	14964	11510	6610	9942	0	35285	2566	19926
Seattle	23	2704	3719	4284	13618	1067	2191	4911	7930	4172	2868	3033	32908	1056	1840	6157	2022	14957	4589	3519	2139	3276	35285	0	940	4951
Tampa	24	12250	2015	8085	17580	4608	6599	2722	1278	12891	2336	1755	3865	1504	20618	3065	3546	28398	6227	569	5431	3820	2566	940	0	6237
Washington	25	16132	565	51895	40708	7050	14181	10802	8447	19500	5616	7266	24583	4588	20937	12044	5065	166694	12359	3520	13541	11799	19926	4951	6237	0

Table D.2: CAB Data: Volume distribution matrix

City name	*Node id*	*Latitude*	*Longitude*
Atlanta	1	33.751748	-84.396973
Baltimore	2	39.296048	-76.618652
Boston	3	42.358544	-71.059570
Chicago	4	41.877741	-87.632446
Cincinnati	5	39.100226	-84.517822
Cleveland	6	41.500350	-81.699829
Dallas-Fort Worth	7	32.805745	-96.778564
Denver	8	39.740986	-104.985352
Detroit	9	42.334184	-83.045654
Houston	10	29.754840	-95.372314
Kansas City	11	39.104489	-94.581299
Los Angeles	12	34.052659	-118.251343
Memphis	13	35.155846	-90.065918
Miami	14	25.790000	-80.227661
Minneapolis	15	44.980342	-93.273926
New Orleans	16	29.959694	-90.071411
New York	17	40.713956	-74.009399
Philadelphia	18	39.951859	-75.173950
Phoenix	19	33.449777	-112.077026
Pittsburgh	20	40.442767	-79.996948
St. Louis	21	38.629745	-90.203247
San Francisco	22	37.770715	-122.420654
Seattle	23	47.606163	-122.332764
Tampa	24	27.950739	-82.457886
Washington	25	38.895308	-77.036133

Table D.3: CAB Data: Geocodes

D.4 Turkish Network Data

The Turkish Network (TN) data was introduced by Çetiner et al. (2010) where it was used to analyse the Turkish postal services (PTT). This TN data contains 81 major Turkish cities. Consisting of 81 nodes, this dataset can be considered to be very large. As with the CAB data, the geocodes for the TN data were manually collected using the names in the original datasets and Google Maps (http://www.maps.google.com).

Node id	1-20	21-40	41-60	61-81
1-40	A	B	C	D
41-81	E	F	G	H

Table D.4: Turkish network data: Table split up syntax

City name	*Node id*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ADANA	1	0	219	382	641	409	327	372	687	596	596	512	419	485	451	447	558	729	384	383	512
ADIYAMAN	2	219	0	601	432	424	503	591	503	815	815	705	233	276	631	667	751	931	523	467	731
AFYON	3	382	601	0	872	393	171	195	824	234	214	140	735	863	279	113	182	347	258	332	150
AĞRI	4	641	432	872	0	490	701	953	265	1102	1045	905	239	156	763	953	945	1125	655	551	1018
AMASYA	5	409	424	393	490	0	223	583	463	625	555	415	424	553	273	504	455	635	165	61	541
ANKARA	6	327	503	171	701	223	0	363	653	402	353	209	599	728	127	281	255	435	87	161	318
ANTALYA	7	372	591	195	953	583	363	0	977	229	340	316	791	857	455	81	358	473	450	521	148
ARTVIN	8	687	503	824	265	463	653	977	0	1055	975	834	271	375	692	935	874	1055	596	492	971
AYDIN	9	596	815	234	1102	625	402	229	1055	0	195	349	949	1077	477	181	295	299	489	563	84
BALIKESIR	10	596	815	214	1045	555	353	340	975	195	0	164	945	1074	283	265	101	133	439	515	192
BILECIK	11	512	705	140	905	415	209	316	834	349	164	0	801	929	142	235	63	244	296	370	265
BINGÖL	12	419	233	735	239	424	599	791	271	949	945	801	0	131	697	800	847	1027	589	485	865
BITLIS	13	485	276	863	156	553	728	857	375	1077	1074	929	131	0	825	929	975	1156	718	614	993
BOLU	14	451	631	279	763	273	127	455	692	477	283	142	697	825	0	373	182	363	157	235	403
BURDUR	15	447	667	113	953	504	281	81	935	181	265	235	800	929	373	0	277	398	369	443	100
BURSA	16	558	751	182	945	455	255	358	874	295	101	63	847	975	182	277	0	181	339	416	291
ÇANAKKALE	17	729	931	347	1125	635	435	473	1055	299	133	244	1027	1156	363	398	181	0	519	597	325
ÇANKIRI	18	384	523	258	655	165	87	450	596	489	439	296	589	718	157	369	339	519	0	104	405
ÇORUM	19	383	467	332	551	61	161	521	492	563	515	370	485	614	235	443	416	597	104	0	479
DENIZLI	20	512	731	150	1018	541	318	148	971	84	192	265	865	993	403	100	291	325	405	479	0
DIYARBAKIR	21	345	137	727	295	468	607	717	367	941	941	808	96	139	734	793	854	1035	615	511	857
EDIRNE	22	779	959	455	1091	601	455	613	1020	439	273	318	1025	1153	328	537	280	145	485	563	465
ELAZIĞ	23	324	189	640	331	366	505	696	363	854	851	706	95	223	632	705	752	933	513	409	770
ERZINCAN	24	450	365	625	247	243	455	706	271	855	799	658	181	309	516	707	698	879	409	305	771
ERZURUM	25	536	353	751	123	369	581	832	151	981	925	784	120	233	642	833	824	1005	535	431	897
ESKIŞEHIR	26	459	651	96	857	378	155	283	809	322	198	53	747	876	192	201	99	280	243	317	238
GAZIANTEP	27	137	100	519	504	406	447	509	575	733	733	649	305	348	575	584	695	866	467	420	649
GIRESUN	28	485	475	577	364	216	407	741	247	809	728	587	363	474	445	688	627	808	349	245	725
GÜMÜŞHANE	29	524	453	672	256	290	501	780	222	903	836	695	255	366	553	781	735	916	455	351	819
HAKKARI	30	600	445	982	289	761	911	972	514	1196	1196	1112	339	227	1033	1047	1158	1329	926	822	1112
HATAY	31	127	213	509	635	470	454	499	681	723	723	639	413	479	579	575	685	857	511	484	639
ISPARTA	32	413	633	113	919	503	281	87	934	195	264	234	766	895	373	34	276	397	368	442	111
IÇEL	33	46	265	377	687	426	322	326	733	555	591	507	465	531	447	407	553	724	379	379	474
ISTANBUL	34	626	805	301	937	447	302	477	867	454	260	165	871	1000	175	396	162	213	331	409	426
IZMIR	35	600	819	218	1087	609	386	297	1039	84	115	278	953	1081	397	249	215	217	473	547	149
KARS	36	671	488	885	145	503	714	965	140	1115	1058	917	255	288	775	966	957	1138	668	564	1031
KASTAMONU	37	460	592	334	658	168	163	526	576	565	447	306	592	721	164	445	346	527	76	130	481
KAYSERI	38	222	291	347	541	232	212	412	565	561	558	413	387	516	339	413	459	640	232	185	477
KIRKLARELI	39	767	946	442	1078	588	443	618	1007	450	284	305	1012	1141	315	537	291	156	472	550	476
KIRŞEHIR	40	250	381	286	627	205	123	381	635	517	469	324	477	605	250	382	370	551	143	144	433

Table D.5: Turkish Network Data: Driving time matrix (A)

City name	*Node id*	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
ADANA	1	345	779	324	450	536	459	137	485	524	600	127	413	46	626	600	671	460	222	767	250
ADIYAMAN	2	137	959	189	365	353	651	100	475	453	445	213	633	265	805	819	488	592	291	946	381
AFYON	3	727	455	640	625	751	96	519	577	672	982	509	113	377	301	218	885	334	347	442	286
AĞRI	4	295	1091	331	247	123	857	504	364	256	289	635	919	687	937	1087	145	658	541	1078	627
AMASYA	5	468	601	366	243	369	378	406	216	290	761	470	503	426	447	609	503	168	232	588	205
ANKARA	6	607	455	505	455	581	155	447	407	501	911	454	281	322	302	386	714	163	212	443	123
ANTALYA	7	717	613	696	706	832	283	509	741	780	972	499	87	326	477	297	965	526	412	618	381
ARTVIN	8	367	1020	363	271	151	809	575	247	222	514	681	934	733	867	1039	140	576	565	1007	635
AYDIN	9	941	439	854	855	981	322	733	809	903	1196	723	195	555	454	84	1115	565	561	450	517
BALIKESIR	10	941	273	851	799	925	198	733	728	836	1196	723	264	591	260	115	1058	447	558	284	469
BILECIK	11	808	318	706	658	784	53	649	587	695	1112	639	234	507	165	278	917	306	413	305	324
BINGÖL	12	96	1025	95	181	120	747	305	363	255	339	413	766	465	871	953	255	592	387	1012	477
BITLIS	13	139	1153	223	309	233	876	348	474	366	227	479	895	531	1000	1081	288	721	516	1141	605
BOLU	14	734	328	632	516	642	192	575	445	553	1033	579	373	447	175	397	775	164	339	315	250
BURDUR	15	793	537	705	707	833	201	584	688	781	1047	575	34	407	396	249	966	445	413	537	382
BURSA	16	854	280	752	698	824	99	695	627	735	1158	685	276	553	162	215	957	346	459	291	370
ÇANAKKALE	17	1035	145	933	879	1005	280	866	808	916	1329	857	397	724	213	217	1138	527	640	156	551
ÇANKIRI	18	615	485	513	409	535	243	467	349	455	926	511	368	379	331	473	668	76	232	472	143
ÇORUM	19	511	563	409	305	431	317	420	245	351	822	484	442	379	409	547	564	130	185	550	144
DENIZLI	20	857	465	770	771	897	238	649	725	819	1112	639	111	474	426	149	1031	481	477	476	433
DIYARBAKIR	21	0	1062	102	271	216	755	209	459	351	314	339	759	391	909	945	351	636	395	1049	484
EDIRNE	22	1062	0	960	844	970	368	903	773	881	1361	907	537	775	153	356	1103	492	667	41	578
ELAZIĞ	23	102	960	0	177	212	653	230	372	264	416	318	671	370	807	858	347	534	293	947	382
ERZINCAN	24	271	844	177	0	126	610	407	195	87	517	495	673	496	691	841	259	411	294	831	381
ERZURUM	25	216	970	212	126	0	736	425	243	135	412	530	799	582	817	967	135	537	420	957	507
ESKIŞEHIR	26	755	368	653	610	736	0	595	562	657	1059	586	201	453	215	275	869	319	360	355	271
GAZIANTEP	27	209	903	230	407	425	595	0	482	494	463	131	550	183	749	737	560	543	235	890	325
GIRESUN	28	459	773	372	195	243	562	482	0	108	653	546	687	531	620	793	377	329	329	761	388
GÜMÜŞHANE	29	351	881	264	87	135	657	494	108	0	545	582	747	570	728	887	269	437	368	869	455
HAKKARI	30	314	1361	416	517	412	1059	463	653	545	0	594	1013	646	1208	1200	378	929	699	1349	788
HATAY	31	339	907	318	495	530	586	131	546	582	594	0	541	173	753	727	665	587	299	894	377
ISPARTA	32	759	537	671	673	799	201	550	687	747	1013	541	0	391	395	255	932	444	379	536	348
IÇEL	33	391	775	370	496	582	453	183	531	570	646	173	391	0	621	595	717	455	217	762	245
ISTANBUL	34	909	153	807	691	817	215	749	620	728	1208	753	395	621	0	374	950	339	514	141	425
IZMIR	35	945	356	858	841	967	275	737	793	887	1200	727	255	595	374	0	1100	549	565	367	501
KARS	36	351	1103	347	259	135	869	560	377	269	378	665	932	717	950	1100	0	671	553	1091	640
KASTAMONU	37	636	492	534	411	537	319	543	329	437	929	587	444	455	339	549	671	0	308	479	219
KAYSERI	38	395	667	293	294	420	360	235	329	368	699	299	379	217	514	565	553	308	0	655	89
KIRKLARELI	39	1049	41	947	831	957	355	890	761	869	1349	894	536	762	141	367	1091	479	655	0	565
KIRŞEHIR	40	484	578	382	381	507	271	325	388	455	788	377	348	245	425	501	640	219	89	565	0

Table D.6: Turkish Network Data: Driving time matrix (B)

City name	*Node id*	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
ADANA	1	552	237	449	259	588	123	353	581	492	191	137	479	617	527	486	467	581	286	714	333
ADIYAMAN	2	731	457	668	123	807	109	197	800	309	345	356	485	569	707	501	259	596	276	893	348
AFYON	3	227	149	67	575	206	505	735	247	808	293	305	548	718	203	447	849	456	464	389	423
AĞRI	4	863	743	909	397	1075	541	345	1114	163	595	626	393	365	839	495	221	603	411	1025	449
AMASYA	5	373	373	430	313	597	353	531	637	497	242	294	187	357	349	87	593	172	148	535	76
ANKARA	6	228	172	207	439	374	394	664	415	673	183	233	377	547	203	276	731	285	293	390	252
ANTALYA	7	403	215	243	631	285	495	725	209	864	358	362	730	873	379	636	839	648	542	565	584
ARTVIN	8	793	768	861	428	1027	577	430	1068	319	619	651	276	106	768	377	439	486	435	955	422
AYDIN	9	380	361	275	789	104	719	949	66	1022	507	519	779	949	401	678	1063	687	691	419	654
BALIKESIR	10	186	363	147	785	91	719	949	261	1019	507	519	699	869	207	597	1063	569	647	253	605
BILECIK	11	91	279	73	641	254	595	865	361	874	385	431	558	728	66	457	933	428	502	253	461
BINGÖL	12	797	590	799	160	941	309	159	961	76	441	473	393	371	773	477	189	579	313	959	383
BITLIS	13	926	719	928	289	1069	385	189	1065	55	570	601	503	475	901	605	65	707	441	1088	512
BOLU	14	101	297	212	567	373	521	791	500	770	311	357	416	586	76	315	859	286	421	263	349
BURDUR	15	322	210	161	640	237	571	801	161	873	359	371	659	829	297	557	915	567	543	484	533
BURSA	16	88	325	115	687	191	641	911	361	920	431	477	598	768	106	497	979	468	548	250	507
ÇANAKKALE	17	266	496	281	867	223	822	1083	365	1101	611	653	779	949	287	677	1159	649	729	125	687
ÇANKIRI	18	257	259	295	459	461	414	678	502	663	203	258	320	490	233	219	739	198	295	419	223
ÇORUM	19	335	312	369	355	535	367	574	576	559	195	247	216	386	311	115	635	199	191	497	119
DENIZLI	20	352	277	191	705	137	635	865	97	938	423	435	695	865	327	594	979	603	607	445	570
DIYARBAKIR	21	835	583	794	167	933	246	63	926	172	449	480	486	467	810	545	125	640	320	997	392
EDIRNE	22	227	593	388	895	363	849	1119	505	1098	639	685	744	914	252	643	1187	614	749	93	677
ELAZIĞ	23	733	495	705	65	846	214	165	866	168	347	378	392	380	708	443	227	538	218	895	290
ERZINCAN	24	617	497	662	242	829	391	334	867	254	348	379	215	203	592	296	370	398	164	779	203
ERZURUM	25	743	623	788	277	955	426	279	993	177	474	505	273	251	718	374	297	483	290	905	329
ESKIŞEHIR	26	141	225	52	587	263	542	812	335	821	331	378	533	703	116	431	879	441	449	303	407
GAZIANTEP	27	675	374	585	165	725	53	217	717	381	289	273	476	610	651	483	331	578	283	837	330
GIRESUN	28	546	531	614	363	781	429	523	821	419	383	414	29	141	521	131	539	239	199	708	175
GÜMÜŞHANE	29	654	571	709	329	875	468	415	916	311	422	453	137	116	629	239	431	347	238	816	249
HAKKARI	30	1134	837	1049	481	1188	517	262	1181	263	753	737	683	615	1109	784	189	893	634	1296	706
HATAY	31	679	365	576	253	715	117	347	708	486	319	264	540	679	655	547	461	642	347	841	394
ISPARTA	32	321	176	161	606	243	537	767	195	839	325	337	658	828	297	557	881	566	509	483	533
IÇEL	33	547	232	443	305	583	169	399	535	538	187	132	525	663	523	493	513	577	332	709	379
ISTANBUL	34	74	440	235	741	350	696	966	520	945	485	532	591	761	99	489	1033	461	595	88	523
IZMIR	35	300	367	223	793	24	723	953	150	1026	511	523	763	933	321	662	1067	671	679	337	638
KARS	36	876	756	921	413	1088	561	415	1127	233	607	639	406	241	851	507	353	616	423	1038	462
KASTAMONU	37	265	335	371	481	537	490	699	578	665	279	334	300	470	240	199	761	122	316	427	244
KAYSERI	38	440	203	412	227	553	182	452	573	461	54	85	323	461	415	299	519	384	130	602	177
KIRKLARELI	39	215	581	375	882	374	837	1107	516	1085	626	673	731	901	239	630	1174	601	736	81	664
KIRŞEHIR	40	351	172	323	317	489	271	541	530	550	61	115	359	529	326	259	609	341	217	513	213

Table D.7: Turkish Network Data: Driving time matrix (C)

City name	*Node id*	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
ADANA	1	568	414	228	459	597	315	503	177	539	193	317	409	474	513	691	709	595	467	161	57	481
ADIYAMAN	2	519	279	73	679	388	408	682	395	436	413	456	201	319	692	507	501	775	647	139	163	661
AFYON	3	669	712	610	77	957	315	325	243	687	224	221	791	856	343	897	945	224	298	543	439	249
AĞRI	4	323	283	413	947	155	560	814	645	204	743	651	246	284	779	207	95	907	733	543	584	793
AMASYA	5	307	330	480	468	645	131	324	279	305	411	173	535	656	289	519	563	417	243	445	419	303
ANKARA	6	498	541	539	245	821	144	179	150	517	246	50	673	785	189	726	775	271	143	487	383	157
ANTALYA	7	824	786	600	196	969	456	501	308	795	250	410	781	846	535	981	1026	400	490	533	429	425
ARTVIN	8	155	307	484	899	379	560	743	669	229	767	603	402	503	697	78	229	836	651	615	630	722
AYDIN	9	900	942	824	185	1171	546	523	457	919	432	452	1005	1070	575	1128	1175	338	529	757	653	447
BALIKESIR	10	819	885	824	150	1167	497	329	457	861	438	403	1005	1070	388	1047	1119	144	372	757	653	253
BILECIK	11	679	745	740	166	1023	353	188	349	720	354	259	875	986	247	907	978	84	231	673	569	112
BINGÖL	12	322	94	213	812	225	462	748	491	203	589	552	131	253	713	275	316	841	667	344	362	727
BITLIS	13	433	223	257	941	112	591	877	620	314	678	681	90	128	842	327	225	969	795	387	428	855
BOLU	14	537	603	666	305	917	271	106	275	578	371	177	801	912	116	765	836	144	89	613	508	30
BURDUR	15	779	793	675	115	1022	425	419	309	796	281	331	857	921	454	982	1027	319	409	609	504	343
BURSA	16	719	785	786	207	1069	399	228	395	760	400	305	921	1032	287	947	1018	46	271	719	615	152
ÇANAKKALE	17	899	965	957	283	1249	579	409	575	941	571	485	1101	1203	468	1127	1199	224	452	891	786	333
ÇANKIRI	18	441	495	559	333	810	164	208	207	471	333	70	681	803	189	669	729	301	130	507	441	187
ÇORUM	19	337	391	511	407	706	69	286	217	367	358	111	577	699	251	565	625	379	205	459	433	265
DENIZLI	20	816	858	740	101	1087	462	449	373	835	348	368	921	986	491	1044	1091	335	445	673	569	373
DIYARBAKIR	21	418	184	117	805	251	469	785	499	299	539	559	67	188	757	371	364	878	711	248	289	764
EDIRNE	22	865	931	994	423	1245	599	374	603	906	669	505	1129	1240	433	1093	1164	271	417	941	836	298
ELAZIĞ	23	331	90	219	717	317	367	683	397	263	495	457	169	290	655	367	408	776	609	269	267	662
ERZINCAN	24	154	87	388	700	401	313	567	398	103	496	405	312	433	533	275	320	660	486	446	444	546
ERZURUM	25	202	162	333	826	277	439	693	524	83	622	531	251	361	659	155	196	786	612	464	479	672
ESKIŞEHIR	26	653	697	687	145	969	299	238	295	672	301	205	821	933	297	881	930	137	281	620	515	162
GAZIANTEP	27	561	320	91	596	460	352	626	313	493	330	400	273	337	636	579	573	719	591	42	80	605
GIRESUN	28	91	282	531	652	519	313	497	433	160	531	357	495	602	451	319	437	589	404	521	495	475
GÜMÜŞHANE	29	67	174	469	747	411	387	605	472	52	570	451	387	494	559	285	329	697	512	533	531	583
HAKKARI	30	612	431	372	1059	135	783	1085	777	493	793	863	247	126	1050	436	285	1177	1003	503	543	1063
HATAY	31	629	408	222	587	591	416	630	304	581	321	444	403	468	640	685	703	723	595	98	85	609
ISPARTA	32	779	759	641	114	988	407	419	275	762	247	331	823	887	453	948	993	318	408	575	470	343
IÇEL	33	614	460	274	454	643	311	498	172	585	157	312	455	520	508	737	755	591	463	207	103	477
ISTANBUL	34	711	777	841	327	1092	446	221	449	753	515	352	975	1087	280	939	1011	117	264	787	683	145
IZMIR	35	884	927	828	141	1175	530	443	461	903	442	436	1009	1074	502	1112	1161	258	486	761	657	367
KARS	36	290	295	469	959	243	573	827	657	217	755	664	378	416	792	62	93	919	745	599	615	805
KASTAMONU	37	421	498	635	409	813	197	181	283	473	409	146	703	824	121	649	731	308	75	583	517	194
KAYSERI	38	412	381	327	425	609	117	391	104	383	202	165	461	573	401	569	614	483	355	275	249	369
KIRKLARELI	39	852	918	981	434	1233	587	361	590	893	656	493	1116	1227	421	1080	1151	258	405	928	823	285
KIRŞEHIR	40	479	467	416	361	699	75	301	73	470	214	75	551	662	311	656	701	394	266	364	307	280

Table D.8: Turkish Network Data: Driving time matrix (D)

City name	*Node id*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
KOCAELI	41	552	731	227	863	373	228	403	793	380	186	91	797	926	101	322	88	266	257	335	352
KONYA	42	237	457	149	743	373	172	215	768	361	363	279	590	719	297	210	325	496	259	312	277
KÜTAHYA	43	449	668	67	909	430	207	243	861	275	147	73	799	928	212	161	115	281	295	369	191
MALATYA	44	259	123	575	397	313	439	631	428	789	785	641	160	289	567	640	687	867	459	355	705
MANISA	45	588	807	206	1075	597	374	285	1027	104	91	254	941	1069	373	237	191	223	461	535	137
KAHRAMANMARAŞ	46	123	109	505	541	353	394	495	577	719	719	595	309	385	521	571	641	822	414	367	635
MARDIN	47	353	197	735	345	531	664	725	430	949	949	865	159	189	791	801	911	1083	678	574	865
MUĞLA	48	581	800	247	1114	637	415	209	1068	66	261	361	961	1065	500	161	361	365	502	576	97
MUŞ	49	492	309	808	163	497	673	864	319	1022	1019	874	76	55	770	873	920	1101	663	559	938
NEVŞEHIR	50	191	345	293	595	242	183	358	619	507	507	385	441	570	311	359	431	611	203	195	423
NIĞDE	51	137	356	305	626	294	233	362	651	519	519	431	473	601	357	371	477	653	258	247	435
ORDU	52	479	485	548	393	187	377	730	276	779	699	558	393	503	416	659	598	779	320	216	695
RIZE	53	617	569	718	365	357	547	873	106	949	869	728	371	475	586	829	768	949	490	386	865
SAKARYA	54	527	707	203	839	349	203	379	768	401	207	66	773	901	76	297	106	287	233	311	327
SAMSUN	55	486	501	447	495	87	276	636	377	678	597	457	477	605	315	557	497	677	219	115	594
SIIRT	56	467	259	849	221	593	731	839	439	1063	1063	933	189	65	859	915	979	1159	739	635	979
SINOP	57	581	596	456	603	172	285	648	486	687	569	428	579	707	286	567	468	649	198	199	603
SIVAS	58	286	276	464	411	148	293	542	435	691	647	502	313	441	421	543	548	729	295	191	607
TEKIRDAĞ	59	714	893	389	1025	535	390	565	955	419	253	253	959	1088	263	484	250	125	419	497	445
TOKAT	60	333	348	423	449	76	252	584	422	654	605	461	383	512	349	533	507	687	223	119	570
TRABZON	61	568	519	669	323	307	498	824	155	900	819	679	322	433	537	779	719	899	441	337	816
TUNCELI	62	414	279	712	283	330	541	786	307	942	885	745	94	223	603	793	785	965	495	391	858
ŞANLIURFA	63	228	73	610	413	480	539	600	484	824	824	740	213	257	666	675	786	957	559	511	740
UŞAK	64	459	679	77	947	468	245	196	899	185	150	166	812	941	305	115	207	283	333	407	101
VAN	65	597	388	957	155	645	821	969	379	1171	1167	1023	225	112	917	1022	1069	1249	810	706	1087
YOZGAT	66	315	408	315	560	131	144	456	560	546	497	353	462	591	271	425	399	579	164	69	462
ZONGULDAK	67	503	682	325	814	324	179	501	743	523	329	188	748	877	106	419	228	409	208	286	449
AKSARAY	68	177	395	243	645	279	150	308	669	457	457	349	491	620	275	309	395	575	207	217	373
BAYBURT	69	539	436	687	204	305	517	795	229	919	861	720	203	314	578	796	760	941	471	367	835
KARAMAN	70	193	413	224	743	411	246	250	767	432	438	354	589	678	371	281	400	571	333	358	348
KIRIKKALE	71	317	456	221	651	173	50	410	603	452	403	259	552	681	177	331	305	485	70	111	368
BATMAN	72	409	201	791	246	535	673	781	402	1005	1005	875	131	90	801	857	921	1101	681	577	921
ŞIRNAK	73	474	319	856	284	656	785	846	503	1070	1070	986	253	128	912	921	1032	1203	803	699	986
BARTIN	74	513	692	343	779	289	189	535	697	575	388	247	713	842	116	454	287	468	189	251	491
ARDAHAN	75	691	507	897	207	519	726	981	78	1128	1047	907	275	327	765	982	947	1127	669	565	1044
IĞDIR	76	709	501	945	95	563	775	1026	229	1175	1119	978	316	225	836	1027	1018	1199	729	625	1091
YALOVA	77	595	775	224	907	417	271	400	836	338	144	84	841	969	144	319	46	224	301	379	335
KARABÜK	78	467	647	298	733	243	143	490	651	529	372	231	667	795	89	409	271	452	130	205	445
KILIS	79	161	139	543	543	445	487	533	615	757	757	673	344	387	613	609	719	891	507	459	673
OSMANIYE	80	57	163	439	584	419	383	429	630	653	653	569	362	428	508	504	615	786	441	433	569
DÜZCE	81	481	661	249	793	303	157	425	722	447	253	112	727	855	30	343	152	333	187	265	373

Table D.9: Turkish Network Data: Driving time matrix (E)

City name	*Node id*	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
KOCAELI	41	835	227	733	617	743	141	675	546	654	1134	679	321	547	74	300	876	265	440	215	351
KONYA	42	583	593	495	497	623	225	374	531	571	837	365	176	232	440	367	756	335	203	581	172
KÜTAHYA	43	794	388	705	662	788	52	585	614	709	1049	576	161	443	235	223	921	371	412	375	323
MALATYA	44	167	895	65	242	277	587	165	363	329	481	253	606	305	741	793	413	481	227	882	317
MANISA	45	933	363	846	829	955	263	725	781	875	1188	715	243	583	350	24	1088	537	553	374	489
KAHRAMANMARAŞ	46	246	849	214	391	426	542	53	429	468	517	117	537	169	696	723	561	490	182	837	271
MARDIN	47	63	1119	165	334	279	812	217	523	415	262	347	767	399	966	953	415	699	452	1107	541
MUĞLA	48	926	505	866	867	993	335	717	821	916	1181	708	195	535	520	150	1127	578	573	516	530
MUŞ	49	172	1098	168	254	177	821	381	419	311	263	486	839	538	945	1026	233	665	461	1085	550
NEVŞEHIR	50	449	639	347	348	474	331	289	383	422	753	319	325	187	485	511	607	279	54	626	61
NIĞDE	51	480	685	378	379	505	378	273	414	453	737	264	337	132	532	523	639	334	85	673	115
ORDU	52	486	744	392	215	273	533	476	29	137	683	540	658	525	591	763	406	300	323	731	359
RIZE	53	467	914	380	203	251	703	610	141	116	615	679	828	663	761	933	241	470	461	901	529
SAKARYA	54	810	252	708	592	718	116	651	521	629	1109	655	297	523	99	321	851	240	415	239	326
SAMSUN	55	545	643	443	296	374	431	483	131	239	784	547	557	493	489	662	507	199	299	630	259
SIIRT	56	125	1187	227	370	297	879	331	539	431	189	461	881	513	1033	1067	353	761	519	1174	609
SINOP	57	640	614	538	398	483	441	578	239	347	893	642	566	577	461	671	616	122	384	601	341
SIVAS	58	320	749	218	164	290	449	283	199	238	634	347	509	332	595	679	423	316	130	736	217
TEKIRDAĞ	59	997	93	895	779	905	303	837	708	816	1296	841	483	709	88	337	1038	427	602	81	513
TOKAT	60	392	677	290	203	329	407	330	175	249	706	394	533	379	523	638	462	244	177	664	213
TRABZON	61	418	865	331	154	202	653	561	91	67	612	629	779	614	711	884	290	421	412	852	479
TUNCELI	62	184	931	90	87	162	697	320	282	174	431	408	759	460	777	927	295	498	381	918	467
ŞANLIURFA	63	117	994	219	388	333	687	91	531	469	372	222	641	274	841	828	469	635	327	981	416
UŞAK	64	805	423	717	700	826	145	596	652	747	1059	587	114	454	327	141	959	409	425	434	361
VAN	65	251	1245	317	401	277	969	460	519	411	135	591	988	643	1092	1175	243	813	609	1233	699
YOZGAT	66	469	599	367	313	439	299	352	313	387	783	416	407	311	446	530	573	197	117	587	75
ZONGULDAK	67	785	374	683	567	693	238	626	497	605	1085	630	419	498	221	443	827	181	391	361	301
AKSARAY	68	499	603	397	398	524	295	313	433	472	777	304	275	172	449	461	657	283	104	590	73
BAYBURT	69	299	906	263	103	83	672	493	160	52	493	581	762	585	753	903	217	473	383	893	470
KARAMAN	70	539	669	495	496	622	301	330	531	570	793	321	247	157	515	442	755	409	202	656	214
KIRIKKALE	71	559	505	457	405	531	205	400	357	451	863	444	331	312	352	436	664	146	165	493	75
BATMAN	72	67	1129	169	312	251	821	273	495	387	247	403	823	455	975	1009	378	703	461	1116	551
ŞIRNAK	73	188	1240	290	433	361	933	337	602	494	126	468	887	520	1087	1074	416	824	573	1227	662
BARTIN	74	757	433	655	533	659	297	636	451	559	1050	640	453	508	280	502	792	121	401	421	311
ARDAHAN	75	371	1093	367	275	155	881	579	319	285	436	685	948	737	939	1112	62	649	569	1080	656
IĞDIR	76	364	1164	408	320	196	930	573	437	329	285	703	993	755	1011	1161	93	731	614	1151	701
YALOVA	77	878	271	776	660	786	137	719	589	697	1177	723	318	591	117	258	919	308	483	258	394
KARABÜK	78	711	417	609	486	612	281	591	404	512	1003	595	408	463	264	486	745	75	355	405	266
KILIS	79	248	941	269	446	464	620	42	521	533	503	98	575	207	787	761	599	583	275	928	364
OSMANIYE	80	289	836	267	444	479	515	80	495	531	543	85	470	103	683	657	615	517	249	823	307
DÜZCE	81	764	298	662	546	672	162	605	475	583	1063	609	343	477	145	367	805	194	369	285	280

Table D.10: Turkish Network Data: Driving time matrix (F)

City name	*Node id*	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
KOCAELI	41	0	366	161	667	276	622	892	446	871	411	458	517	687	25	415	959	387	521	162	449
KONYA	42	366	0	215	430	355	361	591	371	663	149	161	521	664	341	427	705	457	333	528	375
KÜTAHYA	43	161	215	0	639	211	572	802	288	873	360	372	585	755	136	483	916	493	501	323	459
MALATYA	44	667	430	639	0	781	149	231	801	233	281	313	374	445	643	390	292	485	165	829	237
MANISA	45	276	355	211	781	0	711	941	170	1014	499	511	751	921	297	650	1055	659	667	343	626
KAHRAMANMARAŞ	46	622	361	572	149	711	0	270	704	382	236	260	423	561	597	430	368	525	230	784	277
MARDIN	47	892	591	802	231	941	270	0	934	235	506	490	549	531	867	609	153	703	383	1054	455
MUĞLA	48	446	371	288	801	170	704	934	0	1034	519	531	792	962	424	691	1048	700	703	485	667
MUŞ	49	871	663	873	233	1014	382	235	1034	0	515	546	448	420	846	549	120	652	386	1033	457
NEVŞEHIR	50	411	149	360	281	499	236	506	519	515	0	55	372	515	387	309	573	394	184	573	226
NIĞDE	51	458	161	372	313	511	260	490	531	546	55	0	408	547	433	361	604	446	215	620	262
ORDU	52	517	521	585	374	751	423	549	792	448	372	408	0	170	492	101	568	210	209	679	146
RIZE	53	687	664	755	445	921	561	531	962	420	515	547	170	0	662	271	540	380	331	849	316
SAKARYA	54	25	341	136	643	297	597	867	424	846	387	433	492	662	0	391	935	362	497	187	425
SAMSUN	55	415	427	483	390	650	430	609	691	549	309	361	101	271	391	0	666	109	225	577	153
SIIRT	56	959	705	916	292	1055	368	153	1048	120	573	604	568	540	935	666	0	765	445	1121	517
SINOP	57	387	457	493	485	659	525	703	700	652	394	446	210	380	362	109	765	0	320	549	248
SIVAS	58	521	333	501	165	667	230	383	703	386	184	215	209	331	497	225	445	320	0	683	72
TEKIRDAĞ	59	162	528	323	829	343	784	1054	485	1033	573	620	679	849	187	577	1121	549	683	0	611
TOKAT	60	449	375	459	237	626	277	455	667	457	226	262	146	316	425	153	517	248	72	611	0
TRABZON	61	637	615	705	396	872	512	481	913	377	466	497	121	49	613	222	497	331	282	799	267
TUNCELI	62	703	583	749	155	915	304	247	954	167	435	466	302	290	679	383	283	485	251	865	289
ŞANLIURFA	63	767	465	677	179	816	145	125	809	289	381	365	541	585	742	557	239	652	332	929	404
UŞAK	64	253	226	93	652	129	583	813	198	885	371	383	623	793	229	521	927	531	539	403	497
VAN	65	1018	812	1021	382	1163	497	301	1177	149	663	695	548	480	993	649	177	758	535	1180	604
YOZGAT	66	372	247	351	314	518	299	533	559	535	127	179	284	454	347	184	594	269	149	534	138
ZONGULDAK	67	147	348	258	618	419	573	843	546	821	362	409	467	637	122	366	910	303	472	309	400
AKSARAY	68	375	99	310	331	449	286	530	469	565	50	83	422	565	351	332	623	405	234	537	276
BAYBURT	69	679	586	724	329	891	477	363	931	259	437	469	189	168	654	291	379	399	253	841	265
KARAMAN	70	441	79	291	429	430	317	547	441	663	171	117	525	663	417	473	661	531	332	603	379
KIRIKKALE	71	278	201	257	392	424	347	617	465	625	136	191	327	497	253	226	684	268	243	440	202
BATMAN	72	901	647	858	234	997	310	99	990	145	515	546	524	503	877	608	58	707	387	1063	459
ŞIRNAK	73	1013	711	923	355	1062	391	136	1055	183	627	611	631	603	988	729	63	828	508	1175	580
BARTIN	74	206	358	317	602	478	583	821	587	787	372	419	421	591	181	320	882	243	437	368	365
ARDAHAN	75	865	772	933	432	1100	581	434	1141	272	623	655	349	179	841	450	392	559	439	1027	478
IĞDIR	76	937	817	982	473	1149	610	414	1187	259	668	699	467	330	912	568	289	677	484	1099	523
YALOVA	77	43	363	157	711	234	665	935	404	914	455	501	560	730	68	459	1003	430	565	205	493
KARABÜK	78	190	313	301	555	462	537	774	542	740	327	373	375	545	165	273	835	197	391	352	319
KILIS	79	713	399	610	204	749	93	256	742	420	329	298	515	649	689	523	370	617	323	875	369
OSMANIYE	80	609	294	505	202	645	67	297	637	435	248	193	489	628	584	497	411	591	297	771	343
DÜZCE	81	71	327	182	597	343	551	821	470	800	341	387	446	616	46	345	889	316	451	233	379

Table D.11: Turkish Network Data: Driving time matrix (G)

City name	*Node id*	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
KOCAELI	41	637	703	767	253	1018	372	147	375	679	441	278	901	1013	206	865	937	43	190	713	609	71
KONYA	42	615	583	465	226	812	247	348	99	586	79	201	647	711	358	772	817	363	313	399	294	327
KÜTAHYA	43	705	749	677	93	1021	351	258	310	724	291	257	858	923	317	933	982	157	301	610	505	182
MALATYA	44	396	155	179	652	382	314	618	331	329	429	392	234	355	602	432	473	711	555	204	202	597
MANISA	45	872	915	816	129	1163	518	419	449	891	430	424	997	1062	478	1100	1149	234	462	749	645	343
KAHRAMANMARAŞ	46	512	304	145	583	497	299	573	286	477	317	347	310	391	583	581	610	665	537	93	67	551
MARDIN	47	481	247	125	813	301	533	843	530	363	547	617	99	136	821	434	414	935	774	256	297	821
MUĞLA	48	913	954	809	198	1177	559	546	469	931	441	465	990	1055	587	1141	1187	404	542	742	637	470
MUŞ	49	377	167	289	885	149	535	821	565	259	663	625	145	183	787	272	259	914	740	420	435	800
NEVŞEHIR	50	466	435	381	371	663	127	362	50	437	171	136	515	627	372	623	668	455	327	329	248	341
NIĞDE	51	497	466	365	383	695	179	409	83	469	117	191	546	611	419	655	699	501	373	298	193	387
ORDU	52	121	302	541	623	548	284	467	422	189	525	327	524	631	421	349	467	560	375	515	489	446
RIZE	53	49	290	585	793	480	454	637	565	168	663	497	503	603	591	179	330	730	545	649	628	616
SAKARYA	54	613	679	742	229	993	347	122	351	654	417	253	877	988	181	841	912	68	165	689	584	46
SAMSUN	55	222	383	557	521	649	184	366	332	291	473	226	608	729	320	450	568	459	273	523	497	345
SIIRT	56	497	283	239	927	177	594	910	623	379	661	684	58	63	882	392	289	1003	835	370	411	889
SINOP	57	331	485	652	531	758	269	303	405	399	531	268	707	828	243	559	677	430	197	617	591	316
SIVAS	58	282	251	332	539	535	149	472	234	253	332	243	387	508	437	439	484	565	391	323	297	451
TEKIRDAĞ	59	799	865	929	403	1180	534	309	537	841	603	440	1063	1175	368	1027	1099	205	352	875	771	233
TOKAT	60	267	289	404	497	604	138	400	276	265	379	202	459	580	365	478	523	493	319	369	343	379
TRABZON	61	0	241	535	743	477	405	588	516	119	614	448	453	561	542	228	379	681	495	600	579	567
TUNCELI	62	241	0	301	787	316	400	654	485	173	583	491	225	347	619	311	356	747	573	359	357	633
ŞANLIURFA	63	535	301	0	687	369	443	717	405	417	421	491	181	246	727	488	481	810	682	131	171	696
UŞAK	64	743	787	687	0	1034	389	351	321	762	301	295	869	933	410	971	1020	250	373	621	516	275
VAN	65	477	316	369	1034	0	684	969	713	359	790	774	202	240	934	301	150	1061	887	499	540	947
YOZGAT	66	405	400	443	389	684	0	323	148	403	289	94	536	657	318	589	633	415	271	391	365	301
ZONGULDAK	67	588	654	717	351	969	323	0	326	629	422	229	852	963	59	816	887	190	118	664	559	76
AKSARAY	68	516	485	405	321	713	148	326	0	487	141	140	565	651	336	673	718	419	291	338	233	305
BAYBURT	69	119	173	417	762	359	403	629	487	0	585	467	335	442	595	233	277	722	548	533	531	608
KARAMAN	70	614	583	421	301	790	289	422	141	585	0	275	603	667	432	771	816	438	387	355	250	401
KIRIKKALE	71	448	491	491	295	774	94	229	140	467	275	0	626	737	239	676	725	321	193	439	373	207
BATMAN	72	453	225	181	869	202	536	852	565	335	603	626	0	121	824	406	315	945	777	312	353	831
ŞIRNAK	73	561	347	246	933	240	657	963	651	442	667	737	121	0	945	455	353	1056	899	377	417	942
BARTIN	74	542	619	727	410	934	318	59	336	595	432	239	824	945	0	770	853	249	59	674	569	135
ARDAHAN	75	228	311	488	971	301	589	816	673	233	771	676	406	455	770	0	151	909	723	619	634	795
IĞDIR	76	379	356	481	1020	150	633	887	718	277	816	725	315	353	853	151	0	980	806	612	653	866
YALOVA	77	681	747	810	250	1061	415	190	419	722	438	321	945	1056	249	909	980	0	233	757	652	114
KARABÜK	78	495	573	682	373	887	271	118	291	548	387	193	777	899	59	723	806	233	0	629	524	119
KILIS	79	600	359	131	621	499	391	664	338	533	355	439	312	377	674	619	612	757	629	0	105	643
OSMANIYE	80	579	357	171	516	540	365	559	233	531	250	373	353	417	569	634	653	652	524	105	0	538
DÜZCE	81	567	633	696	275	947	301	76	305	608	401	207	831	942	135	795	866	114	119	643	538	0

Table D.12: Turkish Network Data: Driving time matrix (H)

City name	*Node id*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ADANA	1	0	17493	22782	14827	10242	112387	48225	5382	26661	30183	5449	7115	10899	7590	7201	59593	13039	7581	16743	23836
ADIYAMAN	2	17174	0	7544	4910	3391	37216	15969	1782	8828	9995	1804	2356	3609	2513	2385	19733	4318	2510	5544	7893
AFYON	3	22429	7565	0	6412	4429	48604	20856	2328	11530	13053	2357	3077	4714	3282	3114	25772	5639	3279	7241	10308
AĞRI	4	14536	4903	6385	0	2871	31499	13516	1508	7472	8459	1527	1994	3055	2127	2018	16702	3654	2125	4693	6681
AMASYA	5	10016	3378	4400	2864	0	21706	9314	1039	5149	5829	1052	1374	2105	1466	1391	11509	2518	1464	3234	4604
ANKARA	6	116190	39190	51038	33217	22945	0	108040	12058	59729	67619	12208	15941	24418	17003	16133	133508	29211	16985	37509	53401
ANTALYA	7	48130	16234	21142	13760	9505	104299	0	4995	24742	28010	5057	6603	10115	7043	6683	55304	12100	7036	15538	22121
ARTVIN	8	5250	1771	2306	1501	1037	11377	4882	0	2699	3055	552	720	1103	768	729	6033	1320	767	1695	2413
AYDIN	9	26302	8872	11554	7520	5194	56998	24458	2730	0	15307	2764	3609	5528	3849	3652	30223	6613	3845	8491	12089
BALIKESIR	10	29833	10062	13105	8529	5891	64649	27740	3096	15336	0	3135	4093	6270	4366	4142	34279	7500	4361	9631	13711
BILECIK	11	5316	1793	2335	1520	1050	11520	4943	552	2733	3094	0	729	1117	778	738	6108	1336	777	1716	2443
BINGÖL	12	6947	2343	3052	1986	1372	15055	6460	721	3571	4043	730	0	1460	1017	965	7983	1747	1016	2243	3193
BITLIS	13	10663	3597	4684	3048	2106	23107	9915	1107	5482	6206	1120	1463	0	1560	1481	12252	2681	1559	3442	4901
BOLU	14	7412	2500	3256	2119	1464	16062	6892	769	3810	4314	779	1017	1558	0	1029	8517	1863	1084	2393	3407
BURDUR	15	7031	2372	3089	2010	1389	15237	6538	730	3615	4092	739	965	1478	1029	0	8079	1768	1028	2270	3232
BURSA	16	59843	20184	26287	17108	11818	129681	55645	6210	30763	34827	6288	8210	12576	8757	8309	0	15045	8748	19319	27504
ÇANAKKALE	17	12771	4307	5610	3651	2522	27674	11875	1325	6565	7432	1342	1752	2684	1869	1773	14674	0	1867	4123	5869
ÇANKIRI	18	7404	2497	3252	2117	1462	16045	6885	768	3806	4309	778	1016	1556	1084	1028	8508	1861	0	2390	3403
ÇORUM	19	16431	5542	7217	4697	3245	35606	15278	1705	8447	9562	1726	2254	3453	2404	2281	18880	4131	2402	0	7552
DENIZLI	20	23480	7920	10314	6713	4637	50883	21834	2437	12071	13665	2467	3221	4935	3436	3260	26980	5903	3432	7580	0
DIYARBAKIR	21	37933	12794	16663	10845	7491	82201	35272	3937	19500	22076	3986	5204	7972	5551	5267	43587	9537	5545	12246	17434
EDIRNE	22	11047	3726	4853	3158	2182	23940	10273	1146	5679	6429	1161	1516	2322	1617	1534	12694	2777	1615	3566	5077
ELAZIĞ	23	15669	5285	6883	4480	3094	33955	14570	1626	8055	9119	1646	2150	3293	2293	2176	18004	3939	2290	5058	7202
ERZINCAN	24	8683	2929	3814	2482	1715	18816	8074	901	4464	5053	912	1191	1825	1271	1206	9977	2183	1269	2803	3991
ERZURUM	25	25927	8745	11389	7412	5120	56185	24109	2691	13328	15089	2724	3557	5449	3794	3600	29792	6518	3790	8370	11916
ESKIŞEHIR	26	19460	6564	8548	5563	3843	42171	18095	2020	10004	11325	2045	2670	4090	2848	2702	22361	4892	2845	6282	8944
GAZIANTEP	27	35735	12053	15697	10216	7057	77438	33228	3708	18370	20797	3755	4903	7510	5229	4962	41061	8984	5224	11536	16424
GIRESUN	28	14399	4857	6325	4117	2844	31204	13389	1494	7402	8380	1513	1976	3026	2107	1999	16546	3620	2105	4649	6618
GÜMÜŞHANE	29	5114	1725	2246	1462	1010	11081	4755	531	2629	2976	537	702	1075	748	710	5876	1286	747	1651	2350
HAKKARI	30	6476	2184	2845	1851	1279	14033	6022	672	3329	3769	680	888	1361	948	899	7441	1628	947	2091	2976
HATAY	31	34842	11752	15305	9961	6881	75503	32398	3616	17911	20277	3661	4780	7322	5099	4838	40035	8760	5093	11248	16014
ISPARTA	32	14119	4762	6202	4036	2788	30595	13128	1465	7258	8217	1483	1937	2967	2066	1960	16223	3550	2064	4558	6489
IÇEL	33	46169	15573	20281	13199	9117	100050	42931	4791	23734	26869	4851	6334	9703	6756	6411	53051	11607	6749	14905	21220
ISTANBUL	34	320661	108156	140856	91673	63323	694878	298169	33277	164841	186616	33692	43993	67389	46926	44524	368455	80617	46874	103518	147377
IZMIR	35	96757	32635	42502	27662	19107	209674	89970	10041	49740	56310	10166	13275	20334	14159	13435	111178	24326	14144	31236	44470
KARS	36	8908	3005	3913	2547	1759	19304	8283	924	4579	5184	936	1222	1872	1304	1237	10236	2240	1302	2876	4094
KASTAMONU	37	10299	3474	4524	2944	2034	22318	9576	1069	5294	5994	1082	1413	2164	1507	1430	11834	2589	1505	3325	4733
KAYSERI	38	29385	9911	12908	8401	5803	63678	27324	3049	15106	17101	3087	4031	6175	4300	4080	33765	7388	4295	9486	13505
KIRKLARELI	39	9003	3037	3955	2574	1778	19510	8372	934	4628	5240	946	1235	1892	1318	1250	10345	2263	1316	2906	4138
KIRŞEHIR	40	6933	2339	3046	1982	1369	15025	6447	720	3564	4035	729	951	1457	1015	963	7967	1743	1014	2238	3187

Table D.13: Turkish Network Data: Volume distribution matrix (A)

City name	*Node id*	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
ADANA	1	38213	11290	15973	8885	26286	19798	36041	14689	5242	6634	35157	14405	46308	280943	94525	9114	10529	29736	9211	7101
ADIYAMAN	2	12654	3738	5289	2942	8704	6556	11934	4864	1736	2197	11642	4770	15334	93030	31301	3018	3487	9847	3050	2351
AFYON	3	16526	4882	6908	3842	11368	8562	15586	6352	2267	2869	15204	6229	20027	121499	40879	3942	4553	12860	3983	3071
AĞRI	4	10710	3164	4477	2490	7367	5549	10101	4117	1469	1859	9854	4037	12979	78741	26493	2554	2951	8334	2582	1990
AMASYA	5	7380	2180	3085	1716	5077	3824	6961	2837	1012	1281	6790	2782	8944	54259	18256	1760	2033	5743	1779	1371
ANKARA	6	85609	25293	35785	19905	58890	44354	80743	32908	11745	14863	78763	32271	103746	629407	211768	20418	23589	66620	20635	15909
ANTALYA	7	35463	10477	14823	8245	24394	18373	33447	13632	4865	6157	32627	13368	42975	260724	87722	8458	9771	27596	8548	6590
ARTVIN	8	3868	1143	1617	899	2661	2004	3649	1487	531	672	3559	1458	4688	28441	9569	923	1066	3010	932	719
AYDIN	9	19380	5726	8101	4506	13331	10041	18278	7450	2659	3365	17830	7305	23485	142482	47939	4622	5340	15081	4671	3601
BALIKESIR	10	21981	6494	9188	5111	15121	11388	20732	8449	3016	3816	20223	8286	26638	161607	54374	5243	6057	17105	5298	4085
BILECIK	11	3917	1157	1637	911	2694	2029	3694	1506	537	680	3604	1476	4747	28796	9689	934	1079	3048	944	728
BINGÖL	12	5119	1512	2140	1190	3521	2652	4828	1968	702	889	4709	1930	6203	37633	12662	1221	1410	3983	1234	951
BITLIS	13	7857	2321	3284	1827	5404	4070	7410	3020	1078	1364	7228	2962	9521	57762	19434	1874	2165	6114	1894	1460
BOLU	14	5461	1614	2283	1270	3757	2829	5151	2099	749	948	5025	2059	6618	40152	13509	1303	1505	4250	1316	1015
BURDUR	15	5181	1531	2166	1205	3564	2684	4886	1991	711	899	4766	1953	6278	38090	12815	1236	1427	4032	1249	963
BURSA	16	44093	13027	18431	10252	30331	22844	41586	16949	6049	7655	40566	16621	53434	324172	109070	10516	12149	34312	10628	8194
ÇANAKKALE	17	9409	2780	3933	2188	6473	4875	8875	3617	1291	1634	8657	3547	11403	69179	23276	2244	2593	7322	2268	1749
ÇANKIRI	18	5455	1612	2280	1268	3753	2826	5145	2097	748	947	5019	2056	6611	40108	13494	1301	1503	4245	1315	1014
ÇORUM	19	12106	3577	5060	2815	8328	6272	11418	4654	1661	2102	11138	4564	14671	89006	29947	2887	3336	9421	2918	2250
DENIZLI	20	17301	5111	7232	4023	11901	8963	16317	6650	2374	3004	15917	6522	20966	127195	42796	4126	4767	13463	4170	3215
DIYARBAKIR	21	0	8257	11683	6498	19226	14480	26360	10744	3834	4852	25714	10536	33870	205484	69136	6666	7701	21749	6737	5194
EDIRNE	22	8140	0	3402	1893	5599	4217	7677	3129	1117	1413	7489	3068	9864	59845	20135	1941	2243	6334	1962	1513
ELAZIĞ	23	11545	3411	0	2684	7942	5981	10889	4438	1584	2004	10622	4352	13991	84880	28558	2754	3181	8984	2783	2145
ERZINCAN	24	6398	1890	2674	0	4401	3315	6034	2459	878	1111	5886	2412	7753	47036	15826	1526	1763	4979	1542	1189
ERZURUM	25	19104	5644	7985	4442	0	9897	18018	7343	2621	3317	17576	7201	23151	140451	47256	4556	5264	14866	4605	3550
ESKIŞEHIR	26	14339	4236	5994	3334	9863	0	13523	5512	1967	2489	13192	5405	17376	105418	35468	3420	3951	11158	3456	2665
GAZIANTEP	27	26330	7779	11006	6122	18112	13641	0	10121	3612	4571	24224	9925	31908	193578	65131	6280	7255	20489	6346	4893
GIRESUN	28	10610	3135	4435	2467	7298	5497	10006	0	1456	1842	9761	3999	12857	78002	26244	2530	2923	8256	2557	1972
GÜMÜŞHANE	29	3768	1113	1575	876	2592	1952	3554	1448	0	654	3466	1420	4566	27701	9320	899	1038	2932	908	700
HAKKARI	30	4771	1410	1994	1109	3282	2472	4500	1834	655	0	4390	1799	5782	35080	11803	1138	1315	3713	1150	887
HATAY	31	25672	7585	10731	5969	17659	13300	24213	9868	3522	4457	0	9677	31110	188741	63503	6123	7074	19977	6188	4771
ISPARTA	32	10403	3073	4348	2419	7156	5390	9811	3999	1427	1806	9571	0	12606	76481	25733	2481	2866	8095	2507	1933
IÇEL	33	34018	10050	14220	7909	23401	17624	32084	13076	4667	5906	31297	12823	0	250103	84149	8114	9373	26472	8200	6322
ISTANBUL	34	236265	69803	98759	54934	162524	122407	222835	90819	32414	41018	217370	89061	286318	0	584437	56351	65100	183857	56948	43906
IZMIR	35	71291	21063	29800	16576	49040	36935	67239	27404	9781	12377	65590	26874	86394	524138	0	17003	19643	55477	17184	13248
KARS	36	6564	1939	2744	1526	4515	3401	6190	2523	900	1140	6039	2474	7954	48256	16236	0	1809	5108	1582	1220
KASTAMONU	37	7588	2242	3172	1764	5220	3931	7157	2917	1041	1317	6981	2860	9196	55789	18771	1810	0	5905	1829	1410
KAYSERI	38	21651	6397	9050	5034	14893	11217	20420	8323	2970	3759	19919	8161	26238	159179	53557	5164	5966	0	5219	4024
KIRKLARELI	39	6633	1960	2773	1542	4563	3437	6256	2550	910	1152	6103	2501	8039	48770	16409	1582	1828	5162	0	1233
KIRŞEHIR	40	5109	1509	2135	1188	3514	2647	4818	1964	701	887	4700	1926	6191	37559	12637	1218	1408	3975	1231	0

Table D.14: Turkish Network Data: Volume distribution matrix (B)

City name	*Node id*	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
ADANA	1	33821	61472	18421	23938	35337	28109	19772	20059	12721	8691	9761	24894	10262	21204	33906	7394	6325	21174	17487	23219
ADIYAMAN	2	11199	20356	6100	7927	11701	9308	6547	6642	4212	2878	3232	8243	3398	7022	11228	2448	2095	7012	5790	7689
AFYON	3	14626	26585	7966	10352	15282	12156	8551	8675	5502	3758	4221	10766	4438	9170	14663	3198	2736	9157	7562	10042
AĞRI	4	9479	17229	5163	6709	9904	7878	5542	5622	3565	2436	2736	6977	2876	5943	9503	2072	1773	5935	4901	6508
AMASYA	5	6532	11872	3558	4623	6825	5429	3819	3874	2457	1678	1885	4808	1982	4095	6548	1428	1222	4089	3377	4484
ANKARA	6	75770	137718	41269	53629	79168	62973	44296	44939	28500	19470	21867	55772	22989	47505	75962	16565	14171	47437	39176	52019
ANTALYA	7	31387	57048	17095	22215	32794	26086	18349	18615	11806	8065	9058	23103	9523	19678	31466	6862	5870	19650	16228	21548
ARTVIN	8	3424	6223	1865	2423	3577	2846	2002	2031	1288	880	988	2520	1039	2147	3432	749	640	2144	1770	2351
AYDIN	9	17152	31176	9342	12140	17922	14255	10028	10173	6452	4407	4950	12625	5204	10754	17196	3750	3208	10739	8868	11776
BALIKESIR	10	19455	35361	10596	13770	20327	16169	11374	11539	7318	4999	5615	14320	5903	12197	19504	4253	3639	12180	10059	13356
BILECIK	11	3467	6301	1888	2454	3622	2881	2027	2056	1304	891	1000	2552	1052	2173	3475	758	648	2170	1792	2380
BINGÖL	12	4530	8234	2468	3207	4734	3765	2649	2687	1704	1164	1307	3335	1375	2840	4542	990	847	2836	2342	3110
BITLIS	13	6954	12639	3787	4922	7265	5779	4065	4124	2616	1787	2007	5118	2110	4360	6971	1520	1301	4353	3595	4774
BOLU	14	4834	8786	2633	3421	5050	4017	2826	2867	1818	1242	1395	3558	1467	3031	4846	1057	904	3026	2499	3318
BURDUR	15	4585	8334	2497	3245	4791	3811	2681	2720	1725	1178	1323	3375	1391	2875	4597	1002	858	2871	2371	3148
BURSA	16	39025	70931	21255	27621	40775	32434	22815	23146	14679	10028	11263	28725	11840	24467	39124	8532	7299	24432	20177	26792
ÇANAKKALE	17	8328	15137	4536	5895	8701	6921	4869	4939	3132	2140	2403	6130	2527	5221	8349	1821	1558	5214	4306	5718
ÇANKIRI	18	4828	8776	2630	3417	5045	4013	2823	2864	1816	1241	1393	3554	1465	3027	4840	1056	903	3023	2496	3315
ÇORUM	19	10715	19475	5836	7584	11195	8905	6264	6355	4030	2753	3092	7887	3251	6718	10742	2342	2004	6708	5540	7356
DENIZLI	20	15312	27831	8340	10838	15999	12726	8952	9082	5759	3935	4419	11271	4646	9600	15351	3348	2864	9586	7917	10512
DIYARBAKIR	21	24737	44961	13473	17509	25846	20559	14462	14671	9304	6356	7139	18208	7505	15509	24799	5408	4627	15487	12790	16983
EDIRNE	22	7204	13094	3924	5099	7527	5988	4212	4273	2710	1851	2079	5303	2186	4517	7222	1575	1347	4510	3725	4946
ELAZIĞ	23	10218	18572	5565	7232	10676	8492	5974	6060	3843	2626	2949	7521	3100	6406	10244	2234	1911	6397	5283	7015
ERZINCAN	24	5662	10292	3084	4008	5916	4706	3310	3358	2130	1455	1634	4168	1718	3550	5677	1238	1059	3545	2928	3887
ERZURUM	25	16908	30732	9209	11967	17666	14052	9885	10028	6360	4345	4880	12445	5130	10601	16951	3696	3162	10585	8742	11608
ESKIŞEHIR	26	12691	23066	6912	8982	13260	10547	7419	7527	4773	3261	3663	9341	3850	7956	12723	2774	2374	7945	6561	8713
GAZIANTEP	27	23304	42356	12692	16494	24349	19368	13624	13821	8765	5988	6725	17153	7071	14610	23362	5095	4358	14590	12049	15999
GIRESUN	28	9390	17067	5114	6646	9811	7804	5490	5569	3532	2413	2710	6912	2849	5887	9414	2053	1756	5879	4855	6447
GÜMÜŞHANE	29	3335	6061	1816	2360	3484	2771	1950	1978	1254	857	962	2455	1012	2091	3343	729	624	2088	1724	2289
HAKKARI	30	4223	7676	2300	2989	4412	3510	2469	2505	1588	1085	1219	3108	1281	2648	4234	923	790	2644	2183	2899
HATAY	31	22721	41298	12375	16082	23740	18884	13283	13476	8546	5838	6557	16724	6894	14245	22779	4967	4250	14225	11748	15599
ISPARTA	32	9207	16735	5015	6517	9620	7652	5383	5461	3463	2366	2657	6777	2794	5772	9230	2013	1722	5764	4760	6321
IÇEL	33	30108	54724	16399	21310	31458	25023	17602	17857	11325	7737	8689	22162	9135	18877	30184	6582	5631	18850	15567	20670
ISTANBUL	34	209110	380075	113893	148006	218487	173792	122249	124023	78654	53733	60350	153920	63446	131104	209639	45716	39110	130917	108118	143562
IZMIR	35	63097	114685	34366	44660	65927	52441	36888	37423	23733	16213	18210	46444	19144	39560	63257	13794	11801	39503	32624	43319
KARS	36	5809	10559	3164	4112	6070	4828	3396	3445	2185	1493	1677	4276	1763	3642	5824	1270	1086	3637	3004	3988
KASTAMONU	37	6716	12207	3658	4754	7017	5582	3926	3983	2526	1726	1938	4944	2038	4211	6733	1468	1256	4205	3472	4611
KAYSERI	38	19162	34830	10437	13563	20022	15926	11203	11365	7208	4924	5530	14105	5814	12014	19211	4189	3584	11997	9908	13156
KIRKLARELI	39	5871	10671	3198	4155	6134	4879	3432	3482	2208	1509	1694	4322	1781	3681	5886	1284	1098	3676	3036	4031
KIRŞEHIR	40	4521	8218	2463	3200	4724	3758	2643	2682	1701	1162	1305	3328	1372	2835	4533	988	846	2831	2338	3104

Table D.15: Turkish Network Data: Volume distribution matrix (C)

City name	*Node id*	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
ADANA	1	27345	2624	40476	9038	24607	19150	17262	11107	2730	6820	10754	12808	9904	5165	3751	4729	4728	6312	3217	12865	8813
ADIYAMAN	2	9055	869	13403	2993	8148	6341	5716	3678	904	2258	3561	4241	3280	1710	1242	1566	1565	2090	1065	4260	2918
AFYON	3	11826	1135	17505	3909	10642	8282	7465	4803	1181	2949	4651	5539	4283	2234	1622	2045	2045	2730	1391	5564	3811
AĞRI	4	7664	736	11344	2533	6897	5367	4838	3113	765	1911	3014	3590	2776	1448	1051	1325	1325	1769	902	3606	2470
AMASYA	5	5281	507	7817	1746	4752	3699	3334	2145	527	1317	2077	2474	1913	997	724	913	913	1219	621	2485	1702
ANKARA	6	61261	5879	90680	20249	55129	42903	38674	24883	6116	15279	24093	28693	22189	11571	8403	10594	10592	14142	7207	28822	19743
ANTALYA	7	25377	2435	37563	8388	22836	17772	16020	10308	2534	6329	9980	11886	9191	4793	3481	4388	4387	5858	2986	11939	8178
ARTVIN	8	2768	266	4098	915	2491	1939	1748	1124	276	690	1089	1297	1003	523	380	479	479	639	326	1302	892
AYDIN	9	13868	1331	20528	4584	12480	9712	8755	5633	1385	3459	5454	6495	5023	2619	1902	2398	2398	3201	1632	6525	4469
BALIKESIR	10	15729	1510	23283	5199	14155	11016	9930	6389	1570	3923	6186	7367	5697	2971	2158	2720	2719	3631	1851	7400	5069
BILECIK	11	2803	269	4149	926	2522	1963	1769	1138	280	699	1102	1313	1015	529	384	485	485	647	330	1319	903
BINGÖL	12	3663	352	5422	1211	3296	2565	2312	1488	366	914	1441	1716	1327	692	502	633	633	846	431	1723	1180
BITLIS	13	5622	540	8322	1858	5059	3937	3549	2284	561	1402	2211	2633	2036	1062	771	972	972	1298	661	2645	1812
BOLU	14	3908	375	5785	1292	3517	2737	2467	1587	390	975	1537	1830	1416	738	536	676	676	902	460	1839	1259
BURDUR	15	3707	356	5488	1225	3336	2596	2340	1506	370	925	1458	1736	1343	700	509	641	641	856	436	1744	1195
BURSA	16	31552	3028	46704	10429	28394	22097	19919	12816	3150	7869	12409	14778	11428	5959	4328	5456	5455	7284	3712	14845	10169
ÇANAKKALE	17	6733	646	9967	2226	6059	4716	4251	2735	672	1679	2648	3154	2439	1272	924	1164	1164	1554	792	3168	2170
ÇANKIRI	18	3904	375	5778	1290	3513	2734	2464	1586	390	974	1535	1828	1414	737	535	675	675	901	459	1837	1258
ÇORUM	19	8663	831	12823	2863	7796	6067	5469	3519	865	2161	3407	4058	3138	1636	1188	1498	1498	2000	1019	4076	2792
DENIZLI	20	12380	1188	18325	4092	11141	8670	7815	5029	1236	3088	4869	5799	4484	2338	1698	2141	2140	2858	1457	5825	3990
DIYARBAKIR	21	20000	1919	29605	6611	17998	14007	12626	8124	1997	4988	7866	9368	7244	3777	2743	3459	3458	4617	2353	9410	6446
EDIRNE	22	5825	559	8622	1925	5242	4079	3677	2366	582	1453	2291	2728	2110	1100	799	1007	1007	1345	685	2740	1877
ELAZIĞ	23	8261	793	12229	2731	7434	5786	5215	3356	825	2061	3249	3869	2992	1560	1133	1429	1428	1907	972	3887	2662
ERZINCAN	24	4578	439	6777	1513	4120	3206	2890	1860	457	1142	1801	2144	1658	865	628	792	792	1057	539	2154	1475
ERZURUM	25	13670	1312	20235	4518	12302	9574	8630	5553	1365	3410	5376	6403	4951	2582	1875	2364	2363	3156	1608	6432	4406
ESKIŞEHIR	26	10260	985	15188	3391	9233	7186	6477	4168	1024	2559	4035	4806	3716	1938	1407	1774	1774	2369	1207	4827	3307
GAZIANTEP	27	18841	1808	27889	6228	16955	13195	11894	7653	1881	4699	7410	8825	6824	3559	2584	3258	3257	4349	2217	8864	6072
GIRESUN	28	7592	729	11238	2509	6832	5317	4793	3084	758	1894	2986	3556	2750	1434	1041	1313	1313	1753	893	3572	2447
GÜMÜŞHANE	29	2696	259	3991	891	2426	1888	1702	1095	269	672	1060	1263	977	509	370	466	466	622	317	1268	869
HAKKARI	30	3414	328	5054	1129	3073	2391	2155	1387	341	852	1343	1599	1237	645	468	590	590	788	402	1606	1100
HATAY	31	18370	1763	27192	6072	16532	12865	11597	7462	1834	4582	7225	8604	6654	3470	2520	3177	3176	4241	2161	8643	5920
ISPARTA	32	7444	714	11019	2460	6699	5213	4699	3024	743	1857	2928	3487	2696	1406	1021	1287	1287	1718	876	3502	2399
IÇEL	33	24343	2336	36033	8046	21906	17048	15368	9888	2430	6071	9574	11402	8817	4598	3339	4210	4209	5619	2864	11453	7845
ISTANBUL	34	169068	16225	250259	55882	152144	118404	106732	68673	16880	42167	66492	79188	61237	31933	23190	29238	29230	39028	19891	79543	54487
IZMIR	35	51015	4896	75514	16862	45908	35727	32206	20721	5093	12724	20064	23894	18478	9635	6998	8822	8820	11776	6002	24002	16441
KARS	36	4697	451	6952	1552	4227	3289	2965	1908	469	1171	1847	2200	1701	887	644	812	812	1084	553	2210	1514
KASTAMONU	37	5430	521	8038	1795	4887	3803	3428	2206	542	1354	2136	2543	1967	1026	745	939	939	1253	639	2555	1750
KAYSERI	38	15493	1487	22933	5121	13942	10850	9781	6293	1547	3864	6093	7257	5612	2926	2125	2679	2679	3576	1823	7289	4993
KIRKLARELI	39	4747	456	7026	1569	4272	3324	2997	1928	474	1184	1867	2223	1719	897	651	821	821	1096	558	2233	1530
KIRŞEHIR	40	3656	351	5411	1208	3290	2560	2308	1485	365	912	1438	1712	1324	690	501	632	632	844	430	1720	1178

Table D.16: Turkish Network Data: Volume distribution matrix (D)

City name	*Node id*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
KOCAELI	41	33494	11297	14713	9576	6614	72582	31145	3476	17218	19493	3519	4595	7039	4902	4651	38486	8421	4896	10813	15394
KONYA	42	61793	20842	27144	17666	12203	133907	57459	6413	31766	35962	6493	8478	12986	9043	8580	71003	15535	9033	19949	28400
KÜTAHYA	43	18094	6103	7948	5173	3573	39209	16824	1878	9301	10530	1901	2482	3802	2648	2512	20790	4549	2645	5841	8316
MALATYA	44	23582	7954	10359	6742	4657	51103	21928	2447	12123	13724	2478	3235	4956	3451	3274	27097	5929	3447	7613	10838
MANISA	45	35024	11813	15385	10013	6917	75899	32568	3635	18005	20383	3680	4805	7361	5125	4863	40245	8805	5120	11307	16097
KAHRAMANMARAŞ	46	27752	9361	12191	7934	5480	60140	25806	2880	14266	16151	2916	3807	5832	4061	3853	31889	6977	4057	8959	12755
MARDIN	47	19435	6555	8537	5556	3838	42116	18072	2017	9991	11311	2042	2666	4084	2844	2699	22332	4886	2841	6274	8932
MUĞLA	48	19720	6651	8662	5638	3894	42734	18337	2046	10137	11476	2072	2705	4144	2886	2738	22659	4958	2883	6366	9063
MUŞ	49	12458	4202	5472	3561	2460	26996	11584	1293	6404	7250	1309	1709	2618	1823	1730	14314	3132	1821	4022	5726
NEVŞEHIR	50	8492	2864	3730	2428	1677	18403	7897	881	4366	4942	892	1165	1785	1243	1179	9758	2135	1241	2742	3903
NIĞDE	51	9544	3219	4192	2728	1885	20681	8874	990	4906	5554	1003	1309	2006	1397	1325	10966	2399	1395	3081	4386
ORDU	52	24537	8276	10778	7015	4845	53172	22816	2546	12614	14280	2578	3366	5157	3591	3407	28194	6169	3587	7921	11277
RIZE	53	10036	3385	4408	2869	1982	21748	9332	1041	5159	5841	1054	1377	2109	1469	1393	11532	2523	1467	3240	4613
SAKARYA	54	20859	7035	9162	5963	4119	45201	19395	2165	10723	12139	2192	2862	4384	3052	2896	23967	5244	3049	6734	9587
SAMSUN	55	33580	11326	14751	9600	6631	72769	31225	3485	17263	19543	3528	4607	7057	4914	4663	38585	8442	4909	10841	15434
SIIRT	56	7220	2435	3172	2064	1426	15647	6714	749	3712	4202	759	991	1517	1057	1003	8297	1815	1055	2331	3318
SINOP	57	6173	2082	2712	1765	1219	13378	5740	641	3174	3593	649	847	1297	903	857	7094	1552	902	1993	2837
SIVAS	58	20828	7025	9149	5955	4113	45136	19368	2162	10707	12122	2188	2858	4377	3048	2892	23933	5236	3045	6724	9573
TEKIRDAĞ	59	17167	5790	7541	4908	3390	37202	15963	1782	8825	9991	1804	2355	3608	2512	2384	19726	4316	2510	5542	7890
TOKAT	60	22865	7712	10044	6537	4515	49549	21261	2373	11754	13307	2402	3137	4805	3346	3175	26273	5749	3342	7382	10509
TRABZON	61	26987	9102	11854	7715	5329	58481	25094	2801	13873	15706	2836	3702	5671	3949	3747	31009	6785	3945	8712	12403
TUNCELI	62	2556	862	1123	731	505	5539	2377	265	1314	1488	269	351	537	374	355	2937	643	374	825	1175
ŞANLIURFA	63	40228	13569	17671	11501	7944	87176	37407	4175	20680	23412	4227	5519	8454	5887	5586	46224	10114	5881	12987	18489
UŞAK	64	8834	2980	3880	2525	1744	19143	8214	917	4541	5141	928	1212	1856	1293	1227	10150	2221	1291	2852	4060
VAN	65	24250	8179	10652	6933	4789	52550	22549	2517	12466	14113	2548	3327	5096	3549	3367	27864	6097	3545	7829	11145
YOZGAT	66	18817	6347	8266	5380	3716	40778	17498	1953	9673	10951	1977	2582	3955	2754	2613	21622	4731	2751	6075	8649
ZONGULDAK	67	16945	5716	7444	4845	3346	36721	15757	1759	8711	9862	1780	2325	3561	2480	2353	19471	4260	2477	5470	7788
AKSARAY	68	10867	3665	4774	3107	2146	23550	10105	1128	5587	6325	1142	1491	2284	1590	1509	12487	2732	1589	3508	4995
BAYBURT	69	2659	897	1168	760	525	5763	2473	276	1367	1548	279	365	559	389	369	3056	669	389	859	1222
KARAMAN	70	6658	2246	2925	1903	1315	14428	6191	691	3423	3875	700	913	1399	974	924	7650	1674	973	2149	3060
KIRIKKALE	71	10520	3548	4621	3008	2078	22798	9782	1092	5408	6123	1105	1443	2211	1540	1461	12088	2645	1538	3396	4835
BATMAN	72	12543	4231	5510	3586	2477	27180	11663	1302	6448	7300	1318	1721	2636	1836	1742	14412	3153	1833	4049	5765
ŞIRNAK	73	9685	3267	4254	2769	1912	20987	9005	1005	4979	5636	1018	1329	2035	1417	1345	11128	2435	1416	3126	4451
BARTIN	74	5037	1699	2213	1440	995	10916	4684	523	2590	2932	529	691	1059	737	699	5788	1266	736	1626	2315
ARDAHAN	75	3656	1233	1606	1045	722	7922	3399	379	1879	2127	384	502	768	535	508	4201	919	534	1180	1680
IĞDIR	76	4611	1555	2026	1318	911	9993	4288	479	2371	2684	485	633	969	675	640	5299	1159	674	1489	2119
YALOVA	77	4610	1555	2025	1318	910	9990	4287	478	2370	2683	484	632	969	675	640	5297	1159	674	1488	2119
KARABÜK	78	6161	2078	2706	1761	1217	13350	5728	639	3167	3585	647	845	1295	902	855	7079	1549	901	1989	2831
KILIS	79	3135	1057	1377	896	619	6793	2915	325	1611	1824	329	430	659	459	435	3602	788	458	1012	1441
OSMANIYE	80	12599	4250	5535	3602	2488	27303	11716	1308	6477	7333	1324	1729	2648	1844	1749	14477	3168	1842	4067	5791
DÜZCE	81	8612	2905	3783	2462	1701	18663	8008	894	4427	5012	905	1182	1810	1260	1196	9896	2165	1259	2780	3958

Table D.17: Turkish Network Data: Volume distribution matrix (E)

City name	*Node id*	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
KOCAELI	41	24679	7291	10316	5738	16976	12786	23276	9486	3386	4284	22705	9303	29907	181439	61046	5886	6800	19204	5948	4586
KONYA	42	45530	13452	19032	10586	31319	23589	42942	17501	6246	7904	41888	17163	55175	334738	112625	10859	12545	35430	10974	8461
KÜTAHYA	43	13331	3939	5573	3100	9171	6907	12574	5125	1829	2314	12265	5025	16156	98014	32977	3180	3673	10374	3213	2477
MALATYA	44	17375	5133	7263	4040	11952	9002	16388	6679	2384	3017	15986	6550	21056	127745	42981	4144	4788	13521	4188	3229
MANISA	45	25806	7624	10787	6000	17752	13370	24339	9920	3540	4480	23742	9728	31273	189729	63836	6155	7111	20082	6220	4796
KAHRAMANMARAŞ	46	20448	6041	8547	4754	14066	10594	19286	7860	2805	3550	18813	7708	24780	150335	50581	4877	5634	15912	4929	3800
MARDIN	47	14320	4231	5986	3329	9850	7419	13506	5504	1965	2486	13175	5398	17353	105280	35422	3415	3946	11143	3452	2661
MUĞLA	48	14530	4293	6073	3378	9995	7528	13704	5585	1993	2523	13368	5477	17608	106824	35942	3465	4003	11307	3502	2700
MUŞ	49	9179	2712	3837	2134	6314	4755	8657	3528	1259	1594	8445	3460	11123	67484	22705	2189	2529	7143	2212	1706
NEVŞEHIR	50	6257	1849	2616	1455	4304	3242	5902	2405	858	1086	5757	2359	7583	46003	15478	1492	1724	4869	1508	1163
NIĞDE	51	7032	2077	2939	1635	4837	3643	6632	2703	965	1221	6469	2651	8521	51698	17394	1677	1938	5472	1695	1307
ORDU	52	18079	5341	7557	4203	12436	9366	17051	6949	2480	3139	16633	6815	21909	132917	44721	4312	4981	14069	4358	3360
RIZE	53	7394	2185	3091	1719	5087	3831	6974	2842	1014	1284	6803	2787	8961	54365	18291	1764	2037	5754	1782	1374
SAKARYA	54	15369	4541	6424	3573	10572	7962	14495	5908	2108	2668	14140	5793	18625	112992	38017	3666	4235	11960	3704	2856
SAMSUN	55	24742	7310	10342	5753	17020	12819	23336	9511	3394	4296	22763	9327	29984	181906	61204	5901	6817	19254	5964	4598
SIIRT	56	5320	1572	2224	1237	3660	2756	5018	2045	730	924	4895	2005	6447	39113	13160	1269	1466	4140	1282	989
SINOP	57	4549	1344	1901	1058	3129	2357	4290	1748	624	790	4185	1715	5512	33442	11252	1085	1253	3540	1096	845
SIVAS	58	15347	4534	6415	3568	10557	7951	14474	5899	2105	2664	14119	5785	18598	112829	37962	3660	4229	11942	3699	2852
TEKIRDAĞ	59	12649	3737	5287	2941	8701	6553	11930	4862	1735	2196	11638	4768	15329	92997	31290	3017	3485	9843	3049	2351
TOKAT	60	16847	4977	7042	3917	11589	8728	15890	6476	2311	2925	15500	6351	20416	123862	41674	4018	4642	13110	4061	3131
TRABZON	61	19884	5875	8312	4623	13678	10302	18754	7643	2728	3452	18294	7495	24097	146189	49186	4742	5479	15473	4793	3695
TUNCELI	62	1883	556	787	438	1296	976	1776	724	258	327	1733	710	2282	13847	4659	449	519	1466	454	350
ŞANLIURFA	63	29641	8757	12390	6892	20389	15357	27956	11394	4066	5146	27270	11173	35920	217920	73320	7069	8167	23066	7144	5508
UŞAK	64	6509	1923	2721	1513	4477	3372	6139	2502	893	1130	5988	2453	7888	47853	16100	1552	1793	5065	1569	1210
VAN	65	17868	5279	7469	4154	12291	9257	16852	6868	2451	3102	16439	6735	21653	131363	44198	4262	4923	13904	4307	3320
YOZGAT	66	13865	4096	5796	3224	9537	7183	13077	5330	1902	2407	12756	5226	16802	101935	34297	3307	3820	10789	3342	2577
ZONGULDAK	67	12486	3689	5219	2903	8589	6469	11776	4799	1713	2168	11487	4706	15131	91795	30885	2978	3440	9716	3009	2320
AKSARAY	68	8007	2366	3347	1862	5508	4148	7552	3078	1099	1390	7367	3018	9704	58869	19807	1910	2206	6231	1930	1488
BAYBURT	69	1959	579	819	456	1348	1015	1848	753	269	340	1803	739	2375	14406	4847	467	540	1525	472	364
KARAMAN	70	4906	1449	2051	1141	3374	2542	4627	1886	673	852	4513	1849	5945	36066	12135	1170	1352	3817	1182	912
KIRIKKALE	71	7752	2290	3240	1802	5332	4016	7311	2980	1063	1346	7132	2922	9394	56990	19175	1849	2136	6032	1868	1441
BATMAN	72	9242	2730	3863	2149	6357	4788	8716	3552	1268	1604	8502	3484	11199	67945	22860	2204	2546	7192	2228	1717
ŞIRNAK	73	7136	2108	2983	1659	4909	3697	6730	2743	979	1239	6565	2690	8647	52462	17651	1702	1966	5553	1720	1326
BARTIN	74	3712	1097	1551	863	2553	1923	3501	1427	509	644	3415	1399	4498	27288	9181	885	1023	2888	895	690
ARDAHAN	75	2694	796	1126	626	1853	1395	2540	1035	370	468	2478	1015	3264	19803	6663	642	742	2096	649	501
IĞDIR	76	3398	1004	1420	790	2337	1760	3204	1306	466	590	3126	1281	4117	24980	8405	810	936	2644	819	631
YALOVA	77	3397	1004	1420	790	2337	1760	3204	1306	466	590	3125	1280	4116	24973	8402	810	936	2643	819	631
KARABÜK	78	4539	1341	1897	1055	3122	2352	4281	1745	623	788	4176	1711	5501	33372	11228	1083	1251	3532	1094	844
KILIS	79	2310	682	965	537	1589	1197	2178	888	317	401	2125	871	2799	16980	5713	551	636	1797	557	429
OSMANIYE	80	9283	2743	3880	2158	6386	4810	8756	3568	1274	1612	8541	3499	11250	68252	22964	2214	2558	7224	2238	1725
DÜZCE	81	6345	1875	2652	1475	4365	3288	5985	2439	871	1102	5838	2392	7690	46652	15696	1513	1748	4938	1529	1179

Table D.18: Turkish Network Data: Volume distribution matrix (F)

City name	*Node id*	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
KOCAELI	41	0	39700	11896	15460	22822	18153	12769	12955	8216	5613	6304	16077	6627	13694	21897	4775	4085	13675	11293	14996
KONYA	42	40297	0	21948	28522	42104	33491	23558	23900	15157	10355	11630	29661	12226	25264	40399	8810	7537	25228	20835	27665
KÜTAHYA	43	11799	21446	0	8351	12328	9806	6898	6998	4438	3032	3405	8685	3580	7398	11829	2580	2207	7387	6101	8101
MALATYA	44	15378	27951	8376	0	16068	12781	8990	9121	5784	3952	4438	11320	4666	9642	15417	3362	2876	9628	7951	10558
MANISA	45	22840	41514	12440	16166	0	18983	13353	13546	8591	5869	6592	16812	6930	14320	22898	4993	4272	14299	11809	15681
KAHRAMANMARAŞ	46	18098	32894	9857	12809	18909	0	10580	10734	6807	4650	5223	13321	5491	11347	18144	3957	3385	11330	9357	12425
MARDIN	47	12674	23036	6903	8971	13242	10533	0	7517	4767	3257	3658	9329	3845	7946	12706	2771	2370	7935	6553	8701
MUĞLA	48	12860	23374	7004	9102	13436	10688	7518	0	4837	3304	3711	9466	3902	8063	12892	2811	2405	8051	6649	8829
MUŞ	49	8124	14766	4425	5750	8488	6752	4749	4818	0	2088	2345	5980	2465	5093	8144	1776	1519	5086	4200	5577
NEVŞEHIR	50	5538	10066	3016	3920	5786	4603	3238	3285	2083	0	1598	4076	1680	3472	5552	1211	1036	3467	2863	3802
NIĞDE	51	6224	11312	3390	4405	6503	5172	3638	3691	2341	1599	0	4581	1888	3902	6239	1361	1164	3896	3218	4273
ORDU	52	16001	29083	8715	11325	16718	13298	9354	9490	6019	4112	4618	0	4855	10032	16041	3498	2993	10018	8273	10985
RIZE	53	6545	11895	3565	4632	6838	5439	3826	3882	2462	1682	1889	4817	0	4103	6561	1431	1224	4097	3384	4493
SAKARYA	54	13602	24723	7409	9628	14212	11305	7952	8068	5116	3495	3926	10012	4127	0	13637	2974	2544	8516	7033	9339
SAMSUN	55	21898	39802	11927	15500	22880	18200	12802	12988	8237	5627	6320	16119	6644	13729	0	4787	4096	13710	11322	15034
SIIRT	56	4709	8558	2565	3333	4920	3913	2753	2793	1771	1210	1359	3466	1429	2952	4720	0	881	2948	2434	3233
SINOP	57	4026	7317	2193	2849	4206	3346	2354	2388	1514	1034	1162	2963	1221	2524	4036	880	0	2520	2082	2764
SIVAS	58	13583	24688	7398	9614	14192	11289	7941	8056	5109	3490	3920	9998	4121	8516	13617	2969	2540	0	7023	9325
TEKIRDAĞ	59	11195	20348	6098	7924	11697	9304	6545	6640	4211	2877	3231	8241	3397	7019	11224	2448	2094	7009	0	7686
TOKAT	60	14911	27102	8121	10554	15580	12393	8717	8844	5609	3831	4303	10975	4524	9349	14949	3260	2789	9335	7709	0
TRABZON	61	17599	31987	9585	12456	18388	14626	10288	10438	6620	4522	5079	12954	5340	11034	17643	3847	3291	11018	9099	12082
TUNCELI	62	1667	3030	908	1180	1742	1385	975	989	627	428	481	1227	506	1045	1671	364	312	1044	862	1144
ŞANLIURFA	63	26234	47682	14288	18568	27410	21803	15337	15559	9868	6741	7571	19310	7960	16448	26300	5735	4907	16424	13564	18011
UŞAK	64	5761	10470	3138	4077	6019	4788	3368	3417	2167	1480	1663	4240	1748	3612	5775	1259	1077	3607	2978	3955
VAN	65	15814	28743	8613	11193	16523	13143	9245	9379	5948	4064	4564	11640	4798	9915	15854	3457	2958	9901	8176	10857
YOZGAT	66	12271	22304	6684	8685	12822	10199	7174	7278	4616	3153	3542	9033	3723	7694	12302	2683	2295	7683	6345	8425
ZONGULDAK	67	11051	20085	6019	7821	11546	9184	6460	6554	4157	2840	3189	8134	3353	6928	11078	2416	2067	6918	5714	7587
AKSARAY	68	7087	12881	3860	5016	7405	5890	4143	4203	2666	1821	2045	5216	2150	4443	7105	1549	1325	4437	3664	4865
BAYBURT	69	1734	3152	945	1228	1812	1441	1014	1029	652	446	501	1277	526	1087	1739	379	324	1086	897	1191
KARAMAN	70	4342	7892	2365	3073	4536	3608	2538	2575	1633	1116	1253	3196	1317	2722	4353	949	812	2718	2245	2981
KIRIKKALE	71	6861	12470	3737	4856	7168	5702	4011	4069	2581	1763	1980	5050	2082	4301	6878	1500	1283	4295	3547	4710
BATMAN	72	8179	14867	4455	5789	8546	6798	4782	4851	3077	2102	2361	6021	2482	5128	8200	1788	1530	5121	4229	5615
ŞIRNAK	73	6316	11479	3440	4470	6599	5249	3692	3746	2376	1623	1823	4649	1916	3960	6331	1381	1181	3954	3265	4336
BARTIN	74	3285	5971	1789	2325	3432	2730	1920	1948	1236	844	948	2418	997	2060	3293	718	614	2057	1698	2255
ARDAHAN	75	2384	4333	1298	1687	2491	1981	1394	1414	897	613	688	1755	723	1495	2390	521	446	1493	1233	1637
IĞDIR	76	3007	5466	1638	2128	3142	2499	1758	1784	1131	773	868	2213	912	1885	3015	657	562	1883	1555	2065
YALOVA	77	3006	5464	1637	2128	3141	2499	1758	1783	1131	773	868	2213	912	1885	3014	657	562	1882	1554	2064
KARABÜK	78	4017	7302	2188	2843	4198	3339	2349	2383	1511	1032	1159	2957	1219	2519	4028	878	751	2515	2077	2758
KILIS	79	2044	3715	1113	1447	2136	1699	1195	1212	769	525	590	1505	620	1282	2049	447	382	1280	1057	1403
OSMANIYE	80	8216	14934	4475	5815	8585	6829	4803	4873	3090	2111	2371	6048	2493	5151	8237	1796	1537	5144	4248	5641
DÜZCE	81	5616	10208	3059	3975	5868	4668	3283	3331	2112	1443	1621	4134	1704	3521	5630	1228	1050	3516	2904	3856

Table D.19: Turkish Network Data: Volume distribution matrix (G)

City name	*Node id*	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
KOCAELI	41	17660	1695	26140	5837	15892	12368	11148	7173	1763	4405	6945	8271	6396	3335	2422	3054	3053	4077	2078	8309	5691
KONYA	42	32580	3127	48226	10769	29319	22817	20568	13234	3253	8126	12813	15260	11801	6154	4469	5634	5633	7521	3833	15328	10500
KÜTAHYA	43	9540	916	14121	3153	8585	6681	6022	3875	952	2379	3752	4468	3455	1802	1309	1650	1649	2202	1122	4488	3074
MALATYA	44	12434	1193	18405	4110	11189	8708	7849	5050	1241	3101	4890	5824	4503	2348	1705	2150	2150	2870	1463	5850	4007
MANISA	45	18467	1772	27335	6104	16618	12933	11658	7501	1844	4606	7263	8649	6689	3488	2533	3193	3193	4263	2173	8688	5951
KAHRAMANMARAŞ	46	14632	1404	21659	4836	13168	10247	9237	5943	1461	3649	5755	6853	5300	2764	2007	2530	2530	3378	1721	6884	4716
MARDIN	47	10247	983	15168	3387	9221	7176	6469	4162	1023	2556	4030	4800	3712	1935	1406	1772	1772	2365	1206	4821	3302
MUĞLA	48	10397	998	15390	3437	9357	7282	6564	4223	1038	2593	4089	4870	3766	1964	1426	1798	1798	2400	1223	4892	3351
MUŞ	49	6568	630	9723	2171	5911	4600	4147	2668	656	1638	2583	3076	2379	1241	901	1136	1136	1516	773	3090	2117
NEVŞEHIR	50	4478	430	6628	1480	4029	3136	2827	1819	447	1117	1761	2097	1622	846	614	774	774	1034	527	2107	1443
NIĞDE	51	5032	483	7448	1663	4528	3524	3177	2044	502	1255	1979	2357	1823	950	690	870	870	1162	592	2367	1622
ORDU	52	12937	1242	19150	4276	11642	9060	8167	5255	1292	3227	5088	6059	4686	2443	1775	2237	2237	2986	1522	6087	4169
RIZE	53	5291	508	7832	1749	4762	3706	3340	2149	528	1320	2081	2478	1917	999	726	915	915	1221	623	2489	1705
SAKARYA	54	10998	1055	16279	3635	9897	7702	6943	4467	1098	2743	4325	5151	3983	2077	1509	1902	1901	2539	1294	5174	3544
SAMSUN	55	17705	1699	26208	5852	15933	12400	11177	7192	1768	4416	6963	8293	6413	3344	2429	3062	3061	4087	2083	8330	5706
SIIRT	56	3807	365	5635	1258	3426	2666	2403	1546	380	949	1497	1783	1379	719	522	658	658	879	448	1791	1227
SINOP	57	3255	312	4818	1076	2929	2280	2055	1322	325	812	1280	1525	1179	615	446	563	563	751	383	1531	1049
SIVAS	58	10982	1054	16256	3630	9883	7691	6933	4461	1096	2739	4319	5144	3978	2074	1506	1899	1899	2535	1292	5167	3539
TEKIRDAĞ	59	9052	869	13398	2992	8145	6339	5714	3677	904	2258	3560	4240	3278	1710	1242	1565	1565	2089	1065	4259	2917
TOKAT	60	12056	1157	17845	3985	10849	8443	7611	4897	1204	3007	4741	5647	4367	2277	1654	2085	2084	2783	1418	5672	3885
TRABZON	61	0	1366	21062	4703	12804	9965	8983	5779	1421	3549	5596	6664	5154	2687	1952	2461	2460	3285	1674	6694	4586
TUNCELI	62	1348	0	1995	445	1213	944	851	547	135	336	530	631	488	255	185	233	233	311	159	634	434
ŞANLIURFA	63	21210	2036	0	7011	19087	14854	13390	8615	2118	5290	8342	9935	7682	4006	2909	3668	3667	4896	2495	9979	6836
UŞAK	64	4658	447	6894	0	4191	3262	2940	1892	465	1162	1832	2182	1687	880	639	805	805	1075	548	2191	1501
VAN	65	12786	1227	18926	4226	0	8954	8072	5193	1277	3189	5028	5989	4631	2415	1754	2211	2211	2951	1504	6015	4121
YOZGAT	66	9921	952	14686	3279	8928	0	6263	4030	991	2475	3902	4647	3594	1874	1361	1716	1715	2290	1167	4668	3197
ZONGULDAK	67	8934	857	13225	2953	8040	6257	0	3629	892	2228	3514	4185	3236	1687	1226	1545	1545	2062	1051	4203	2879
AKSARAY	68	5730	550	8481	1894	5156	4013	3617	0	572	1429	2253	2684	2075	1082	786	991	991	1323	674	2696	1847
BAYBURT	69	1402	135	2076	463	1262	982	885	570	0	350	551	657	508	265	192	242	242	324	165	660	452
KARAMAN	70	3510	337	5196	1160	3159	2458	2216	1426	350	0	1381	1644	1271	663	482	607	607	810	413	1652	1131
KIRIKKALE	71	5547	532	8211	1833	4992	3885	3502	2253	554	1383	0	2598	2009	1048	761	959	959	1280	653	2610	1788
BATMAN	72	6613	635	9789	2186	5951	4631	4175	2686	660	1649	2601	0	2395	1249	907	1144	1143	1527	778	3111	2131
ŞIRNAK	73	5106	490	7558	1688	4595	3576	3224	2074	510	1274	2008	2392	0	964	700	883	883	1179	601	2402	1646
BARTIN	74	2656	255	3931	878	2390	1860	1677	1079	265	662	1045	1244	962	0	364	459	459	613	312	1250	856
ARDAHAN	75	1927	185	2853	637	1735	1350	1217	783	192	481	758	903	698	364	0	333	333	445	227	907	621
IĞDIR	76	2431	233	3599	804	2188	1703	1535	988	243	606	956	1139	881	459	333	0	420	561	286	1144	784
YALOVA	77	2431	233	3598	803	2187	1702	1534	987	243	606	956	1138	880	459	333	420	0	561	286	1144	783
KARABÜK	78	3248	312	4808	1074	2923	2275	2051	1319	324	810	1277	1521	1176	613	446	562	562	0	382	1528	1047
KILIS	79	1653	159	2446	546	1487	1157	1043	671	165	412	650	774	599	312	227	286	286	382	0	778	533
OSMANIYE	80	6643	638	9833	2196	5978	4652	4194	2698	663	1657	2613	3111	2406	1255	911	1149	1149	1533	782	0	2141
DÜZCE	81	4541	436	6721	1501	4086	3180	2867	1844	453	1133	1786	2127	1645	858	623	785	785	1048	534	2136	0

Table D.20: Turkish Network Data: Volume distribution matrix (H)

City name	*Node id*	*Latitude*	*Longitude*
ADANA	1	36.991859	35.325851
ADIYAMAN	2	37.76393	38.277054
AFYON	3	38.763453	30.540276
AĞRI	4	39.721844	43.056622
AMASYA	5	40.649974	35.833454
ANKARA	6	39.920796	32.853928
ANTALYA	7	36.884838	30.706444
ARTVIN	8	41.18337	41.81654
AYDIN	9	37.84463	27.84605
BALIKESIR	10	39.648526	27.882614
BILECIK	11	40.150407	29.982891
BINGÖL	12	38.885621	40.498095
BITLIS	13	38.400335	42.116776
BOLU	14	40.739193	31.611528
BURDUR	15	37.726601	30.288448
BURSA	16	40.183595	29.067764
ÇANAKKALE	17	40.155392	26.414223
ÇANKIRI	18	40.599617	33.616276
ÇORUM	19	40.550331	34.955921
DENIZLI	20	37.775871	29.08699
DIYARBAKIR	21	37.914409	40.23056
EDIRNE	22	41.666372	26.56683
ELAZIĞ	23	38.680284	39.22617
ERZINCAN	24	39.74983	39.500055
ERZURUM	25	39.903942	41.268597
ESKIŞEHIR	26	39.78374	30.51899
GAZIANTEP	27	37.065725	37.383556
GIRESUN	28	40.912734	38.389664
GÜMÜŞHANE	29	40.459944	39.481516
HAKKARI	30	37.583086	43.733482
HATAY	31	36.19982	36.166821
ISPARTA	32	37.76678	30.550404
IÇEL	33	36.799938	34.63131
ISTANBUL	34	41.004775	28.97644
IZMIR	35	38.418628	27.128677
KARS	36	40.605091	43.096619
KASTAMONU	37	41.388401	33.782272
KAYSERI	38	38.731858	35.484123
KIRKLARELI	39	41.735518	27.224379
KIRŞEHIR	40	39.145905	34.159927

Table D.21: Turkish Network Data: Geocodes of node 1 to 40

City name	*Node id*	*Latitude*	*Longitude*
KOCAELI	41	40.852254	29.881439
KONYA	42	40.852254	29.881439
KÜTAHYA	43	39.416701	29.982719
MALATYA	44	38.353503	38.30864
MANISA	45	38.618748	27.428398
KAHRAMANMARAŞ	46	37.583358	36.932945
MARDIN	47	37.312154	40.735159
MUĞLA	48	37.215634	28.363523
MUŞ	49	38.743239	41.506176
NEVŞEHIR	50	38.624381	34.723492
NIĞDE	51	37.966565	34.683409
ORDU	52	40.983138	37.883263
RIZE	53	41.020189	40.523243
SAKARYA	54	40.75688	30.377197
SAMSUN	55	41.29277	36.331615
SIIRT	56	37.928086	41.942196
SINOP	57	42.026471	35.154705
SIVAS	58	39.747454	37.017746
TEKIRDAĞ	59	40.977565	27.510109
TOKAT	60	40.31697	36.551685
TRABZON	61	41.004257	39.718151
TUNCELI	62	39.10662	39.547305
ŞANLIURFA	63	37.155255	38.789635
UŞAK	64	38.673315	29.406967
VAN	65	38.4931	43.379517
YOZGAT	66	39.820601	34.806747
ZONGULDAK	67	41.455979	31.79821
AKSARAY	68	38.369925	34.027405
BAYBURT	69	40.255064	40.224853
KARAMAN	70	37.17933	33.224545
KIRIKKALE	71	39.847558	33.513622
BATMAN	72	37.885557	41.129723
ŞIRNAK	73	37.520755	42.455721
BARTIN	74	41.634305	32.33757
ARDAHAN	75	41.110432	42.702141
IĞDIR	76	39.921059	44.041958
YALOVA	77	40.65056	29.267235
KARABÜK	78	41.200873	32.627506
KILIS	79	36.717696	37.117739
OSMANIYE	80	37.06956	36.253681
DÜZCE	81	40.843035	31.157913

Table D.22: Turkish Network Data: Geocodes of node 41 to 81

Appendix E

Verification and Validation

E.1 Expert session

Below is a collection of all the responses of the interviewed experts on questions regarding the verification, validation and user experience of POHST. These five experts all have university grades in either econometrics, mathematics or mathematical engineering. Some of them work as programmers; they program and test mathematical models. Others work as analysts and use these models to analyse network characteristics.

E.1.1 Verification

The verification is used to provide confidence that the model in its coded form is structured as intended by the model design and behaves as envisioned.

1. What are the strengths of the model structure?
    - Arnoud Kuiper (AK): It is an intuitive, logical basic model. The model is fast and it would be easy to expand and adjust.
    - Ferdy Niks (FN): It is fast. It structurally finds a starting solution, which should accelerate the next steps of the model.
    - Frans van Helden (FvH): The information is organized at the level of the locations, such that fine tuning and data editing can be done by local managers, without having to access the whole model.
    - Timon van Dijk (TvD): It could be useful if a certain driving time between hub locations is desirable. For instance less than 9 hours because of driving time. The model structure is fast and simple and it gives a better start solution than a random selection of locations.

- Tom Plat (TP): It is graphical and visual (you can see what is happening in the model).

2. What are the weaknesses of the model structure?

   - AK: It is not possible to design a network with a given (preset) amount of hubs. Limiting the model to a transport movement from a depot to a hub, to another hub and then a depot is not realistic. You have to model movements from depots to other depots and also movements from a depot to a hub and then to a depot as well.
   - FN: A greedy heuristic (algorithm type) only finds a local optimum and does not guarantee a global optimum. However, when it is only designed to find a starting solution, this does not matter. The transport assumption is depot-hub-hub-depot. It does not consider other flows (with the exception of close region combinations). The added value of agent based modelling does not become clear in this proof of concept.
   - FvH: The local organization structure can make the model relatively slow, due to the necessary communication. The part of the model showed here has as weakness that imbalances in volume are not considered.
   - TvD: The results are very sensitive to the adjustments of the parameters. Driving times between locations are often more important for the results than the distribution of the volumes. The driving time to the region centre is not considered in calculating the region-region flows. Hub locations are chosen based on region-region flows, which might not be good for the consolidation in other directions.
   - TP: Missing data analysis such as volumes to give insight (bubble charts would be handy). More guidance for users: what to do where and why?

3. Considering the identified strengths and weaknesses, do you believe this model to be well constructed?

   - AK: Yes.
   - FN: When it is used to generate a starting solution: yes.
   - FvH: I think the model was implemented correctly
   - TvD: It can do a good job in quickly making a start solution.
   - TP: Yes, it serves its purpose to generate a starting solution.

4. Do you expect this model to behave as intended by the design?

   - AK: Yes.
   - FN: Yes.
   - TvD: Yes, although I have not seen any code or done severe testing, the results of the examples I have seen were as expected.
   - TP: I can't say.

E.1.2 Validation

The validation is used to provide confidence that the model outcomes are useful in real life.

1. Do you consider the model to meet all the design requirements? In case not, please state the requirements and the reason(s) behind your judgement.
    - FvH: The model does select proper locations. However, the way the model selects these locations could be more sophisticated.
    - TvD: The proof of concept does not have the same scope as the whole model, therefore it does not meet all requirements like minimizing cost and the service commitments.
2. Do the model outcomes change as you would expect when altering the t^{max} parameter? Parameter t^{max} defines the radius of regions in travel time.
    - FN: Yes.
    - FvH: Yes.
    - TvD: Yes.
    - TP: Yes, we saw LA (CAB data) pop up as hub, which made sense because it has a lot of volume.
3. Do the model outcomes change as you would expect when altering the t^{min} parameter? Parameter t^{min} is used to determine whether a region combination is distant or close.
    - FN: Yes.
4. Do the model outcomes change as you would expect when altering the t^{close} parameter? Parameter t^{close} defines how far hubs should be at the least.
    - FN: Yes
5. Do you consider the Hub Location Problem to be well represented by the model?
    - AK: This proof of concept only generates an initial solution. How well the HLP is represented would be determined in the next step. To serve the goal of this proof of concept, the model is well represented.
    - FN: Yes.
    - FvH: As said before, the model selects depots in a relatively rough estimation. To build an initial solution, this is good enough. If this would be the whole solution, it would be too rough, I feel.
    - TvD: No cost or time constraints are taken into account in the generation of the start solution.
    - TP: Not yet, the testing was too superficial to say if the solutions are good.

E.1.3 User experience

1. What are the strengths of the user interface?
   - AK: It is straight-forward, easy to use.
   - FN: It is simple. The map provides good insight into the model. By decelerating the iterations you can easily see which areas are the most interesting to focus on.
   - FvH: Very intuitive, nice visualisation with the map.
   - TvD: It is a simple interface with nice graphical visualization of the results.
   - TP: Its graphics.
2. What improvements can be made to the user interface?
   - AK: Providing geographical insight to the different transport flows.
   - FN: Renaming the parameters and functions. For example: give the t-parameters a logical name. Display the results more clearly.
   - FvH: More data should be accessible, like distances, driving times, etc.
   - TvD: adding units at the parameters.
   - TP: A bubble chart and other graphics to show why which hubs are selected.
3. Do you find the application useful?
   (a) What are the strengths of the application?
      - FN: It is quite useful to quickly gain insight into which hubs are interesting to research.
      - FvH: It provides easy and quick insight in the data.
      - TvD: Same answer as verification question 1: It could be useful if a certain driving time between hub locations is desirable. For instance less than 9 hours because of driving time. The model structure is fast and simple and it gives a better start solution than a random selection of locations. And user experience question 1: It is a simple interface with nice graphical visualization of the results.

   (b) What are the weaknesses of the application?
      - FN: It is not very user-friendly. The user needs guidance to know the meaning of the parameters and the format of the inputs.
      - FvH: It is not very flexible in the input and output of data.
      - TvD: Same as verification question 2: The results are very sensitive to the adjustments of the parameters. Driving times between locations are often more important for the results than the distribution of the volumes. The driving time to the region centre is not considered in calculating the region-region flows. Hub locations are chosen based on region-region flows, which might not be good for the consolidation in other directions.

4. What situations would you expect that the tool could be used in?
   - FN: To gain insight into a network. To generate a starting solution.
   - FvH: It could be used in the initial setup of a model. For the rest of the trajectory it is too roughly estimated.
   - TvD: Quickly generating a start solution.
   - TP: Perhaps a starting solution for a more advanced 'hub' tool.
5. Any additional comments?
   - TP: It is quite a nice model considering the time frame in which it was built.

E.1.4 Expert recommendations

One of the expert recommendations was to include a bubble chart of the volume distributions. A bubble chart is a graphical overlay on the map showing a filled circle at each node. The size of each circle is relative to the parcel volume at that node. Such a chart would help understand the operation of the tool.

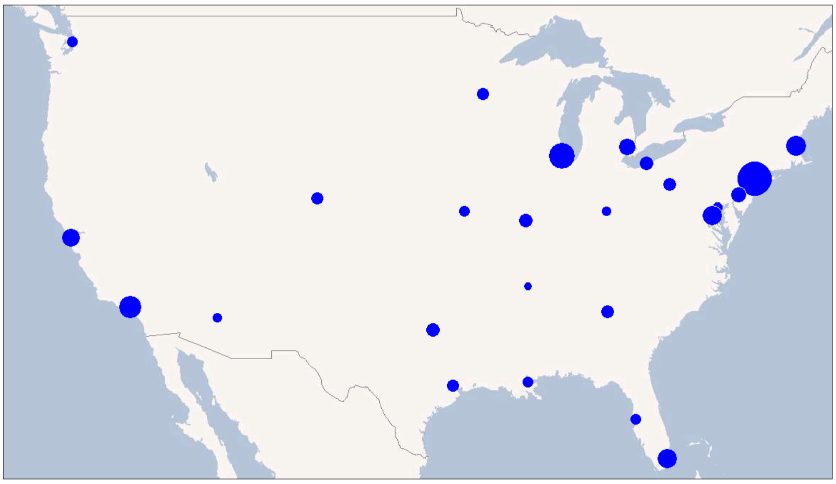


Figure E.1: Bubble chart showing the volume distributions of the CAB data

Besides adding bubble charts, the experts also made a few other recommendations. One of these, to add units to the parameters in the GUI, was implemented immediately. Other recommendations (like renaming the parameters in the GUI, adding a user guide, providing geographical insight into the transported flows, adding PUD to the tool, etc.) have not been implemented either due to the time frame of the research or because it was out of the scope of the tool (although inside the scope of the model design).

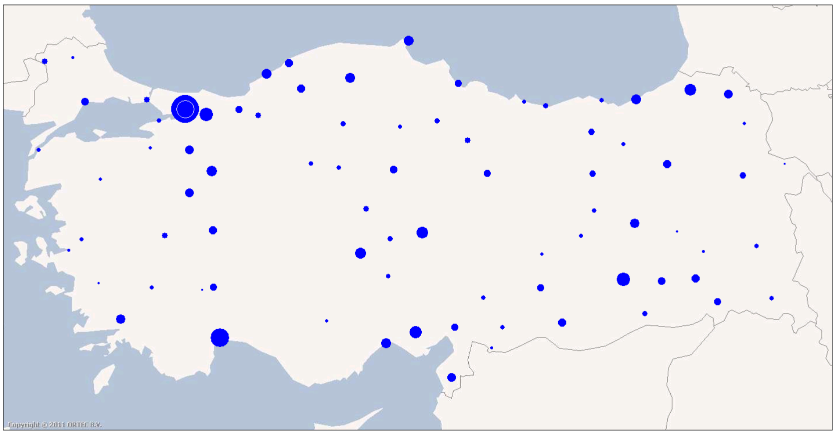


Figure E.2: Bubble chart showing the volume distributions of the Turkish Network data