Generating alternative solutions within the supply chain optimization model BOSS:

a mathematical programming approach

Master thesis Industrial Engineering and Management
by Roxanne Busschers

ORTEC

ADDING CREDIBILITY

This thesis is submitted for the Master Industrial Engineering and Management (Production and Logistics Management).
University of Twente, Enschede, The Netherlands.
School of Management and Governance.

Title:	Generating alternative solutions within the supply chain optimization model BOSS: a mathematical programming approach
Host organization:	ORTEC
Research period:	September 2011 - July 2012
AUTHOR	
Name:	R.A. Busschers
Student number:	s0128104
University:	University of Twente
Master degree program:	Industrial Engineering and Management
Track:	Production and Logistics Management
Contact:	r.a.busschers@alumnus.utwente.nl
SUPERVISOR COMMITTEE	
Supervisors University of Twente:	dr. ir. J.M.J. Schutten prof. dr. J.L. Hurink
Supervisor ORTEC:	ir. G.C. Brandt

UNIVERSITY OF TWENTE.

Preface

This thesis is part of my final project for the Master Industrial Engineering and Management, specialisation Production and Logistic Management, at the University of Twente in Enschede. In September 2011, I started my graduation project at ORTEC in Gouda, while at the same time working there on a part-time basis. At the time I started at ORTEC, I knew that my research was going to be about the supply chain optimization tool BOSS. However, at that time, the research objective was not defined yet. This placed me for the challenge of finding a subject for my research that could contribute to improving BOSS, was interesting for me to work on for at least six months, with the right scope and complexity for a graduation project. However, it was especially an opportunity for me to give direction, in collaboration with ORTEC and University Twente, to my research. I believe this has resulted in a successful project and I am pleased with the achieved results.

Of course, I could never have done this research on my own. I thank Gregor Brandt as my supervisor at ORTEC for his ideas and guidance. Also, I thank Frans van Helden for sharing his knowledge on the model and code in BOSS. Furthermore, my thanks go to Niels Uenk for his useful comments on my thesis and his encouragement. Finally, there are many more people at ORTEC who have helped me to complete this project and made it such a pleasant experience for me. Although, I cannot name them all here, I am very grateful to them.

From the university my project was supervised by Marco Schutten and Johann Hurink. Their guidance, comments, and thinking along throughout the project helped me a lot and I thank them for that.

Roxanne Busschers.

Management summary

ORTEC offers the software tool BOSS, which is used for supply chain optimization studies and decision support at a strategic and tactical level. To find the optimal configuration of a supply chain, BOSS uses mathematical programming. In mathematical programming, it is generally assumed that all data necessary to solve the model is accurately known at the moment of decision. However, in supply chain modeling, parameters like actual demand for products (right-hand side uncertainty) and prices of products (cost parameter uncertainty) are not precisely known when critical decisions have to be made. Furthermore, a mathematical model is generally a simplification of a real business problem. Such a model typically leaves out details that are difficult to express by formal expressions or that make the model hard to solve. Also, some optimization criteria are inherently subjective and difficult to quantify. This thesis describes research into improving the capabilities of BOSS to deal with data and model uncertainties.

Modeling to Generate Alternatives (MGA) has been proposed as a framework for dealing with complex problems for which there are important unmodeled issues (Chang et al., 1983). MGA techniques are designed to provide the decision maker a set of alternative solutions that are good with respect to the modeled objectives and different from each other in the decisions they make. Literature describes several methods to generate such alternatives. We propose to apply an approach based on the Hop, Skip, and Jump (HSJ) method by Brill et al. (1982), which we refer to as the Generalized HSJ (GHSJ) framework.

The GHSJ framework uses mathematical optimization with a different objective function than the original model uses. A constraint is added to this new model to ensure that the cost of an alternative solution does not exceed the optimal value by more than a pre-specified percentage. Furthermore, the constraints of the original model should hold. We propose four realizations of the GHSJ method that all use a different objective function, depending on the purpose of the method with respect to the obtained alternative solutions:

1. to obtain maximally different solutions, we apply the Standard HSJ method;
2. to obtain a large number of alternative solutions, we apply the Random HSJ method;
3. to obtain alternative solutions that perform better in case of cost parameter uncertainty, we apply the Cost uncertain HSJ method;
4. to obtain alternative solutions that are more robust against right-hand side uncertainty, we apply two versions of the Robust HSJ method.

Thus, the four approaches of the GHSJ method deal with both model uncertainty (method 1 and 2) and data uncertainty (method 3 and 4).

To test the proposed methods, we use two test cases based on studies performed for customers of ORTEC. The test results are promising and show that each method is able to obtain the type of alternative solutions that it aims for. Based on these results, we recommend ORTEC

to implement all four methods in BOSS. When doing a strategic network study for a customer with BOSS, we advise ORTEC to always use the Standard HSJ method. This method provides insight to the decision maker in the existence of solutions that are close to optimal, but with very different strategic decisions. We recommend to use the Random HSJ method, when cost may only increase from optimality by a small amount, for example 0.5%, and the Standard HSJ finds too few solutions. When cost uncertainty plays a role, even if there is only a slight presumption that actual cost may be different than assumed, we recommend to apply the Cost uncertain HSJ method. Initial results should point out whether there exist alternatives that outperform the initial solution. Also, if the Cost uncertain HSJ does not find good alternatives, the method still serves a purpose, since the decision maker gets more confidence in the proposed solution. We recommend applying the Robust HSJ method (version 1) if a customer wants to consider alternatives that allocate unused capacity differently over production facilities and distribution locations. Finally, when the customer indicates that he prefers capacity to be not fully utilized for all production or distribution locations, we recommend to apply the Robust HSJ method (version 2).

Contents

Chapter 1

Introduction

This report describes the graduation research into improving the capabilities of the supply chain optimization tool BOSS, to deal with uncertainties in input data and model formulations.

In strategic and tactical supply chain optimization studies, modelers are confronted with uncertainty of input data. This may influence the applicability and suitability of a proposed supply chain design. Furthermore, the decision maker almost always has some objectives that are not modeled. Therefore, he prefers to have several alternative solutions to choose from. These issues are also relevant for the supply chain optimization tool BOSS, which is developed at ORTEC.

Before we elaborate on this subject, we first introduce the company ORTEC in Section 1.1 and the supply chain optimization tool BOSS in Section 1.2. Section 1.3 provides a description of the relevant problems for this research. Section 1.4 discusses the research scope, objectives, and corresponding questions. Finally, Section 1.5 describes the organization of this thesis.

1.1 Company description

ORTEC is one of the largest providers of advanced planning and optimization software solutions and consultancy services. ORTEC was founded in 1981 and has grown to a company that employs over 700 employees in several offices in Europe, North America, Asia, and the Pacific Region with more than 1,550 customers worldwide. Industries in which ORTEC operates are among others: oil, gas and chemicals, aviation, trade transport and logistics, consumer packaged goods, professional and public services, and health care. The mission of ORTEC is as follows (*ORTEC company profile*, n.d.):

> *To support companies and public institutions in their strategic and operational decision making through the delivery of sophisticated planning and optimization software solutions, professional consulting, and mathematical modeling services.*

ORTEC's portfolio includes advanced software solutions to support and optimize operational planning for a wide range of business applications, such as fleet routing and dispatch, workforce scheduling, service planning, and vehicle load optimization. These standardized advanced planning systems are typically installed at customer site, and operated by customer staff. The focus of these systems is mainly tactical and operational.

Furthermore, ORTEC provides logistics consultancy and services dedicated specifically to individual customer needs and situation. The consultancy and services typically have a strategic or tactical focus, and are supported by dedicated logistics decision support systems, developed to meet individual customer needs. Also, ORTEC conducts network studies using software tools that are not fit to hand-over to customers.

This research concerns the consultancy department of ORTEC and, more specifically, the consultancy business unit Consulting and Information technology Services (CIS). This department has a large number of customers, mainly in logistics, for which it provides customized software solutions.

1.2 BOSS introduction

BOSS is a software tool that is used for supply chain optimization studies and decision support at a strategic and tactical level. A typical supply chain, as displayed in Figure 1.1, comprises suppliers, production sites, storage facilities, and customers. Suppliers are most upstream of the supply chain, providing raw materials to a production location. Each production location, also called plant, may have more than one supplier. Products produced at the plants will often be stored at one or two intermediary stages in the supply chain, namely warehouses and smaller distribution centers. Each warehouse may be supplied from more than one production location. Similarly, a distribution center can be supplied from more than one warehouse, although in practice it is most often only supplied from one warehouse. It is possible that one or more stages in the supply chain are located at the same site, such that there is no transport between these stages. For example, a warehouse may be located at a production site. Another deviation from the standard supply chain described here is when intermediary steps are left out. For example, the supply chain may not consist of warehouses or more extremely, products are transported directly from the production locations to the customers.

A supply chain should be managed in the most efficient way to minimize cost, delivery delays, and inventories, and to maximize profit and customer service levels. To this end, supply chain management involves several strategic and tactical decisions (Tsiakis et al., 2001):

- Location decisions consider the number, size, and physical location of production plants, warehouses, and distribution centers.
- Production decisions consider the products to be produced at each production site and also the allocation of suppliers to plants and of plants to warehouses.
- Transportation decisions consider the allocation of plants or warehouses to distribution centers and of distribution centers to customers.

BOSS is able to cover the entire supply chain from obtaining the raw materials to the delivery at the customers. Based on data on, among others, costs, capacities, and demand, BOSS calculates the optimal supply chain configuration within the allowed solution space. To find the optimal design of the supply chain, BOSS uses mathematical programming. Chapter 2 presents a more extensive description of the BOSS model.

The approach ORTEC takes with BOSS is that it developed a software tool that can be operated at a customer site, with a standardized ‘core’ of basic functionalities. The design allows tailoring and extension to meet customer requirements. Also, in its general form, BOSS can be used by consultants of ORTEC in studies performed for customers.

1.3 Problem introduction

In mathematical optimization, it is generally assumed that all data necessary to solve the model is accurately known at the moment of decision. However, in many real world optimization problems, data uncertainty is present. In supply chain optimization, the actual demand for products, financial returns, prices of products, material requirements, machine reliability, and

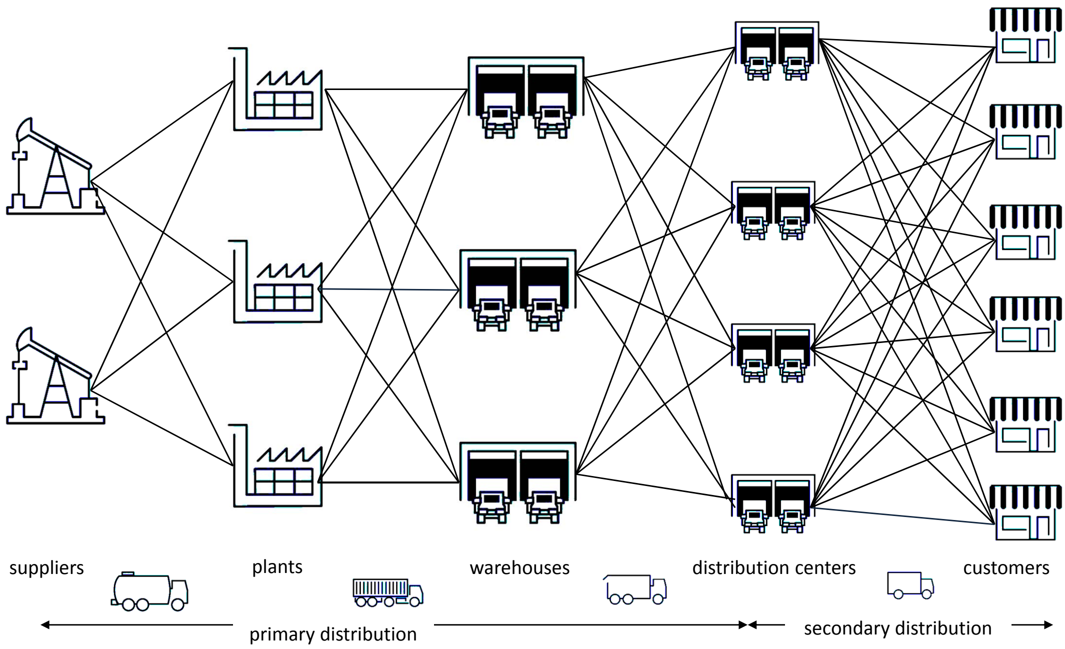


Figure 1.1: A typical supply chain network.

other resources are not precisely known when critical decisions have to be made. The moment a company makes important strategic decisions for the upcoming months, or even years, a lot of the input data is still uncertain. In BOSS, a linear programming problem is solved for which all parameters are set at their most likely values. Nevertheless, the model is likely to violate the constraints with the actual data or the model might be far from optimal for the realized data. For example, when actual demand is higher than expected, it might be infeasible to fulfill all demand in time. The obtained solution can be far from optimal when, for example, realized production cost for a location are underestimated.

In many practical applications, constraint violation may influence the usability of a solution as shown in an extensive case study on several linear optimization problems by Ben-Tal and Nemirovski (2000). They conclude that:

> *In real world applications of linear programming one cannot ignore the possibility that a small uncertainty in the data can make the usual optimal solution completely meaningless from a practical viewpoint.*

Since actual input data cannot be used, the goal is to find a solution which is less sensitive to a deviation from the expected input data. When parameters are equal to their expected values, the objective value should not loose too much in optimality. Obtaining such a solution would be relevant to the BOSS model, since many decision-makers are willing to trade off some optimality for a solution that is more robust against data uncertainty.

Another critical assumption made in mathematical optimization also concerns uncertainty, namely model uncertainty. Generally, a mathematical model is assumed to accurately describe the real life situation. However, the mathematical model is almost always a simplification of the real business problem. Such a model may deliberately leave out details that are difficult to express by mathematical expressions or that make the model hard to solve. More importantly, some optimization criteria are inherently subjective and difficult to quantify. For example, a decision-maker might from experiences have some idea in mind about a preferred schedule. In practice, a business manager prefers a schedule with little 'chaos' in it, i.e., a simple, orderly schedule with as little as possible deviations from some basic schedule. This means for example, that customers are preferably supplied every time from one single depot instead of from changing depots. These kind of preferences might not be easily modeled.

Therefore, from a decision-making perspective, it is preferable to have several alternative solutions. These solutions should differ significantly from each other regarding the decisions made. For example, they might have different facility locations or a different allocation of customers to depots. However, their objective value should not deviate too much from the optimal solution. The decision-maker should decide which solution best matches the actual situation and fits both the modeled and unmodeled objectives.

1.4 Research design

This section gives a formal description of the research into the two uncertainty related problems discussed in Section 1.3. Section 1.4.1 specifies the scope of this research. Section 1.4.2 identifies the research objectives. To meet the research objectives, a number of sub goals are identified and several research questions need to be answered. Section 1.4.3 discusses these goals and research questions.

1.4.1 Research scope

The research in this thesis concentrates on how to deal with uncertainty in the BOSS model. Since the BOSS model is formulated as a mixed integer linear programming (MILP) problem, this research specifically focuses on MILP problems.

As BOSS is a consulting tool, it is applied in many different settings and with different purposes. For this research, we focus on strategic level decisions in supply chain optimization. These strategic decisions concern, among others, the locations that should be opened and the allocation of customers to distribution centers. Therefore, we refer to this type of problem as the *location-allocation* problem. Chapter 2 provides an exact definition of the model for this study.

Section 1.3 discusses two sources of uncertainty in mathematical programming that are relevant for supply chain planning, namely:

- data uncertainty, i.e., uncertainty about, among others, actual demand, capacities, and costs;
- model uncertainty, i.e., uncertainty about whether the model accurately describes the realistic situation.

For this study, our goal is to investigate how we can deal with both types of uncertainty. However, the primary focus of this research is on generating alternative solutions to account for model uncertainty. The main reason for this choice is that business wise it is more relevant to generate alternatives than to obtain robust solutions, since in practice there is a lot of demand for it. This is among others caused by the fact that, for robust optimization a lot of in depth knowledge on the business case and also on the uncertain data is required. However, this information is often not readily available. For generating alternatives, no extra information is required, since the sole purpose of generating the alternatives is to account for unknown information. Furthermore, a decision maker is usually not looking for the safest solution; mostly he is interested in solutions that are close to optimal, but differ on the critical decisions made. He can then choose the solution that best fits the modeled and unmodeled objectives. Additionally, by implementing a method to generate alternative solutions, we might obtain solutions that are besides significantly different from each other, also more robust against data uncertainty. More concretely, we can explicitly incorporate the objective of robustness in the framework for generating alternative solutions. Finally, from the literature overview in Chapter 3, it follows that methods to deal with model uncertainty are much better suited for implementation in BOSS for the model instances we want to deal with.

1.4.2 Research objectives

We identify the need to gain insight into methods to deal with uncertainty in the location-allocation problem in BOSS and ORTEC's wish to improve the ability of BOSS to handle these uncertainties. Therefore, we formulate the following research objective:

> *The objective of this research is to improve the capabilities of BOSS to deal with both data and model uncertainties.*

In order to make this objective more concrete, we formulate for both sources of uncertainty a separate objective. To account for data uncertainty, solutions should be obtained that are more robust against small deviations in input data. To deal with model uncertainty, alternative solutions should be generated, such that the decision maker can decide which solution best fits the modeled and unmodeled objectives. We formulate the following two sub-objectives:

SO 1 *The objective of this research with respect to data uncertainty is to create insight in available methods to obtain solutions to the location-allocation problem in BOSS that are more robust against data uncertainty.*

SO 2 *The objective of this research with respect to model uncertainty is to develop a method to obtain alternative solutions to the location-allocation problem in BOSS.*

1.4.3 Research questions

To meet the research objectives as formulated in Section 1.4.2, this research consists of several goals that all serve their own purpose and together work towards the research objectives. We now discuss the goals and the research questions arising from these goals. Figure 1.2 displays the relation between the questions.

The first goal of this research is to provide a formal description of BOSS. To this end, we formulate a mixed integer linear programming (MILP) model for the location-allocation problem in BOSS. This goal leads to the following question:

RQ 1 *What is the (mixed integer) linear programming model which describes the strategic location-allocation problem that can be modeled in BOSS?*

The second goal of this research is to identify relevant methods in literature to obtain solutions to linear programming problems that are less sensitive to uncertainty in the input data. We refer to these kind of solutions as robust solutions. This goal leads to the following two questions:

RQ 2 *What approaches exist in literature to obtain solutions to linear programming problems that are more robust against uncertainties in the input data than the solution that is optimal for specific parameter values?*

RQ 3 *What identified methods to obtain solutions that are more robust against data uncertainty are best suited for implementation in BOSS?*

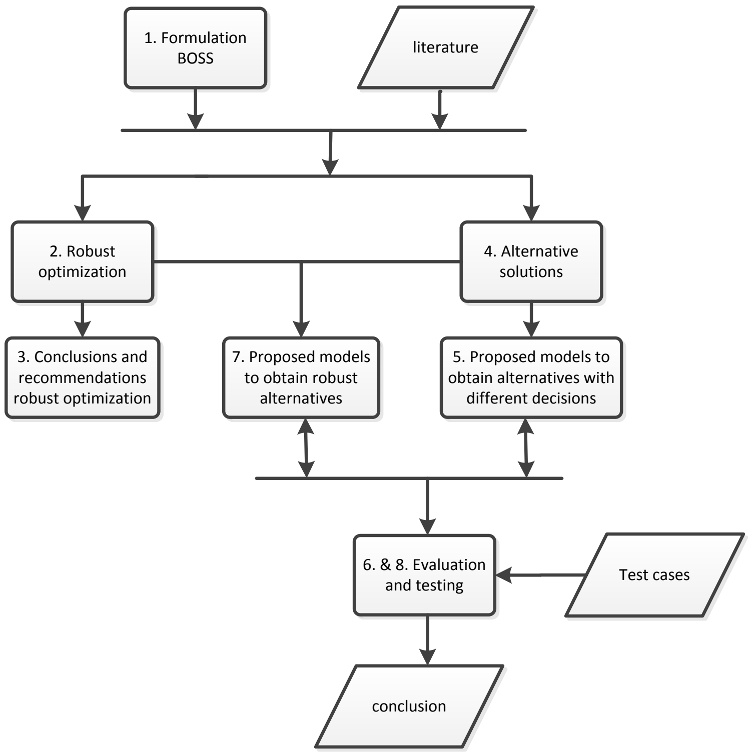


Figure 1.2: Relation between research questions.

The third goal of this research is to identify relevant methods in literature to obtain alternative solutions and to assess the applicability of these methods to the location-allocation problem in BOSS. The alternative solutions should perform well with respect to the modeled objective(s) and they should differ from each other with respect to critical decisions. To this end, we propose a framework to generate alternative solutions to a linear programming model and test realizations of this framework for applicability in BOSS with two test cases. The research questions corresponding to this research goal are:

RQ 4 *What approaches exist in literature to obtain alternative solutions to a linear programming problem, that make different critical decisions from the optimal solution?*

RQ 5 *How can we define a framework to generate alternative solutions to the location-allocation problem in BOSS, that are close to optimal with respect to the original objective, but make different decisions?*

RQ 6 *To what extent are the proposed models able to generate alternative solutions that differ in the decisions taken?*

Finally, the fourth goal of this research is to identify ways in which the proposed framework to deal with model uncertainty, can also be used to account for data uncertainty. We want to asses the applicability of such methods to the location-allocation problem in BOSS. The following research questions correspond to this goal:

RQ 7 *How can we use the defined framework to obtain solutions that are more robust against uncertainties in input data?*

RQ 8 *To what extent are the proposed models able to generate one or more alternative solutions that are more robust against data uncertainties?*

1.5 Organization of the thesis

This section describes the organization of this thesis and gives a brief summary of the following chapters. Chapter 2 describes in detail the location-allocation problem that can be modeled in BOSS and gives the mathematical formulation for this model. By comprehending the model formulation, we understand how the initial solution to the problem is obtained. Furthermore, it enables us to specify with respect to which decisions, alternative solutions should differ from the initial solution and each other. Also, this chapter discusses the data input that is considered to be uncertain in BOSS.

Chapter 3 gives an overview of relevant literature. This literature study helps us to place our research into perspective and defines a direction for the remainder of the thesis. The literature study reviews methods to account for either data uncertainty (Section 3.1) or model uncertainty (Section 3.2). Based on this study, we decide to focus on methods to generate alternative solutions.

Based on the discussed literature, Chapter 4 describes a general framework to generate alternative solutions. We apply this framework to the location-allocation problem in BOSS. Subsequently, this chapter proposes four realizations of this framework to generate alternatives that either make different decisions or are better able to deal with data uncertainty than the initial solution.

Chapter 5 describes two test cases from practice that are used to test the proposed methods. We apply all methods to these test cases. This chapter also discusses the obtained results in detail and describes further research that is done on the sensitivity of the methods. Furthermore, Chapter 5 discusses some modifications to improve obtained results and tests these adapted models.

Finally, Chapter 6 summarizes the answers to the research questions. Also, this chapter contains a section that makes concrete recommendations to ORTEC on the application of the proposed methods in studies ORTEC performs for customers with BOSS. Based on the assumptions made in this research and the knowledge gained on methods to generate meaningful alternative solutions to the location-allocation problem in BOSS, we conclude with some suggestions for further research.

Chapter 2

BOSS model description

BOSS is a decision support tool for supply chain optimization at a strategic and tactical level. The tool can be applied in many different settings and for different time horizons and types of decisions. This chapter provides a formal description of BOSS. Section 2.1 discusses the type of problems solved with BOSS on which this research focuses. Section 2.2 provides a description of the corresponding mathematical formulation of the BOSS model. Section 2.3 discusses data uncertainty in BOSS. Finally, Section 2.4 discusses on which decisions we want alternative solutions to differ from the initial solution and each other.

2.1 Introduction to BOSS model

BOSS defines locations as the main entities and theoretically between any two locations transportation of goods may occur. Furthermore, a location can have one or more of the following roles: production, storage, and demand. One physical location can be modeled as a supply chain itself as well, i.e., a location can have more than one production facility, and output of one production facility may be input to another. Figure 2.1 displays the typical set up of a supply chain in BOSS. Products can refer both to end products sold to customers, as well as to intermediate products used in the production process. Products can be defined as supply at a location if the modeler wishes to no further specify the upstream supply chain of this product. In the example, this is the case for $P1$, which is modeled to be available at a certain price at location $S1$ and the modeler takes no further interest in where the product originates from. Commodities are used in the production process; they are available at the start of the period at a location and, if not used by the end of the period, are no longer available. Examples of commodities that can limit the total production are available time or maximum CO_2 emission.

The scope of the research is the application of BOSS to strategic level studies, with a time horizon of 1 to 5 years. Even though BOSS is able to handle multiple periods, we consider only one time period models, where the data is aggregated to this level. The main study questions for this time horizon are:

- At which locations should new production facilities/depots be opened?
- Which existing production locations/depots should be closed?
- In which production locations/depots should be invested to increase the capacity, and how much should be invested?

Since BOSS is a widely deployable tool with many options for lower level studies, we might easily end up using an unnecessarily complicated model with too many details. Examples of

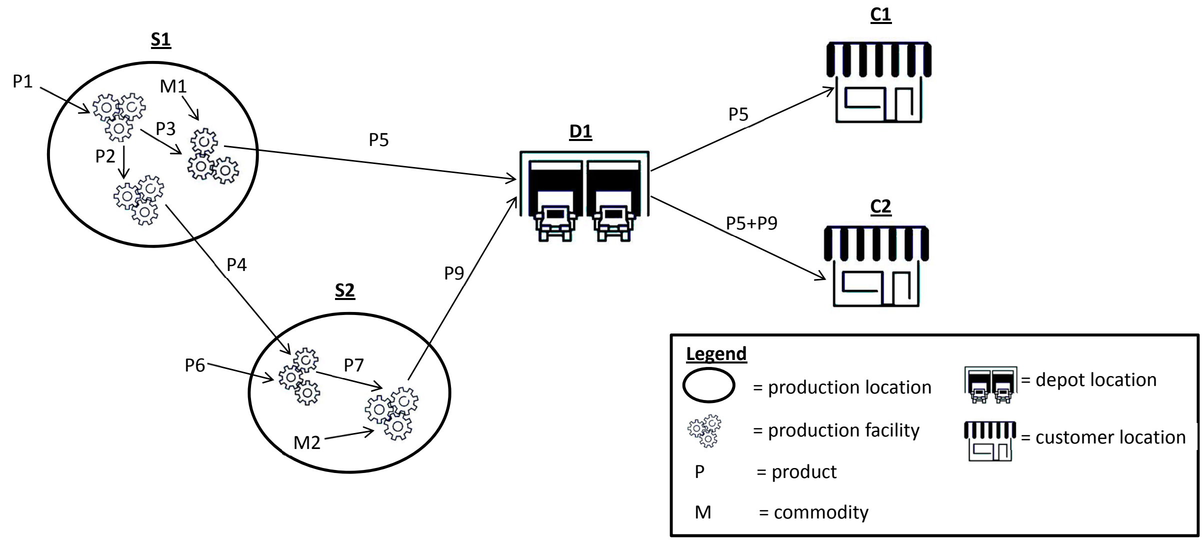


Figure 2.1: Typical supply chain as modeled in BOSS.

issues that are not relevant for our study are:

- stock levels and inventory cost;
- different production modes at each production facility;
- different transport modalities, with different transportation time and cost between any two locations;
- exchange/swap deals with competitors;
- non-monotonous tranched pricing for products and commodities;
- routing and timing of deliveries;
- workforce scheduling.

Although these issues are not taken into account in our study, all except for the last two can be modeled in BOSS. Section 2.2 discusses the mathematical formulation corresponding to the BOSS model described above. This model formulation is used throughout this thesis.

2.2 Mathematical formulation

This section describes the mathematical formulation of the supply chain model outlined in Section 2.1. Each subsection describes for an element, or part, of the supply chain how it is modeled in BOSS. Appendix A gives a summary of this mathematical model. Note that we differentiate between parameters and decision variables by beginning the name of a decision variable with a capital letter and a parameter name not.

2.2.1 Supply chain

A supply chain consists of locations and flows of products between them; the potential flows are represented by arcs. To model the general network we define:

- $\mathcal{L}$: set of all physical locations in the network;

- $\mathcal{P}$: set of products;
- $Flow(p, l_1, l_2)$: decision variable giving the number of units of product p transported from location l_1 to location l_2;
- $LocOpen(l)$: decision variable indicating whether or not a location is in use:

$$LocOpen(l) = \begin{cases} 1 & \text{if location } l \text{ is in use;} \\ 0 & \text{otherwise.} \end{cases}$$

The maximum throughput of a location is determined by the distribution capacity of the location. This distribution capacity is given for each location:

- $flowcap(p, l)$: maximum throughput of product p at location l.

Furthermore, for a location that is used for either production or distribution, fixed cost are paid. To model this, we ensure by a constraint that the decision variable $LocOpen(l)$ is set to 1, when a location is used. However, for a customer location, $LocOpen(l)$ does not have to be set to 1, since no fixed cost are associated with such a location. Thus, products (either intermediates or end products) can only originate from a location for which $LocOpen(l)$ is 1. On the other hand, products can be received by locations that are not 'opened', i.e., the decision variable $LocOpen(l)$ may equal zero for such a location. Also, we want to ensure that throughput at a location is never higher than the throughput capacity. To model both situations, we introduce the following restriction:

$$\sum_{l_1 \in \mathcal{L} | l_1 \neq l} Flow(p, l, l_1) \leq LocOpen(l) \cdot flowcap(p, l) \qquad \forall p \in \mathcal{P}, l \in \mathcal{L}. \tag{2.1}$$

However, since the distribution capacity of a location can be increased by making an investment, we replace constraint (2.1) by two other constraints, see Section 2.2.5.

2.2.2 Production process

For a location to be able to manufacture products, it needs to have production facilities. Therefore, we define:

- $\mathcal{F}$: set of all production facilities;
- $\mathcal{F}^l \subseteq \mathcal{F}$: set of all production facilities located at physical location l, i.e., $\mathcal{F}^l = \emptyset$ if no production can take place at location l, for example, if l is a customer location.

Note the difference between production facilities and distribution facilities; whereas the distribution capacity is defined by the model for the entire location, production facilities are modeled separately at a location and one location can have multiple production facilities.

The input of the production process, called ingredients, consists of products and commodities. Commodities are available at a location for a certain time period. Examples of commodities are electricity and CO_2. Thus, we define:

- $\mathcal{M}$: set of commodities;
- $\mathcal{G}$: set of ingredients for the production process, i.e., $\mathcal{G} = \mathcal{M} \cup \mathcal{P}$.

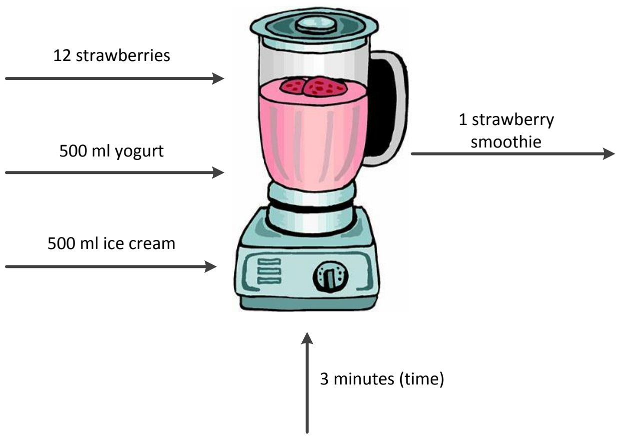


Figure 2.2: Example production process strawberry smoothie.

Each production facility produces according to a specific recipe. For each ingredient and a specific facility, a recipe indicates the number of units of that ingredient used or produced in the production process of that production facility; a negative number denotes the use of an ingredient and a positive number denotes the production of the ingredient. We assume that commodities can only be used in the production process and cannot be produced, whereas products can be both used and produced. A recipe of a production facility gives the ratios of the ingredients in the production process of that production facility. Thus, we introduce the following parameter:

- $recipe_f\,(g)$: number of units of ingredient g used or produced at production facility f.

We clarify this concept by an example of producing a strawberry smoothie. The production facility is a blender and the ingredients are: 12 strawberries, 500 ml of yogurt, and 500 ml of vanilla ice cream. Furthermore, we need to blend for 3 minutes. If time is a constraining factor, we should explicitly model it as a commodity. Figure 2.2 summarizes this process. Note that each ingredient can have its own unit of measurement. For the example, we have the following recipe:

1. $recipe_{blender}\,(strawberries) = -12;$
2. $recipe_{blender}\,(yogurt) = -500;$
3. $recipe_{blender}\,(ice\,cream) = -500;$
4. $recipe_{blender}\,(time) = -3;$
5. $recipe_{blender}\,(smoothies) = 1.$

To measure the production level of facility f, we introduce a decision variable $Production\,(f)$, indicating the number of times $recipe_f$ is carried out. The unit of measurement is called production units, such that if $Production\,(f) = 1$, $recipe_f$ is carried out once and the number of production units equals 1. The actual number of units of an ingredient used or produced in the production process can be derived by multiplying the production level by the corresponding recipe. Therefore, we define:

- $Production\,(f)$: production level at facility f, we measure this variable by production units;

- $ProdIn\,(p, f)$: number of units of product p used in the production process of facility f;

$$ProdIn\,(p, f) \;=\; -1 \cdot Production\,(f) \cdot recipe_f\,(p) \quad \forall\, p \in \mathcal{P}, f \in \mathcal{F} |\, recipe_f\,(p) < 0; \tag{2.2}$$

- $ProdOut\,(p, f)$: number of units of product p produced at facility f;

$$ProdOut\,(p, f) \;=\; Production\,(f) \cdot recipe_f\,(p) \quad \forall\, p \in \mathcal{P},\; f \in \mathcal{F} |\, recipe_f\,(p) \geq 0; \tag{2.3}$$

- $CommUse\,(m, f)$: number of units of commodity m used in the production process of production facility f;

$$CommUse\,(m, f) \;=\; -1 \cdot Production\,(f) \cdot recipe_f\,(m) \qquad \forall\, m \in \mathcal{M},\; f \in \mathcal{F}. \tag{2.4}$$

Note that $ProdIn\,(p, f)$, $ProdOut\,(p, f)$, and $CommUse\,(m, f)$ are auxiliary variables, introduced to increase readability. If we return to the example of Figure 2.2 and consider the case that we make 3 strawberry smoothies, i.e., $ProdOut\,(smoothies) = 3$, we get $Production\,(blender) = 3$ by Equation (2.3). Table 2.1 displays the values of $ProdIn\,(p, f)$ and $ProdOut\,(p, f)$ for the products.

	strawberries	*yogurt*	*ice cream*	*smoothies*
$ProdIn\,(p, blender)$	36	1500	1500	
$ProdOut\,(p, blender)$				3

Table 2.1: Input and output of products, for the example of Figure 2.2 and a production level of 3 strawberry smoothies.

2.2.3 Production capacity

The maximum production level of a production facility is determined by the production capacity:

- $prodcap\,(f)$: maximum production capacity of production facility f.

The following constraint models that the production level at a facility is restricted by the capacity of that facility:

$$Production\,(f) \leq prodcap\,(f) \qquad\qquad \forall\, f \in \mathcal{F}. \tag{2.5}$$

However, the production capacity of a facility can be increased by making an investment; Section 2.2.5 discusses the modification of this constraint, such that it also covers investments.

Next we consider products for which the upstream supply chain is no further specified, for example because a product is purchased from an external supplier. Such products are supplied to a location and are not produced in the scope of the supply chain. Therefore, these products are modeled as if they are available at a location against a certain price. The transport of a supplied product to the location is not modeled, since transportation costs are included in the supply costs. So, we define:

- $Supply\,(p, l)$: decision variable indicating the number of units of product p supplied to location l.

Thus, there are two option for products to enter the supply chain, either as a result of the production process in the supply chain, or as supply from outside the relevant supply chain.

The production level at a facility is restricted by the capacity of a production facility, as expressed in restriction (2.5). However, for each product at each location a maximum supply is given and this can also restrict the total production capacity of a location. To model this, we introduce the following parameter:

- $maxsupply\,(p,l)$: maximum supply of product p at location l.

The following constraint ensures that the total supply at a location is no bigger than the maximum available supply for each product:

$$Supply\,(p,l) \;\le\; maxsupply\,(p,l) \qquad\qquad \forall\, p \in \mathcal{P},\; l \in \mathcal{L}. \tag{2.6}$$

Finally, the actual production level may also be restricted due to the available units of a commodity:

- $commavail\,(m,l)$: available number of units of commodity m at location l.

The following constraint models the restriction on the use of commodities:

$$\sum_{f\in\mathcal{F}^l} CommUse\,(m,f) \le commavail\,(m,l) \qquad\qquad \forall\, m \in \mathcal{M},\; l \in \mathcal{L}. \tag{2.7}$$

2.2.4 Location balance

Ultimately, the purpose of the supply chain is to make money by selling products to customers. Therefore, we introduce the parameter:

- $demand\,(p,l)$: number of units demand for product p by location l.

To maximizes profit and customer satisfaction, all demand by customers has to be fulfilled. Furthermore, the flow from and to a location, together with the supply, production process, and demand should be balanced, such that a product cannot suddenly appear at a location without it being first produced, supplied, or transported to that location. Therefore, the following balance equation should hold:

$$\begin{aligned}
&\sum_{l_1\in\mathcal{L}|l_1\neq l} Flow\,(p,l_1,l) + \sum_{f\in\mathcal{F}^l} ProdOut\,(p,f) + Supply\,(p,l) = \\
&\sum_{l_2\in\mathcal{L}|l_2\neq l} Flow\,(p,l,l_2) + \sum_{f\in\mathcal{F}^l} ProdIn\,(p,f) + demand\,(p,l) \qquad \forall\, p \in \mathcal{P},\; l \in \mathcal{L}.
\end{aligned} \tag{2.8}$$

The above equation holds for all types of locations. For example for a distribution location, $ProdOut\,(p,f)$, $ProdIn\,(p,f)$, and $demand\,(p,l)$ equal zero and the remaining equation states that for each product total flow to a location, including flow from outside the scope of the supply chain ($Supply\,(p,l)$), equates to the total flow out of the location. Similarly, for a customer location the above equation states that the flow into the location should equal the demand, i.e., all demand has to be fulfilled.

2.2.5 Investments

Only in very rare cases, a study performed with BOSS will be a greenfield study, i.e., a study in which no previous facilities exist, such that the entire supply chain has to be configured. More likely, a study is performed on an existing supply chain, with some fixed locations for production and distribution. Nevertheless, it might be possible to invest in the supply chain to change the current configuration. For example, the throughput capacity of a distribution location can be increased. Even though the choice of opening new locations is also an investment issue, we do not formally define it as such. We define two types of investments: investments in the distribution capacity of a location and investments in the production capacity of a production facility. Furthermore, investments can be divided into positive and negative investments; capacity is increased by a positive investment and decreased by a negative investment. Evidently, a positive investment costs money and a negative investment brings in money.

Since it is not realistic to assume that we can linearly increase capacity by investing a certain amount, the management has to specify a set of potential investments. For example, they might be aware of the opportunity to buy a new machine at a certain price to double the production capacity of a certain production facility. Also, an investment can be defined as closing a distribution location to make a certain amount of money. We have:

- $\mathcal{I}$: set of potential investments. Each investment is related to either a location in its entirety or to a single facility;
- $\mathcal{I}^{fl(l)} \subseteq \mathcal{I}$: set of potential investments that increase or decrease the distribution capacity of location l;
- $\mathcal{I}^{pr(f)} \subseteq \mathcal{I}$: set of potential investments that increase or decrease the production capacity of facility f.

Furthermore, we define:

- $InvDone\,(i)$: decision variable indicating whether or not an investment i is made:

$$InvDone\,(i) = \begin{cases} 1 & \text{if investment } i \text{ is made;} \\ 0 & \text{otherwise.} \end{cases}$$

It is possible to define several different investment options for one production facility; the management may for example have the option to choose between two machines with different cost and capacity. However, investments at a facility are mutually exclusive. Similarly, even though at each location several investment options can be defined to increase or decrease the distribution capacity, only one investment can be chosen. We model these two restrictions in the following way:

$$\sum_{i \in \mathcal{I}^{fl(l)}} InvDone\,(i) \leq 1 \qquad \forall\, l \in \mathcal{L}. \tag{2.9}$$

$$\sum_{i \in \mathcal{I}^{pr(f)}} InvDone\,(i) \leq 1 \qquad \forall\, f \in \mathcal{F}. \tag{2.10}$$

If the production capacity of a location can be changed by an investment, constraint (2.5) has to be adapted. We introduce a new parameter giving the production capacity corresponding to an investment:

- $investprodcap\,(i, f)$: maximum production capacity of facility f if investment i is made.

If no investment is made, the production capacity equals the original production capacity (*prodcap*). If an investment is made, the actual production capacity equals the investment production capacity (*investprodcap*) and the original capacity is no longer valid. Therefore, we determine the actual available production capacity as follows:

- $AvailProdCap\,(f) = prodcap\,(f) \cdot \left(1 - \sum_{i \in \mathcal{I}^{pr(f)}} InvDone\,(i)\right)$

$$+ \sum_{i \in \mathcal{I}^{pr(f)}} investprodcap\,(i, f) \cdot InvDone\,(i) \qquad \forall\, f \in \mathcal{F}. \tag{2.11}$$

Thus, we replace constraint (2.5) by:

$$Production\,(f) \leq AvailProdCap\,(f) \qquad \forall\, f \in \mathcal{F}. \tag{2.12}$$

For the distribution capacity, a similar situation holds as for the production capacity. Therefore, we introduce the following parameter:

- $investflowcap\,(i, p, l)$: maximum throughput capacity of product p at location l if investment i is made.

We determine the actual available distribution capacity as follows:

- $AvailFlowCap\,(p, l) = flowcap\,(p, l) \cdot \left(1 - \sum_{i \in \mathcal{I}^{fl(l)}} InvDone\,(i)\right)$

$$+ \sum_{i \in \mathcal{I}^{fl(l)}} investflowcap\,(i, p, l) \cdot InvDone\,(i). \qquad \forall\, p \in \mathcal{P},\, l \in \mathcal{L}. \tag{2.13}$$

The adapted restriction on the distribution capacity (2.1) is given by:

$$\sum_{l_1 \in \mathcal{L} | l_1 \neq l} Flow\,(p, l, l_1) \leq AvailFlowCap\,(p, l) \qquad \forall\, p \in \mathcal{P},\, l \in \mathcal{L}. \tag{2.14}$$

However, another purpose of constraint (2.1) was to ensure that the decision variable $LocOpen\,(l)$ is set to 1 if a location is used for either production or distribution. In this constraint, $flowcap\,(p, l)$ had the function of a so called 'bigM' parameter. As discussed by Camm et al. (1990), the choice of the constant 'bigM' can have serious computational implications. By making this parameter arbitrarily large, the feasible region of the LP relaxation is unnecessarily expanded. Therefore, 'bigM' should always be chosen as small as possible. However, if the distribution capacity increases due to the decision to make an investment, constrain (2.1) does not suffice anymore. We introduce a new 'bigM'-type parameter, which equals the maximum distribution capacity at a location:

- $bigMFlow\,(p, l) = \max\left(\max_{i \in \mathcal{I}^{f}l(l)}\left(investflowcap\,(i, p, l)\right), flowcap\,(p, l)\right).$

Now, we ensure that the decision variable $LocOpen\,(l)$ is set to 1 if a location is used, by the following constraint:

$$\sum_{l_1 \in \mathcal{L} | l_1 \neq l} Flow\,(p, l, l_1) \leq LocOpen\,(l) \cdot bigMFlow\,(p, l) \qquad \forall\, p \in \mathcal{P},\, l \in \mathcal{L}. \tag{2.15}$$

2.2.6 Objective function

The objective of BOSS is to minimize the total costs of the supply chain design. Therefore we define the following decision variable:

- Cost : total cost of supply chain design:

$$\begin{aligned} Cost = & \ TransportCost + LocationCost + ProductionCost \\ & + SupplyCost + InvestmentCost, \end{aligned} \tag{2.16}$$

where

$$TransportCost = \sum_{\substack{p \in \mathcal{P} \\ l_1, l_2 \in \mathcal{L} | l_1 \neq l_2}} Flow\,(p, l_1, l_2) \cdot vartransportcost\,(p, l_1, l_2); \tag{2.17}$$

$$LocationCost = \sum_{l \in \mathcal{L}} LocOpen\,(l) \cdot fixedlocationcost\,(l); \tag{2.18}$$

$$\begin{aligned} ProductionCost = & \sum_{l \in \mathcal{L}} \sum_{\substack{m \in \mathcal{M} \\ f \in \mathcal{F}^l}} CommUse\,(m, f) \cdot varcommoditycost\,(m, l) \\ & + \sum_{f \in \mathcal{F}} Production\,(f) \cdot varproductioncost\,(f); \end{aligned} \tag{2.19}$$

$$SupplyCost = \sum_{\substack{p \in \mathcal{P} \\ l \in \mathcal{L}}} Supply\,(p, l) \cdot varsupplycost\,(p, l); \tag{2.20}$$

$$InvestmentCost = \sum_{i \in \mathcal{I}} InvDone\,(i) \cdot fixedinvestmentcost\,(i). \tag{2.21}$$

The meaning of the cost parameters is as follows:

- $vartransportcost\,(p, l_1, l_2)$: variable cost of transporting one unit of product p from location l_1 to location l_2;
- $fixedlocationcost\,(l)$: fixed cost of using location l;
- $varcommoditycost\,(m, l)$: variable cost of one unit of commodity m at location l;
- $varproductioncost\,(f)$: variable cost of one production unit at production facility f;
- $varsupplycost\,(p, l)$: variable cost of one unit of product p supplied to location l. This amount includes transportation cost and purchase price;
- $fixedinvestmentcost\,(i)$: fixed cost of making investment i.

The objective of BOSS is simply:

$$\min\ Cost. \tag{2.22}$$

Note that while there are fixed costs associated with using a location, these cost are not defined for facilities. We assume that a facility that is currently utilized remains to be used in the future, unless the entire location is closed and in that case the facility closing benefits are part of the $fixedlocationcost\,(l)$ parameter. Conversely, the option to buy a new facility at a location is modeled as an investment.

Furthermore, since the model will most likely not be used for a period equal to the lifetime of investments in facilities and locations, $fixedlocationcost\,(l)$ and $fixedinvestmentcost\,(i)$ should be set equal to the depreciation cost over the planning horizon.

2.3 Data uncertainty in BOSS

For supply chain studies, it is very relevant to obtain solutions that are robust against data uncertainty in the demand parameters. When making decisions at a strategic level, actual orders of customers are most likely not be received yet. Therefore, customer demand is often estimated based on historical data. The accuracy of these estimates strongly depends on business characteristics and resulting variance in demand. For example, the demand for coffee can be estimated reasonably well, while estimates for road salt most likely differ significantly, especially on the long term, from actual demand.

Furthermore, cost components are inherently uncertain. For example, products that are bought on the spot market can vary considerably in price. For investments, the cause of uncertainty is not only the variability of prices, but rather that it is very difficult to estimate actual cost beforehand. This also holds for the fixed cost of opening a location. In conclusion, we consider uncertainty in the following input parameters:

- $demand\,(p, l)$;
- $vartransportcost\,(p, l_1, l_2)$;
- $fixedlocationcost\,(l)$;
- $varcommoditycost\,(m, l)$;
- $varproductioncost\,(l, f)$;
- $varsupplycost\,(p, l)$;
- $fixedinvestmentcost\,(i)$.

2.4 Alternative solutions in BOSS

As discussed in Chapter 1, our goal is to obtain multiple near-optimal solutions for the mathematical model, so that the decision maker can examine them, and choose the solution that best fits his preferences with respect to both the modeled objective (lowest cost) and unmodeled objectives. Now that we have described the mathematical model of BOSS, we can explicitly define the aspects in which we like the alternative solutions to differ from each other.

The decisions made by BOSS that are most influential to the design of the supply chain are: which locations are opened and which investments, if any, are made to increase or decrease the capacity of production facilities and distribution locations. The alternative solutions should therefore differ from each other with respect to the following two binary decision variables:

- $LocOpen\,(l)$;
- $InvDone\,(i)$.

We consider two solutions to be different if and only if, they differ by at least one of these binary variables. Another reason for choosing these binary variables and no continuous variables, is that there might exist an infinite number of solutions that differ by at least one continuous variable and in that case it is less obvious how to define when two solutions are significantly different (Danna et al., 2007).

Chapter 3

Literature overview

In this thesis, we have indicated two relevant and related research problems. First, a key difficulty in the strategic perspective of our location-allocation problem is addressing data uncertainty. Second, we like to account for model uncertainties in BOSS by offering the decision maker a choice between multiple near-optimal solutions that are significantly different in decision space.

This chapter gives an overview of the relevant literature for dealing in a mathematical model with data uncertainty (Section 3.1) and model uncertainty (Section 3.2).

3.1 Optimization under data uncertainty

A large number of problems in production planning and scheduling, allocation, transportation, finance, and engineering design require decisions to be made in the presence of uncertainty. Addressing data uncertainty in mathematical programming models has long been recognized as a central problem in optimization (Bertsimas and Sim, 2003). A key difficulty in optimization under uncertainty is in dealing with an uncertain space that is huge and frequently leads to very large-scale optimization models. This section begins with an overview of the different philosophies encountered in literature on optimization under data uncertainty (Section 3.1.1). Section 3.1.2 discusses the applicability of these models to BOSS and provides recommendations for further research.

3.1.1 Overview of modeling philosophies

Approaches to optimization under uncertainty have followed a variety of modeling philosophies, including expectation minimization, minimization of maximum costs, worst-case optimization, optimization over soft constraints, and minimization of deviation from goals (Sahinidis, 2004). This section briefly discusses these five different philosophies.

Expectation minimization

The early work of Dantzig (1955) recognizes the problem of uncertainty in input data and proposes to minimize the expected cost of a solution. Dantzig divides a problem into two or more stages. Decisions are only required to be taken for quantities or activities in the first stage. Those in the second or later stages cannot be determined in advance, since they depend on the outcomes of the first stage and the realization of uncertainties. For example, in a supply chain context, we should decide on the location and size of production and distribution facilities before actual demand is known. Next, it should be decided what quantities are produced and shipped to customers, based on realized demand and the locations chosen in the first stage. It is important to note that Dantzig assumes that the set of activities is complete in the sense that, whatever choices are made in the earlier stages, there is a possible choice of activities in the later stages.

A formal description of the two-stage stochastic linear program as given by Birge and Louveaux (1997) is:

$$\begin{aligned} \min \quad & c^T x + E_\omega \left[Q(x, \omega) \right] \\ \text{s.t.} \quad & Ax \le b \\ & x \ge 0, \end{aligned} \tag{3.1}$$

with

$$\begin{aligned} Q(x, \omega) &= \min q^T y \\ \text{s.t.} \quad Tx + Wy &\ge h \\ y &\ge 0. \end{aligned} \tag{3.2}$$

Here ω is a random vector formed by the components of q^T, h^T, and T, and W denotes the effect of corrective actions. Problem (3.1) with variables x constitutes the first stage on which decisions need to be made prior to the realization of the uncertain parameters ω. Problem (3.2) constitutes the second stage with variables y. Under the assumption of discrete distributions of the uncertain parameters, the problem can be equivalently formulated as a large-scale linear program, which can be solved using standard linear programming approaches (Sahinidis, 2004).

Mulvey et al. (1995) propose to add a variability measure of the second stage cost, such as variance, to the objective function. However, this introduces nonlinearity into the model and thereby severely complicates the solution process.

Minimization of maximum cost

Like expectation minimization, the *minimization of maximum cost* approach also deals with uncertainty in the cost parameters of the objective function. Consider the following model:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \\ & x \ge 0. \end{aligned} \tag{3.3}$$

We assume for the vector c that $c \in C$. Note that c represents a potential realization of cost, i.e., it occurs with a positive, but possibly unknown probability. Furthermore, the use of scenarios to structure the data uncertainty allows for relationships between the uncertain parameters. The formulation of problem (3.3) that minimizes the maximum cost is:

$$\begin{aligned} \min \max_{c \in C} \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \\ & x \ge 0. \end{aligned} \tag{3.4}$$

Kouvelis and Yu (1997) term these kind of solutions *absolute robust solutions*. They remark that absolute robust solutions are of a conservative nature, as these solutions are based on the notion that the worst might happen. However, such risk averse decision making might be appropriate in an environment in which budgeted values are set as benchmarks to assess the quality of the decisions regardless of the realized scenario. Kouvelis and Yu introduce as an alternative the *robust deviation solution*, which is the decision that exhibits the smallest deviation from the optimal solution over all realizable data input. The objective function is then

$$\min \max_{c \in C} \left(c^T x - c^T x_{c^*} \right), \tag{3.5}$$

where x_{c^*} is the optimal solution to (3.3) in case the realized cost are c. This decision criterion is appropriate for environments in which the quality of the decision is evaluated using the actual realized data.

Worst-case optimization

Ben-Tal and Nemirovski (2000) show that in many practical applications, data uncertainty can cause the usual optimal solution to be infeasible for specific realizations of the data. The *worst-case optimization* modeling philosophy has exactly this issue as a starting point. Thus, the focus is on the reliability of the system, i.e., the ability of the system to remain feasible in an uncertain environment. This approach therefore deals, as opposed to the previous discussed approaches, with uncertainty in the constraint data.

Consider again problem (3.3), however this approach assumes uncertainty in the matrix A and the vector b, instead of in the vector c. Worst-case optimization is concerned with hard constraints, i.e., constraints which must be satisfied for any realization of the data (A, b) within a reasonable prescribed uncertainty set $\mathcal{U}$. The robust counterpart of the uncertain linear programming problem (3.3) is now defined as:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \qquad \forall\, (A, b) \in \mathcal{U} \\ & x \geq 0. \end{aligned} \tag{3.6}$$

The advantage of this method is that $\mathcal{U}$ describes the possible realizations of the data (A, b), without the requirement to specify the corresponding probabilities. Furthermore, even though, the number of restrictions can become very high with a large set of possible data realizations, the model remains linear.

Optimization over soft constraints

The approach in worst-case optimization might seem too pessimistic in many real-life situations, especially when very little is known on the actual data and for safety reasons $\mathcal{U}$ is chosen unnecessarily large. In an attempt to resolve this issue, Charnes and Cooper (1959) developed the probabilistic approach.

In the probabilistic approach, the reliability of the model is expressed as a minimum requirement on the probability of satisfying constraints. In essence, the philosophy of this approach is that infeasibilities in the second stage constraints are allowed only if the probability of occurrence is lower than some predefined value. Thus, also in this approach, uncertainty is assumed in the constraint data, while the cost parameters are assumed to be known with certainty.

Consider again the classical linear programming model (3.3). If we assume that there is uncertainty regarding the matrix A and the right-hand side vector b, then the corresponding probabilistic linear program can be stated as follows:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & P(Ax \leq b) \geq p \\ & x \geq 0, \end{aligned} \tag{3.7}$$

where $p = (p_1, p_2, \ldots, p_m)$ and $p_i \in (0, 1)$ is the minimum required probability of satisfying constraint i. Consider the case with one constraint, a deterministic parameter a, and random right-hand side variable b with probability function F. Let β be such that $F(\beta) = p$. Then the constraint $P(a^T x \leq b) \geq p$ can be written as $a^T x \leq \beta$. In this simple case, the probabilistic model is similar to a standard linear program. Kataoka (1963) applies this model to a transportation problem, where he assumes data to be normally distributed.

Minimization of deviations from goals

Again we consider the mathematical program of (3.3). In the *minimization of deviation from goals* philosophy, uncertainty is not modeled by probability functions, but random parameters are considered as fuzzy numbers and constraints are treated as fuzzy sets. In classical set theory, the membership of elements in a set is assessed in binary terms, i.e., an element either belongs to a set or it does not. However, fuzzy set theory permits the gradual assessment of the membership of elements in a set. In fuzzy mathematical programming, some constraint violation is allowed and the degree of satisfaction of a constraint is defined as a membership function of the constraint. A typical linear membership function for a constraint $a^T x \leq \beta$, where β can take values in the range $[b, b + \Delta b]$, is:

$$u(x) = \begin{cases} 1 & \text{if } a^T x \leq b \\ 1 - \frac{a^T x - b}{\Delta b} & \text{if } b \leq a^T x \leq b + \Delta b \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

Objective functions in fuzzy mathematical programming are treated as constraints, with the upper bound and lower bound defined subjectively by the decision maker.

Many of the developments in the area of fuzzy mathematical programming are based on the early paper by Bellman and Zadeh (1970). To introduce the concept, we consider flexible programming, which is a type of fuzzy programming that deals with right-hand side uncertainties. Problem (3.3) can be rewritten as follows to a flexible programming problem:

$$\begin{aligned} c^T x &\lesssim z_0 + \Delta z \\ Ax &\lesssim b + \Delta b \\ x &\geq 0, \end{aligned} \tag{3.9}$$

where z_0 denotes the optimal objective value to problem (3.3) and $z_0 + \Delta z$ denotes the aspiration level of the decision maker. Since no optimal decision is defined for this problem, any solution that satisfies all constraints can be accepted. By defining $u_1(x), \cdots, u_n(x)$ as the set of membership functions of the constraints of the model, including the constraint on the objective value, an optimal fuzzy decision x^* can be defined. A common choice is to maximize the minimum satisfaction level, i.e.,

$$x^* = \max_{x \geq 0} \min_{i=1,\ldots,n} u_i(x). \tag{3.10}$$

Zimmerman (1978) shows that if all membership functions are linear, then (3.10) can be reduced to a classical linear program. For the membership function as defined in (3.8) this results in:

$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & c^T x + \Delta z \, \lambda \leq z_0 + \Delta z \\ & Ax + \Delta b \, \lambda \leq b + \Delta b \\ & x \geq 0 \\ & 0 \leq \lambda \leq 1. \end{aligned} \tag{3.11}$$

Problem (3.11) includes one more variable and one more constraint than the original problem.

While flexible programming deals with right-hand side uncertainties, possibilistic programming recognizes uncertainties in the objective function as well as in the constraint coefficients. In a similar fashion to (3.11), an equivalent possibilistic program can be derived by introducing a new variable to the model.

3.1.2 Discussion

Section 3.1.1 describes several alternative approaches to deal with uncertainty in the input data. The approaches differ with respect to:

1. the parameters in which uncertainty is assumed; some approaches deal with uncertainty in the right-hand side of constraints, while others handle uncertainty in parameters in the objective function.

2. the way in which the approach deals with uncertainty, i.e., the objective with respect to uncertainty. For example, where one method aims at minimum expected cost, another provides the solution with the lowest maximum cost.

Section 2.3 indicates that for BOSS there is uncertainty in both cost parameters and right-hand side parameters, namely demand. Only the fuzzy mathematical programming approach is able to handle both uncertainties simultaneously. However, other methods might be adapted or combined such that they are also able to handle both variants of uncertain data.

Regarding the second issue, none of the methods directly match the objective we have for BOSS with respect to uncertain input data. We like to obtain solutions which are close to optimal and 'almost' feasible for 'most' scenarios. The first approach, minimization of expected cost, is too narrow for our objective and realized cost may deviate too much from optimality for different scenarios. Extending the objective function by variance, may solve this problem. However, it introduces nonlinearity into the model. This method is therefore not suitable for the large problem instances we want to model with BOSS, since no solver available on the market can solve such a model in reasonable time. Minimizing the maximum cost over different scenarios as well as worst-case optimization takes the worst possible outcome as the relevant scenario for respectively cost parameter and right-hand side uncertainty. Since these approaches produce over-conservative solutions, they are not consistent with our objective. The optimization over soft constraints approach partly solves this issue. It however, requires specifying a probability distribution and it does not account for uncertainty in the cost parameters. Nevertheless, this approach might be relevant for handling uncertainty about customer demand.

The last discussed approach, fuzzy mathematical programming, can handle both types of uncertainties. It allows for some constraint violation for fuzzy defined constraints. Furthermore, this method does not introduce nonlinearity into the model, if membership functions are defined linear. This is in contrast to many other approaches. For these reasons, this method might be very useful for dealing with data uncertainty in BOSS. Further research is required on the choice of the membership function and to determine reasonable bounds on the uncertain input data.

However, we have decided to do no further research into these methods to obtain robust solutions. Instead, we focus on methods to generate alternative solutions and on adapting these methods such that they can also deal with data uncertainty. Section 3.2 discusses different approaches from literature to generate alternative solutions to mathematical programming problems.

3.2 Generating alternative solutions

Modeling to Generate Alternatives (MGA) has been proposed as a framework for dealing with complex problems for which there are important unmodeled issues (Chang et al., 1983). MGA techniques are designed to provide the decision maker with a set of alternatives that are good with respect to the modeled objectives and different from each other in decision space.

Section 3.2.1 gives an overview of approaches to generate alternative solutions. Section 3.2.2 discusses the methods and the applicability to BOSS.

3.2.1 Overview of approaches

This section discusses six different approaches to generate alternative solutions. Assume that we have the following linear programming problem to which we want to find close to optimal alternative solutions:

$$\begin{aligned} \min \quad & Z = c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0. \end{aligned} \tag{3.12}$$

Furthermore, z_0 is the objective value of the optimal solution to (3.12).

Adjust model parameters

One approach to generate alternative solutions is to judgmentally change model parameters or equations, to examine alternative sets of conditions. This approach is applied to scheduling of police cars by Kolesar et al. (1975). In this study, constraints are added to include preferences about starting times of shifts. Also, the number of petrol cars is restricted for each optimization by a different value to obtain different models, resulting in different solutions. Balakrishnan and AltinKemer (1992) vary the right-hand side of one of the constraints in a mixed integer programming model to obtain alternative designs for a network communication system.

Adapting constraints requires a very high level of in-depth knowledge of the problem at hand. Furthermore, this method is ad hoc and provides no means for obtaining solutions that are different, but perform good with respect to the original set of conditions. For these reasons, we do not discuss this method further.

Branch and Bound method

A second method discussed in literature explicitly examines solutions routinely obtained in the optimization process. Nakamura and Brill (1979) and Huang et al. (1997) apply a method they call Brand-and-Bound/Screen (BBS). In this method, all solutions obtained by a branch-and-bound algorithm that is applied in the optimization process, are evaluated.

The branch-and-bound algorithm, which forms the basis of the BBS method, works in such a way that the potential solutions are grouped in mutually exclusive subsets. Lower and upper bounds on the least-cost alternative in the subset are determined. If the lower bound found for subset A is higher than the upper bound for subset B, the solutions in subset A can be discarded. The remaining subsets are further partitioned in smaller mutually exclusive sets. The process continues until a solution is obtained, such that the lower bound on all remaining subsets exceeds its cost. By this process, many solutions are explicitly evaluated. The BBS method obtains alternative solutions by storing those solutions the branch-and-bound algorithm comes across, that are within a certain pre-specified cost bound. The generation step is succeeded by a post-screening step in which solutions that are similar are eliminated from the set of alternative solutions.

The process can be adapted to find not only the encountered, but all solutions within a certain cost bound, by growing the branch-and-bound tree beyond the point required to obtain the optimal solution. This means that a subset is discarded if and only if its lower bound is higher than the specified cost bound, instead of when it is higher than the upper bound of any other subset. However, by this altered approach, the solution process will become computationally highly extensive.

A major advantage of this method is that it does not necessitate running, a potentially computational extensive, optimization run for each required alternative solution. Instead solutions are obtained by examining the routinely obtained solutions in the optimization process. However, the method has no guiding features specifically for producing solutions that are different in decision space, since only those solutions encountered in the search process to the optimal solution are examined. The adapted algorithm, finding all solutions within a certain cost bound, has the large disadvantage that the solution process will become computationally too extensive for the instances we want to solve with BOSS. Since both types of searching the decision space have a unsurmountable disadvantage, we do not discuss this approach further.

Random method

One of the first approaches mentioned in literature to generate alternative solutions, discussed by Brooks (1958), is to randomly generate these alternatives. However, it is in general difficult to generate a set of values of decision variables at random such that the solution is feasible, especially if there are many mathematical constraints. To this end, Chang (1981) develops a method to randomly generate feasible solutions. A set of indices is randomly generated and then the sum over the values of the corresponding decision variables is maximized under the original constraints of the model. Also, a constraint is added to the model to reduce the feasible decision space to a space in which all solutions are good with respect to the original objective function. Mathematically, we can write the model as

$$\max \quad Y = \sum_{j \in J} x_j \tag{3.13}$$

$$\text{s.t.} \quad c^T x \leq (k+1) \cdot z_0 \tag{3.14}$$

$$Ax \leq b \tag{3.15}$$

$$x \geq 0, \tag{3.16}$$

where J is a set of randomly generated indices and $k \geq 0$ is a constant.

Distance metric method

One of the most intuitive approaches to MGA, described, among others, by Kripakaran and Gupta (2006), literally maximizes the ‘distance’ of a solution to previous solutions. A distance metric $\delta_{\alpha,\beta}$ is defined to measure the difference between solution α and solution β. The mathematical model can be formulated as:

$$\max_{\beta} \quad Y = \delta_{\alpha,\beta} \tag{3.17}$$

$$\text{s.t.} \quad (3.14) \text{ - } (3.16) \text{ holds,}$$

where $\alpha = (x_{\alpha,1}, \ldots, x_{\alpha,m})$ is the optimal solution and $\beta = (x_{\beta,1}, \ldots, x_{\beta,m})$ the alternative solution. There are different choices for the metric δ, for example, the euclidean norm $\|(x_{\alpha,1} - x_{\beta,1}, \cdots, x_{\alpha,m} - x_{\beta,m})\|_2$ can be used to measure the difference between two solutions.

Note that this metric is nonlinear. To obtain more alternative solutions, the objective function is modified to include all previously obtained solutions, i.e.,

$$\max_{\beta} \quad Y = \sum_{j=1}^{n} \delta_{j,\beta}, \tag{3.18}$$

where n is the number of previously obtained solutions, including the optimal solution. Naturally, the distance matrix could be defined differently. For example, Kripakaran and Gupta (2006) propose to use the *Hamming distance*, which counts for two solutions the number of variables that differ. Thus, in case of solely binary variables, the Hamming distance equals $\|(x_{\alpha,1} - x_{\beta,1}, \cdots, x_{\alpha,m} - x_{\beta,m})\|_1$.

The main disadvantage of the distance metric method is that, with defining a distance metric which is nonlinear, as are both of the metrics mentioned above, nonlinearity is introduced into the model and solutions are much harder to obtain. Genetic algorithms may be applied to optimize the nonlinear objective function, see Loughlin et al. (2001) and Zechman and Ranjithan (2004) for more details.

HSJ method

Brill et al. (1982) propose a method similar to the previous method, which they call Hop, Skip, and Jump (HSJ). As for the distance metric method, HSJ is designed to generate a solution that is 'maximally' different from the initial solution. To this end, the objective function is modified to set as many as possible decision variables, that are nonzero in the initial solution, to zero. Again the initial constraints are included and supplemented by a constraint on the value of the original objective function. Thus, the first alternative solution is obtained by solving the following system:

$$\begin{aligned} \min \quad & Y = \sum_{j \in J} x_j \\ \text{s.t.} \quad & (3.14) \text{ - } (3.16) \text{ holds,} \end{aligned} \tag{3.19}$$

where J is the set of indices of the decision variables which are nonzero in the initial solution. To obtain the second and following alternatives, a formulation similar to (3.19) can be used, except that J should include all nonzero variables in all previous solutions. A series of alternative solutions can be generated by continuing the process, where general stopping criteria are:

1. a pre-specified number of alternatives is found;
2. no new decision variables enter the solution.

These stopping criteria can obviously also be applied in the previous method. The biggest advantages of this method are the linearity of the model and the easy implementation in any optimization software.

Fuzzy HSJ method

The Fuzzy HSJ approach, as the name reveals, is based on the HSJ method. Section 3.1.1 discusses fuzzy mathematical programming in the context of dealing with uncertain data. Here we only provide a brief summary: in fuzzy mathematical programming, constraints belong to fuzzy sets and objective functions are also treated as constraints with the upper and lower bound specified subjectively by the decision maker. The membership of an element in fuzzy set theory is not assessed in binary terms, but can be gradually assessed. Some constraint violation is allowed and the degree of satisfaction of a constraint is defined by a membership function.

The Fuzzy HSJ method uses a formulation similar to (3.19), except that the objective function and the cost constraint are fuzzy. This means that the solution should 'significantly' differ from the initial solution and the cost should not be 'much higher' than the optimal objective value z_0. The corresponding mathematical formulation replacing (3.19) is:

$$\begin{aligned} &\sum_{j \in J} x_j \lesssim d + \Delta d \\ &c^T x \lesssim z_0 + \Delta z \\ &\text{s.t.} \quad (3.15) \text{ - } (3.16) \text{ holds,} \end{aligned} \tag{3.20}$$

where J is again the set of indices of nonzero variables in the initial solution and d is the optimal difference level desired. Since there is no mathematical method to obtain a solution to this problem directly, the fuzzy relationships in (3.20) should be converted to a corresponding binary formulation. To this end, we assume the membership function, or satisfaction level, of the first constraint to be defined as follows:

$$\mu_1(x) = \begin{cases} 1 & \text{if } \sum_{j \in J} x_j \leq d \\ 1 - \left(\sum_{j \in J} x_j - d \right) \setminus \Delta d & \text{if } d \leq \sum_{j \in J} x_j \leq d + \Delta d \\ 0 & \text{otherwise,} \end{cases} \tag{3.21}$$

where $[d, d + \Delta d]$ gives the desirable range of values from the difference function. The membership function of the second constraint is similarly defined, with the desirable range for cost equal to $[z_0, z_0 + \Delta z]$. Finally, an assumption has to be made on how the decision maker evaluates the satisfaction levels among different objectives. A common choice is to assume that the decision maker wants to maximize the minimum satisfaction of the objectives. In that case, with the choice of membership function as in (3.21), (3.20) can be converted to:

$$\begin{aligned} &\max_{x \geq 0} \min \left[1 - \frac{\sum_{j \in J} x_j - d}{\Delta d}, 1 - \frac{c^T x - z_0}{\Delta z} \right] \\ &\text{s.t.} \quad (3.15) \text{ - } (3.16) \text{ holds.} \end{aligned} \tag{3.22}$$

Zimmerman (1978) has shown that (3.22) can be rewritten to the following system:

$$\begin{aligned} \max \quad & s \\ \text{s.t.} \quad & \sum_{j \in J} x_j + \Delta d \, s \leq d + \Delta d \\ & c^T x + \Delta z \, s \leq z_0 + \Delta z \\ & 0 \leq s \leq 1 \\ & (3.15) \text{ - } (3.16) \text{ holds.} \end{aligned} \tag{3.23}$$

The advantage of the linearity assumption on the membership functions combined with the objective of maximizing the minimum satisfaction, is that the resulting model (3.23) is linear and therefore relatively easy to solve. Chang et al. (1983) apply the fuzzy HSJ approach to two examples, for which they obtain results roughly similar to the original HSJ approach. Also, the results from the original HSJ approach are easier to interpret and implementation is simpler for this method. Therefore, we prefer the original HSJ approach over the fuzzy approach.

3.2.2 Discussion

Section 3.2.1 discusses six different approaches to generate alternative solutions to linear programming problems. Of these approaches, we have already discarded the first two methods. Adjusting model parameters is an ad hoc method which does not guarantee 'good' or even feasible solutions with respect to the original problem. The Branch-and-Bound/Screen method provides no guidance to truly different solutions. The random method by Chang (1981), the distance metric method, and the HSJ approach have in common that they try to maximize the difference between solutions, while restraining the original objective value. The Fuzzy HSJ approach is a variant of the HSJ approach with a different objective function. However, since the original HSJ approach is simpler to implement and results are easier interpreted, we do not discuss the Fuzzy HSJ approach further. Chapter 4 introduces a generalized framework of which we show that it covers the HSJ approach, the random approach, and the distance metric approach.

All methods mentioned are designed for continuous variables, and might be less suitable for integer and binary variables. Chang et al. (1982) have tested several variations of the HSJ method and concluded that the sole use of zero-one variables in the objective function is less effective than the use of continuous variables. The apparent reason for this is that the use of binary variables does not produce a driving force to reduce the capacities of plants and depots if those capacities cannot be forced to zero. However, for our study we have indicated that alternative solutions should differ with respect to locations in use and investments made. Both decisions are represented by binary variables in the mathematical model, see for details Section 2.2. As far as we are aware, no literature has been dedicated to generating alternatives to binary decisions. Nevertheless, all described methods may be applied to binary variables as well. Therefore, Section 4.4 proposes several methods, based on the models discussed in this section.

Chapter 4

Approaches to generate alternative solutions in BOSS

The discussed methods to generate alternative solutions (Section 3.2) are developed for continuous variables. For the location-allocation problem in BOSS (Chapter 2), we are interested in alternatives with respect to binary decisions. Furthermore, we like to obtain solutions that are better able to deal with data uncertainty. This chapter proposes a general framework for methods to generate alternative solutions. This framework can be applied in multiple forms to obtain different types of alternative solutions. In this way, the framework can be used to deal with both types of indicated uncertainties relevant in mathematical optimization, i.e., model uncertainty and data uncertainty.

For binary variables, Section 4.1 shows the equivalence of the distance metric method and the HSJ method. Section 4.2 introduces the general framework based on the HSJ approach and Section 4.3 provides a description of this framework applied to the location-allocation problem in BOSS. Subsequently, In Section 4.4, we propose several realizations of the framework to generate alternative solutions to the location-allocation problem in BOSS. Section 4.5 discusses stopping criteria for the proposed methods. Finally, Section 4.6 describes a number of quantitative measures to evaluate the results of these methods, which we use when we apply the methods to test cases in Chapter 5.

4.1 Distance metric method for binary variables

The distance metric method obtains an alternative solution by literally maximizing the difference of a new solution with the initial solution. To this end, a so-called distance metric is maximized. In literature, the following two distance metrics are applied:

1. Hamming distance metric: $\|\mathbf{x}_\alpha - \mathbf{x}\|_1$;

2. Euclidean distance metric: $\|\mathbf{x}_\alpha - \mathbf{x}\|_2$,

where $\mathbf{x}_\alpha$ is the vector containing the original solution and $\mathbf{x}$ is the alternative solution. Section 4.1.1 shows the consequences of applying the Hamming distance metric to binary decision variables and Section 4.1.2 does the same for the Euclidean distance metric. Subsequently, Section 4.1.3 compares the application of both distance metrics with the HSJ method.

4.1.1 Hamming distance metric

The Hamming distance metric measures the sum of the absolute differences between all individual elements of two vectors. For ease of notation, we introduce a set J, such that

- J : set of indices of the (binary) decision variables that are 1 in the initial solution.

With this notation, we have that

$$x_{\alpha,j} = 1 \quad \forall\, j \in J,$$
$$x_{\alpha,j} = 0 \quad \forall\, j \notin J.$$

It holds that

$$\begin{aligned} \|\mathbf{x}_\alpha - \mathbf{x}\|_1 &= \sum_{j=1}^{m} |x_{\alpha,j} - x_j| \\ &= \sum_{j\,\in\,J} (1 - x_j) + \sum_{j\,\notin\,J} x_j \\ &= \sum_{j\,\in\,J} 1 - \sum_{j\,\in\,J} x_j + \sum_{j\,\notin\,J} x_j. \end{aligned} \tag{4.1}$$

We have that maximizing

$$-\sum_{j\,\in\,J} x_j + \sum_{j\,\notin\,J} x_j + constant, \tag{4.2}$$

is equivalent to minimizing

$$\sum_{j\,\in\,J} x_j - \sum_{j\,\notin\,J} x_j. \tag{4.3}$$

Thus, we have shown in this section that, for binary variables, maximizing the Hamming distance metric is equivalent to the formulation in (4.3). This means that, by applying this metric, the objective is to maximize the number of decision variables that have a different value from the initial solution. In other words, the objective is to change as many decisions as possible.

4.1.2 Euclidean distance metric

The Euclidean distance metric measures the sum of the square of the difference between all individual elements of two vectors. We define $\widetilde{x} = x_j - x$. Since, $x_j, x \in \{0, 1\}$, it holds that $\widetilde{x} \in \{-1, 0, 1\}$. In general, for a vector $x \in \{-1, 0, 1\}^n$, it holds that $\sum_{j=1}^{m} |x_j| = \sum_{j=1}^{m} x_j^2$. Therefore, we have that

$$\|\mathbf{x}_\alpha - \mathbf{x}\|_2 = \|\mathbf{x}_\alpha - \mathbf{x}\|_1. \tag{4.4}$$

We conclude from this, by following the same argumentation as in Section 4.1.1, that maximizing the Euclidean distance metric is equivalent to minimizing equation (4.3). Thus, for both metrics, the objective is to change as many as possible 0-1 decisions.

4.1.3 Comparison of distance metric and HSJ method

For binary variables, there is a strong resemblance between the distance metric method and the HSJ method. In the HSJ approach, the objective is

$$\min \sum_{j\,\in\,J} x_j. \tag{4.5}$$

Sections 4.1.1 and 4.1.2 show that the distance metric method for binary variables, with the two metrics applied in literature, has the objective

$$\min \left\{ \sum_{j \in J} x_j - \sum_{j \notin J} x_j \right\}. \tag{4.6}$$

Both models are subject to all restrictions of the initial model, complemented by a restriction on the maximum cost of an alternative solution. Thus, besides for the existence of the second term in (4.6), the HSJ method and the distance metric method are completely equal.

In words, the HSJ approach minimizes the sum over the decision variables that are nonzero in the initial solution and the distance metric method minimizes this same quantity, while also maximizing the sum over the decision variables that are zero in the initial solution. We refer to decision variables that are nonzero as *impact variables*. The objective in (4.5) forces indirectly variables that are zero in the initial solution to become impact variables. This is caused by the fact that, as far as the cost restriction allows it, all nonzero variables are forced to become zero and the original restrictions still have to be fulfilled. For BOSS the primary strategic decisions are the choice of production and distribution locations to be used and the investments to be made. Since the HSJ approach attempts to close as much as possible current locations, automatically new locations are opened to fulfill demand. The difference is that (4.6) also maximizes the number of non-impact variables that become impact variables. For BOSS that implies that the model maximizes the number of opened locations that were not opened in the initial solution. This means for example, that the distance metric model prefers to open two small depots over one larger depot. However, since we search for solutions that open different depots, not necessarily more depots, this is not a sensible objective for our model.

Furthermore, the second term of the objective function in (4.6) may result in unnecessary high cost. This is for example the case, if an investment can be done within the cost restriction, but this investment does not contribute to the performance of the supply chain. Therefore, we conclude that for binary decision variables the distance metric method is less appropriate than the HSJ method; hence, we do not further explore the distance metric method.

4.2 Generalized HSJ approach

Section 3.2.1 discusses the Hop, Skip, and Jump (HSJ) method developed by Brill et al. (1982). In this section, we formulate a more general form of this method, such that it can be applied for multiple purposes. The HSJ method generates alternative solutions by introducing a new objective function, while the original objective value may not deteriorate by more than a pre-specified amount. This new objective function maximizes the difference among alternative solutions. The first alternative solution is obtained by minimizing the sum over the decision variables that are nonzero in the initial solution. To obtain the second and following alternatives, the set of decision variables in the objective function is expanded by all decision variables that are nonzero in any of the previously obtained solutions.

The HSJ method is most suitable for continuous variables, and it is less useful for binary variables (Chang, 1981). However, Section 2.4 motivates that the decision variables in which we like to see differences are binary variables. Furthermore, the HSJ method is designed such that it returns maximally different alternative solutions. Nevertheless, we like to generalize this method, such that it can also be used with a different purpose. For example, as discussed in Section 2.3, the demand and several cost parameters are not known with certainty when a strategic supply chain design is evaluated. Therefore, it contributes to the applicability of our model, if it can also be used to obtain solutions that are less sensitive to data changes.

We write the Generalized HSJ (GHSJ) method as:

$$\begin{aligned} \min \sum_{j \in J} \alpha_j \cdot x_j \; - \; & \sum_{k \in K} \beta_k \cdot y_k - \gamma \cdot S \qquad (4.7) \\ \text{s.t.} \quad c^T x + S &= (k+1) \cdot z_0 \\ Ax + y &= b \\ x, y &\geq 0, \end{aligned}$$

where J and K are given sets of indices. Whereas for the discussed HSJ method by Brill et al. J contains all nonzero variables in the initial solution and K is empty, the choice of J and K is free in the GHSJ method. Furthermore, z_0 is the optimal solution to the original problem (3.12), k is a constant, and $\boldsymbol{\alpha} = \{\alpha_j \,|\, j \in J\}$ is a vector containing given weight factors. The vector y is introduced to measure the difference between used capacity and available capacity, we refer to this difference as *slack*. By increasing the value of a slack variable, more robustness is created against right-hand side uncertainties. The vector $\boldsymbol{\beta} = \{\beta_k \,|\, k \in K\}$ contains given weight factors corresponding to the slack variables.

The variable S indicates the difference between the realized and allowed cost increase with respect to the optimal solution. By increasing the value of S, a solution with lower total cost is obtained. Appendix D provides a discussion on the choice of the weight factor γ.

Note that the choice of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, γ, J, and K can be chosen based on the approach taken. For example, for the standard HSJ approach of Brill et al. (1982) we have $\boldsymbol{\alpha} = \mathbf{1}$, $\boldsymbol{\beta} = \mathbf{0}$, $\gamma = 0$, K is empty, and J contains all indices of the nonzero variables in the optimal solution. Similarly, by choosing $\boldsymbol{\alpha} = -\mathbf{1}$, $\boldsymbol{\beta} = \mathbf{0}$, $\gamma = 0$, K empty, and J is the set of randomly generated indices, the model is equal to the random approach by Chang (1981), discussed in Section 3.2.1. Thus, the model in (4.7) offers a lot of flexibility and can therefore be used for different approaches. Section 4.4 discusses these approaches. However, first Section 4.3 describes the application of the GHSJ method to BOSS.

4.3 GHSJ method for BOSS

For the objective function (4.7), we define X as the set of decision variables belonging to J: $X = \{x_j \,|\, j \in J\}$. X contains variables of the following form:

- $LocOpen\,(l)$;
- $InvDone\,(i)$;
- $Flow\,(p, l_1, l_2)$;
- $Production\,(f)$;
- $Supply\,(p, l)$.

Note that X contains only direct decision variables from the model described in Chapter 2, i.e., all dependent decision variables are excluded. Furthermore, we define Y as the set of decision variables belonging to K: $Y = \{y_k \,|\, k \in K\}$. This set contains slack variables, i.e., variables to measure the robustness of a solution with respect to right-hand side uncertainties. Y contains variables of the following form:

- $ProdSlack\,(f)$: unused production capacity of facility f;

- $SmallestProdSlack = \min_{f \in \mathcal{F}} ProdSlack\,(f)$: smallest unused production capacity over all facilities;
- $FlowSlack\,(p, l)$: unused distribution capacity for product p and location l;
- $SmallestFlowSlack\,(p) = \min_{l \in \mathcal{L}} FlowSlack\,(p, l)$: smallest unused distribution capacity for product p over all locations.

While the actual content of the objective function in (4.7) depends on the choice of the method, the constraints that should hold for the model are always the same. These constraints are mainly equal to those discussed in Section 2.2. However, some constraints should be adapted to introduce the variables in Y and other constraints need to be added to the mathematical model. We measure the production and distribution slack as the difference between available and used capacity. And this slack is only nonzero for locations that are used. We now discuss the adaptations to the original model of Chapter 2.

The optimal value of the original objective function equals z_0. The constraint that ensures that the original objective does not deviate too much from optimality can now be written as:

$$Cost + S \;=\; (k+1) \cdot z_0, \tag{4.8}$$

where $Cost$ denotes the cost with respect to the original objective. Constraint (2.12) should be replaced by two new constraints to enforce a correct calculation of the production slack $ProdSlack\,(f)$ of production facility f and the minimum production slack $SmallestProdSlack$. As opposed to the model described in (4.7), the introduction of these slack variables does not result in the constraints getting an equality sign. The reason for this is that $Prodslack\,(f)$ should equal zero if the location is not in use. $SmallestProdSlack$ does not depend on facility f, whereas the constraint does. Therefore, in this constraint equality cannot hold. We model the described situation by the following two constraints:

$$Production\,(f) + ProdSlack\,(f) \;\leq\; AvailProdCap\,(f) \qquad \forall\, f \in \mathcal{F}; \tag{4.9}$$
$$Production\,(f) + SmallestProdSlack \;\leq\; AvailProdCap\,(f) \qquad \forall\, f \in \mathcal{F}. \tag{4.10}$$

To ensure that $ProdSlack\,(f)$ is equal to zero if production facility f is located at a location that is not opened, a third new constraint is added to the model:

$$\sum_{f \in F^l} ProdSlack\,(f) \;\leq\; LocOpen\,(l) \cdot bigMProd\,(l) \qquad \forall\, l \in \mathcal{L}, \tag{4.11}$$

where

$$bigMProd\,(l) \;=\; \sum_{f \in F^l} \left(\max \left(\max_{i \in \mathcal{I}^{Pr}(f)} (investprodcap\,(i, f))\,, prodcap\,(f) \right) \right). \tag{4.12}$$

For constraint (2.14), giving a restriction on the distribution capacity, an equivalent situation holds. Therefore, this is modeled by the following three constraints:

$$\sum_{l_1 \in \mathcal{L}} Flow\,(p, l, l_1) + FlowSlack\,(p, l) \;\leq\; AvailFlowCap\,(p, l) \qquad \forall\, p \in \mathcal{P},\ l \in \mathcal{L}; \tag{4.13}$$
$$\sum_{l_1 \in \mathcal{L}} Flow\,(p, l, l_1) + SmallestFlowSlack\,(p) \;\leq\; AvailFlowCap\,(p, l) \qquad \forall\, p \in \mathcal{P},\ l \in \mathcal{L}; \tag{4.14}$$
$$\sum_{p \in \mathcal{P}} FlowSlack\,(p, l) \leq LocOpen\,(l) \cdot bigMFlow\,(p, l) \qquad \forall\, l \in \mathcal{L}, \tag{4.15}$$

where

$$bigMFlow\,(p,l) \;=\; \max\left(\max_{i\in\mathcal{I}^f l(l)}\left(investflowcap\,(i,p,l)\right), flowcap\,(p,l)\right). \tag{4.16}$$

Furthermore, all other constraints remain valid, i.e., constraints (2.2) - (2.4), (2.6) - (2.10), (2.11), and (2.13) should hold.

4.4 Realizations of GHSJ framework

Section 4.3 describes the GHSJ framework for BOSS. This section proposes several realizations of this framework. Section 4.4.1 describes the details of applying the HSJ method by Brill et al. (1982) to the location-allocation problem in BOSS. However, the framework offers more flexibility to apply it in different manners. Therefore, Section 4.4.2 discusses a second realization of the GHSJ framework to generate a large amount of alternative solutions.

The first two realizations of the GHSJ method correspond to the research objective to deal with model uncertainty. However, Section 1.4.2 also indicates a research objective with respect to data uncertainty. For BOSS we have indicated two types of data uncertainty, namely:

- right-hand side uncertainty: uncertainty about demand data;
- uncertainty in the objective function: uncertainty about cost parameters.

Therefore, we describe for both types of data uncertainty a realization of the GHSJ model to account for it. These methods are discussed in respectively Section 4.4.3 and Section 4.4.4.

This section concludes with the description of some other realizations of the GHSJ framework that are business wise relevant, but are not applied to test cases in this research.

4.4.1 Standard HSJ

The Standard HSJ implements the HSJ approach as described by Brill et al. (1982) for the binary decision variables $LocOpen\,(l)$ and $InvDone\,(i)$. However, we add the decision variable S to the objective function. Thus, the objective function is:

$$\min \quad \sum_{l\,\in\,J_L} LocOpen\,(l) + \sum_{i\,\in\,J_I} InvDone\,(i) - \gamma\,\cdot\,S, \tag{4.17}$$

where J_L is for $LocOpen\,(l)$ and J_I for $InvDone\,(i)$ the set of indices of nonzero variables in all previously obtained solutions. Thus:

$$\begin{aligned} J_L \;&=\; \{\, l \mid LocOpen\,(l) = 1 \text{ in at least one previously obtained solution}\}, \\ J_I \;&=\; \{\, i \mid InvDone\,(i) = 1 \text{ in at least one previously obtained solution}\}. \end{aligned}$$

This method fits in the GHSJ framework by choosing $\boldsymbol{\alpha} = \mathbf{1}$, J as defined above, K empty, and X contains variables of the form $LocOpen\,(l)$ and $InvDone\,(i)$.

The process of updating J_L and J_I, and solving the resulting problem is repeated until one or more of the following happens:

1. $J_L = \mathcal{L}$ and $J_I = \mathcal{I}$, i.e., all decision variables have been added to the objective function;
2. the obtained solution has the same value for all variables $LocOpen\,(l)$ and $InvDone\,(i)$ as one of the previously obtained solutions;

3. a total of $MaxSol$ different solutions is obtained.

The objective of the Standard HSJ approach is to obtain solutions, within the allowed cost-range, that are as different from each other as possible with respect to $LocOpen\,(l)$ and $InvDone\,(i)$. The choice for these two decision variables is motivated in Section 2.4. The Standard HSJ method only generates a limited number of alternative solutions. However, this is not necessarily a disadvantage, since human beings can not process and evaluate more than a few alternatives.

4.4.2 Random HSJ

The Random HSJ method uses an objective function with randomly chosen weights for the variables $LocOpen, (l)$ and $InvDone\,(i)$. Also, we add the variable S to the objective function. Thus, the objective function is:

$$\min \quad \sum_{l \in \mathcal{L}} \alpha_L(l) \cdot LocOpen\,(l) + \sum_{i \in \mathcal{I}} \alpha_I(i) \cdot InvDone\,(i) - \gamma \cdot S, \tag{4.18}$$

where $\alpha_L\,(l)$ is the weight corresponding to $LocOpen\,(l)$, and $\alpha_I\,(i)$ is the weight corresponding to $InvDone\,(i)$. Both weights are in each iteration randomly generated from the range $[0, 1]$. This method corresponds to the GHSJ framework by choosing $\alpha = \{\alpha_L\,(l), \alpha_I\,(i)\}$ random from $[0, 1]$, $J = \mathcal{L} \cup \mathcal{I}$, K empty, and X contains variables of the form $LocOpen\,(l)$ and $InvDone\,(i)$. The process of generating new weights, and solving the resulting objective function, is repeated until either $MaxSol$ solutions are obtained or no new solution is obtained for $MaxEq$ iterations. Section 4.5 discusses stopping criteria in more detail.

The objective of the Random HSJ approach is to generate many close to optimal solutions. However, this approach has no guiding feature to obtain solutions that are different from previously obtained solutions. Therefore, after each iteration, an obtained solution that does not differ by at least one decision variable from all of the previously obtained solutions is removed from the set of solutions in a post-screening step.

4.4.3 Cost uncertain HSJ

The goal of the Cost uncertain HSJ method is to account for uncertainty in the cost parameters. For this purpose, we use an objective function that differs significantly from those used for the Standard HSJ and Random HSJ method. For this model the original objective function (2.16) is slightly modified by replacing the uncertain cost parameters by a random number in the specified uncertain range. Thus, the objective function for this model is:

$$\begin{aligned}\min \sum_{l\in\mathcal{L}} \alpha_L(l) \cdot LocOpen\,(l) &+ \sum_{i\in\mathcal{I}} \alpha_I(i) \cdot InvDone\,(i) + \sum_{\substack{p\in\mathcal{P}\\ l\in\mathcal{L}}} \alpha_S(p,l) \cdot Supply\,(p,l) \\ &+ \sum_{f\in\mathcal{F}} \alpha_P(f) \cdot Production\,(f) + \sum_{l\in\mathcal{L}} \sum_{\substack{m\in\mathcal{M}\\ f\in\mathcal{F}^l}} \alpha_C(l,m,f) \cdot Production\,(f) \cdot recipe_f\,(m) \\ &+ \sum_{\substack{p\in\mathcal{P}\\ l_1,l_2\in\mathcal{L}|l_1\neq l_2}} \alpha_F(p,l_1,l_2) \cdot Flow\,(p,l_1,l_2).\end{aligned} \tag{4.19}$$

To see that (4.19) is a realization of the general form of the objective function in the GHSJ framework in (4.7), we choose $\boldsymbol{\alpha} = \{\alpha_L(l), \alpha_I(i), \alpha_S(p,l), \alpha_P(f), \alpha_C(l,m,f), \alpha_F\,(p,l_1,l_2)\}$, X contains variables of the form $LocOpen\,(l)$, $InvDone\,(i)$, $Supply\,(p,l)$, $Production\,(f)$, and $Flow\,(p,l_1,l_2)$. Furthermore, $\boldsymbol{\beta} = \mathbf{0}$, $\gamma = 0$, and K is empty.

Objective (4.19) strongly resembles the original objective (2.16), except that the cost parameters are replaced by random numbers. These random numbers are drawn from an interval specified by the modeler. For example, assume that the cost of supplying one unit of product A to location X is not known exactly when a strategic plan is made. However, the modeler is fairly sure that it costs between €1.20 and €1.40. For the initial model, the modeler should decide on a single most likely value of this cost parameter, for example, €1.30. A solution is then obtained that is optimal in case the realized cost parameters are equal to the expected values. However, this does not mean that it would also be optimal in case the realized cost parameters are different than expected.

In the Cost uncertain HSJ approach, alternatives are generated by using different random values within the specified cost ranges. Thus, for the example $\alpha_S(A, X)$ is drawn from the interval $[1.20, 1.40]$. However, the total cost of an alternative solution, calculated with the expected cost parameters, should be below the specified bound, as indicated in constraint (4.8). The reason for this is that the decision maker might not want the total cost for the expected case to increase too much, since it might be difficult to sell such a solution business wise. However, since the objective of this method is still minimizing cost, k could also be set sufficiently high, such that it has no influence on the obtained solution. For each alternative solution, the cost in case of m different realizations can easily be calculated. To this end, given that cost have no impact on the feasibility of a solution, only the cost function should be recalculated. The alternatives can then be compared on total cost in the different realizations of the uncertain cost parameters.

In conclusion, the objective of the Cost uncertain HSJ approach is to obtain one or more alternative solutions to a problem where uncertainties in cost parameters play a role. Such a solution should have lowest cost in a higher number of cost realizations than the original solution or a lower average cost over these realizations.

4.4.4 Robust HSJ

The goal of the Robust HSJ method is to obtain one or more alternative solutions that are more robust against right-hand side uncertainties, those are uncertainties in demand and capacities. To measure this robustness, we have introduced production and distribution slack variables in Section 4.3. These slack variables measure the difference between available and actually used capacity. We define two versions of the Robust HSJ method, one that maximizes individual slack and one that maximizes minimum slack. These two versions are complementary to each other. Both versions fit in the GHSJ framework by choosing J empty, $\boldsymbol{\alpha} = \mathbf{0}$, and $\gamma = 0$. The value of $\boldsymbol{\beta}$ and the content of Y differs for both methods.

Version 1

For the Robust HSJ method (version 1), the aim is to find a number of alternative solutions that have a higher total slack as the initial solution and allocate slack differently over locations and production facilities. For this method we define the following two variables, that measure the utilization of capacity:

- $\beta_{PS}(f)$: equals the utilization of production facility f in the previously obtained solution.
- $\beta_{FS}(p, l)$: equals the utilization of the distribution capacity for product p at location l in the previously obtained solution.

To denote previously obtained solutions, we add a subscript s to the decision variables. Then, we calculate the utilizations by dividing the production or distribution by the actual capacities as defined in (2.11) and (2.13):

$$\begin{aligned}\beta_{PS}(f) &= \frac{Production_s\,(f)}{AvailProdCap_s\,(f)};\\ \beta_{FS}(p,l) &= \frac{\sum_{l_1\in\mathcal{L}|l_1\neq l} Flow_s\,(p,l_1,l)}{AvailFlowCap_s\,(p,l)}.\end{aligned}$$

This approach maximizes the sum over the slack variables $ProdSlack\,(f)$ and $FlowSlack\,(p,l)$. However, we attach weights to these variables equal to the utilization of the capacity in the previous solution. Thus, the objective function of the Robust HSJ method (version 1) is:

$$\max \quad \sum_{f\in\mathcal{F}} \beta_{PS}(f)\cdot ProdSlack\,(f) + \sum_{\substack{p\in\mathcal{P}\\ l\in\mathcal{L}}} \beta_{FS}(p,l)\cdot FlowSlack\,(p,l). \tag{4.20}$$

With this approach a higher weight is assigned to the slack for higher utilized facilities and distribution locations. The rational behind this is that it should result in more slack for production facilities and distribution locations that have little slack in the last obtained solution. Thereby, this method does not only potentially increases total slack, it also allocates slack over different locations and facilities.

The discussed parameters are updated after each new obtained solution, and so the weights $\beta_{PS}\,(f)$ and $\beta_{FS}\,(p,l)$ are recalculated after each iteration. If a solution is obtained that has been found before, calculated weights stay equal to those in a previous run. Therefore, the objective function is the same and the run can be stopped.

Version 2

The Robust HSJ method (version 2) aims to increase the slack for all production facilities and distribution locations, instead of allocating slack differently and maximizing total slack. To this end, we maximize the minimum slack for both production and distribution. This version of the robust HSJ approach uses the objective function:

$$\max \quad SmallestProdSlack + \sum_{p\in\mathcal{P}} SmallestFlowSlack\,(p), \tag{4.21}$$

where the decision variables $SmallestProdSlack$, and $SmallestFlowSlack\,(p)$ are as defined in Section 4.3. With this approach an alternative solution is obtained in which the minimum production slack is maximized together with the minimum distribution slack. Thus all production slack variables $ProdSlack\,(f)$ are at least equal to $SmallestProdSlack$. Equivalently, for each location l the decision variable $FlowSlack\,(p,l)$ is at least equal to $SmallestFlowSlack\,(p)$. Therefore, we obtain by this approach an alternative solution where production slack and distribution slack are better balanced over the different facilities and locations.

In contrast to Robust HSJ (version 1), this approach only obtains one alternative solution per choice of the cost bound k, since there are no weights that alter after an iteration.

4.4.5 Dedicated methods

In this section we briefly describe some other methods that are business wise relevant and can be implemented within the GHSJ framework. To this end, only the proposed objective function, corresponding to a specific goal of the decision maker, has to be implemented. Even thought, this can be easily done, we have not done experiments with these dedicated methods.

Minimize CO_2 emission

Nowadays, reducing the carbon footprint is an important objective for many organizations. The carbon footprint is expressed in the amount of Carbon Dioxide (CO_2) emission. A lower CO_2 emission can for example be achieved by combining truckloads or producing in a more efficient production mode. The objective function for this approach is:

$$\begin{aligned} \min \quad & \sum_{f \in \mathcal{F}} Production\,(f) \cdot recipe_f\,(CO_2) \\ & + \sum_{\substack{p \in \mathcal{P} \\ l_1, l_2 \in \mathcal{L} | l_1 \neq l_2}} Flow\,(p, l_1, l_2) \cdot emission\,(CO_2, p, l_1, l_2), \end{aligned} \tag{4.22}$$

where $emission\,(CO_2, p, l_1, l_2)$ is a new introduced parameter giving the amount of CO_2 emission for the transport of 1 unit of product p from location l_1 to location l_2.

Minimize fixed cost

When having to make a trade-off between fixed and variable cost, many decision makers prefer a solution with higher variable cost and lower fixed cost. The reason is that fixed cost create a financial risk to the company, where purely variable cost do not. Take, for example, a company that expects a growing demand and should therefore increase production capacity. This company can choose between two machines:

- Machine A: cost €12,000,000 and produces one unit of product X for €1.00;
- Machine B: cost €15,000,000 and produces one unit of product X for €0.70.

Suppose, that this company expects to sell 15 million units of product X in the next 5 years and 5 years is also the depreciation period of both machines. Furthermore, we assume capacity and maintenance cost of both machines to be equal. The total cost when the company chooses for machine A are: $12,000,000 + 15,000,000 \cdot 1.00 = €27,000,000$. When it chooses for machine B, total production cost are: $15,000,000 + 15,000,000 \cdot 0.70 = €25,500,000$. From these numbers, one could conclude that machine B is the best option. However, when actual demand is lower than 10 million, this is no longer the case. Another reason for a lower investment to be preferred is that available funds might be limited.

To obtain an alternative solution with lower fixed cost, the following objective function should be minimized:

$$\begin{aligned} \min \quad & \sum_{l \in \mathcal{L}} LocOpen\,(l) \cdot fixedlocationcost\,(l) \\ & + \sum_{i \in \mathcal{I}} InvDone\,(i) \cdot fixedinvestmentcost\,(i). \end{aligned} \tag{4.23}$$

The GHSJ framework ensures that total cost of the alternative solution does not increase by more than k times the minimum cost z_0.

Minimize external purchasing

A company confronted with a so-called make-or-buy decision has to decide on either manufacturing in-house or purchasing from an external supplier. In a make-or-buy decision, the two most important factors to consider are cost and availability of production capacity. However, even if purchasing from an external supplier is cheaper, preferences may be to produce in-house, if there is idle production capacity. Reasons for this include better quality control or proprietary technology that needs to be protected.

We could use the GHSJ framework to obtain an alternative solution with a minimal amount of supply from external sources, while total cost increases by at most k times the optimal cost. The objective function for this approach is:

$$\min \quad \sum_{\substack{p\in\mathcal{P}\\ l\in\mathcal{L}}} \mathit{Supply}\,(p,l). \tag{4.24}$$

Products that cannot be manufactured in-house can be excluded from this objective function.

4.5 Stopping criteria

When applying the GHSJ method, there are some method specific stopping criteria that determine when no new iteration is started anymore. For the standard HSJ method, it holds that the method stops when the last obtained solution does not set any binary variable to 1 that has not been set to 1 in any of the previously obtained solutions. Thus, if the new solution has no used locations or investments that where not used in all earlier solutions, the method stops. This is an inherent characteristic of the method, since the objective function is in that case exactly the same as for the previous iteration. Thus, optimizing this objective function again results in the same solution. The Robust HSJ (version 1) method stops when a solution is obtained that has been found before. Again, the reason for this is that the new objective function is in that case equal to the objective in one of the previous iterations. The Robust HSJ method (version 2) only obtains one alternative solution, since the objective function does not change after an iteration.

Theoretically, the Random HSJ and Cost uncertain HSJ method can be repeated infinitely many times. Therefore, we introduce two stopping criteria for these methods. The first reason to stop the method is when *MaxSol* number of solutions are obtained, including the initial solution. Since humans cannot process more than 5 to 10 alternatives, in general a reasonable value for *MaxSol* would be around 10. For testing purposes, we often set this value higher to obtain more solutions. The second stopping criterion is when *MaxEq* number of times a solution is obtained that is already in the set of obtained solutions. This second stopping criterion ensures that the method stops in a finite number of iterations. The maximum number of iterations is thus $MaxSol + MaxEq$, while the minimum number of iterations is $\min(MaxSol, MaxEq)$.

4.6 Evaluate results

To be able to evaluate the obtained set of alternative solutions, we need quantitative measures. This is especially relevant when the purpose is to obtain solutions that make different decisions, i.e., for the Standard HSJ and Random HSJ. Since it is then not evident how to assess the obtained solutions. One obvious measure to quantify a solution is the value of the original objective function, specifying the 'real' cost of the solution. Derived from this, we can calculate for an alternative solution the deviation from optimality as a percentage of the optimal objective value.

Since the purpose of generating alternatives is to account for unmodeled objectives, alternative solutions could be evaluated based on these objectives. Even though, many objectives might be unknown or difficult to measure, some objectives can be quantified. For our model, the total transport distance is relevant, since it largely determines the degree of air pollution. However, it is not modeled as an objective in the objective function. Another example of such an unmodeled objective is the fraction of fixed cost. We use these two extra measures to evaluate the alternative solutions.

Another relevant measure is the degree of difference among alternative solutions. The alternative solutions must be close to optimal with respect to the original objective, i.e., cost of an alternative solution may not increase by more than k times the cost of the initial solution. However, the alternatives are typically different with respect to the values of the decision variables, and these differences can be measured. To calculate the difference between a new solution and all previous solutions, Chang (1981) first calculates the pairwise differences. The pairwise difference is calculated as the number of different nonzero variables between two solutions. Then he defines the least pairwise difference as the smallest difference between the new solution and any of the previous solutions. We refer to this measure as the *binary difference metric (BDM)* and define it formally as

$$BDM\,(s) \;=\; \min_{s' \in S} \; \sum_{i \in B} |s_i - s'_i|, \tag{4.25}$$

where S is the set of solutions, B is the set of binary decision variables, and s_i indicates the value of binary decision variable i. Since the Cost uncertain HSJ and Robust HSJ method use, besides binary variables, also continuous variables in the objective function, we add another difference measure that takes the influence of these different decisions into account. To this end, we multiply the pairwise difference on each decision variable by the influence of this decision variable, which we define by $C(i)$. For $LocOpen\,(l)$, we define the influence as the capacity of location l and for $InvDone\,(i)$ we define it as the absolute difference between the capacity of the corresponding location if investment i is made and when it is not. We refer to this measure as the *capacity difference metric (CDM)* and define it formally as

$$CDM\,(s) \;=\; \min_{s' \in S} \; \sum_{i \in B} |s_i - s'_i| \cdot C(i). \tag{4.26}$$

These measures give an indication of the difference of an alternative solution with respect to previously obtained solutions, which could for example be used to exclude solutions that do not differ enough. This is, however, not our primary purpose with the difference metrics.

To draw conclusions about the performance of the methods, we like to have a metric that gives insight in the degree of diversity of solutions that can be obtained with a method. To this end, we introduce the *diversity measure* ($D(S)$) of Danna et al. (2007), which defines the diversity of a set S of solutions as the average pairwise distance, i.e.,

$$D(S) = \frac{1}{|S|^2|B|} \sum_{s,s' \in S} \sum_{i \in B} |s_i - s'_i|. \tag{4.27}$$

A higher value of $D(S)$ indicates a higher degree of diversity in the set of solutions. It holds that $D(S) \leq \frac{1}{2}$ for all sets S of solutions. To see this, we look at each variable $i \in B$ individually. It is clear that $\sum_{s,s' \in S} |s_i - s'_i|$ is maximal if $\lfloor |S|/2 \rfloor$ of the solutions have $s_i = 1$ and the remaining solutions have $s_i = 0$. In that case, at most half of the terms $|s_i - s'_i|$ are equal to 1 and at least half of them are 0. It follows that $D(S) \leq \frac{1}{|S|^2|B|} \sum_{i \in B} \frac{1}{2}|S|^2 = \frac{1}{2}$.

4.7 Conclusion

The Hop, Skip, and Jump (HSJ) method by Brill et al. (1982) can be generalized such that it can be applied for multiple purposes, i.e.,

1. the Standard HSJ method is proposed to obtain maximally different alternatives;

2. the Random HSJ method is proposed to obtain a large number of different alternatives;
3. the Cost uncertain HSJ method is proposed to obtain alternatives that are more robust against cost parameter uncertainties than the initial solution;
4. the Robust HSJ method is proposed to obtain alternatives that are more robust against right-hand side uncertainties than the initial solution.

For these methods the following two stop criteria were defined:

- *MaxSol*: the maximum number of obtained solutions, including the initial solution.
- *MaxEq*: the maximum number of times a solution is obtained that is already in the set of solutions.

The total number of iterations is thus between $\min(MaxSol, MaxEq)$ and $MaxEq + MaxSol$. Furthermore, when applying any of the above mentioned methods, the modeler has to decide how much total cost of an alternative solution may deviate from optimality, by setting the value of k. For example, if $k = 0.1$, cost of an alternative solution may be at most 10% higher than the cost of the optimal solution.

Finally, a number of measures were introduced to quantify the performance of alternative solutions, such as actual cost deviation and total transport distance. Furthermore, the binary difference metric $BDM\,(s)$ and the capacity difference metric $CDM\,(s)$ can be applied to measure the difference between a solution and the current set of solutions. The diversity of the entire set of solutions can be measures with the diversity measure $D\,(S)$ by Danna et al. (2007).

Chapter 5

Application of HSJ methods to test cases

Chapter 4 proposes four methods to obtain alternative solutions to the location-allocation problem in BOSS, namely the Standard HSJ, Random HSJ, Cost uncertain HSJ, and Robust HSJ method. Each of these four methods correspond to a specific demand for certain alternative solutions. This chapter discusses the test results obtained on two different test cases.

Section 5.1 describes the characteristics of the test cases. Section 5.2 discusses, based on obtained results, for each of the four defined goals in what extent the proposed method is able to obtain the requested alternatives. In this section, also modifications to the models based on the obtained results are discussed. Finally, Section 5.3 summarizes the main results of this chapter.

5.1 Test cases

To test the proposed models, we use two different test cases, both are based on studies performed for customers of ORTEC. From the first study, we define a small and a large test instance. The second test case originates also from a situation from practice. This section introduces the test cases and describes their characteristics.

5.1.1 Test cases Bread

The Bread test case originates from the Dutch retail market, and it concerns the production and transportation of bread. For the remainder of this thesis, we refer to this test case as TC Bread. Production currently happens at 8 different bakeries, located throughout the Netherlands. From these bakeries, the bread is distributed to 791 retail shops, also located throughout the Netherlands. The bakeries are modeled as production and distribution locations and the retail shops are considered as customer locations. In 5 years time, demand is expected to grow beyond the current production capacity. Therefore, strategic decisions have to be made about network configurations, to ensure that demand can be met in the future. There are two options to increase the total production capacity of the network. One is to open new production locations and the other is to invest in the production capacity of the current production locations.

All production locations are named by the country, NL for Netherlands or BE for Belgium, followed by three or four numbers. The current production facilities are all located in the Netherlands and are numbered NL001 to NL008. The potential new locations are numbered by their zip code, for example 'NL8601'. Appendix B displays the current and potential new locations graphically. For this test case, the fixed cost to use one of the current locations is

location	situation	capacity	fixed cost	variable prod. cost
NL001	current	8,241	€14,700	€6.70
NL001	closing	0	- €36,600	-
NL001	investment option 1	12,500	€37,050	€6.70
NL001	investment option 2	15,000	€61,200	€6.70
NL002	current	6,497	€11,200	€6.90
NL002	closing	0	- €27,900	-
NL002	investment option 1	9,600	€28,350	€6.90
NL002	investment option 2	11,000	€40,050	€6.90
NL2678	opening	3,821	€53,300	€4.30
NL2678	not opening	0	€0	-
BE2960	opening	10,450	€83,600	€7.50
BE2960	not opening	0	€0	-

Table 5.1: Example data for TC Bread on capacities and costs corresponding to opening, closing, or investing in a location.

significantly lower than the cost of opening a new production location. The reason is that all facilities are already in place for the current locations. Furthermore, closing one of these locations does not only save money from not having to pay some fixed costs, it also yields a fixed amount from selling off the building and facilities. Besides fixed cost for the use of a location and for making investments, also variable cost play a role; i.e., variable production cost for a production location and transport cost for distributing one product from a production location to a customer are given. Table 5.1 displays for two current and two new locations data on the cost and capacities corresponding to all options that exist for such a location.

BOSS calculates distribution cost for the transport of one product between any production location and any customer based on, among others, geographical data, truck capacity, and demand data. The cost of transporting one unit of a product from a production location to a customer varies between €0.50 and €5.00. There are some outliers for allocations that will not happen in practice, for example, the bakery near Amsterdam will not distribute to retail shops in Maastricht.

For testing purposes, we use a small and a large instance of this test case. For both instances, the described characteristics are as discussed above. However, for the small case a total of 10 potential new production locations are defined, whereas for the large case there are 30 potential new production locations. Also, whereas the small test case represents a realistic situation, we use the large test case to analyze how the methods behave in an extreme situation. Therefore, we increase demand by an extra 80% for the large test case. For the small case about 3 of the 10 potential new production locations should be opened to fulfill demand, for the large case about 10 of the 30 potential locations have to be opened. Furthermore, for both instances it is possible to close any of the current production locations. However, for the small case only 3 investment options are defined to increase the capacity of a current location, whereas for the large case 12 of such investment options are defined. Table 5.3 gives a summary of the characteristics of these test cases, together with the characteristics of the test case described in the next section.

5.1.2 Test case Oil and gas market

The Oil test case originates from the American oil and gas market; we refer to this case in the remainder of this thesis as TC Oil. This test case concerns the production and distribution of one product, which we simply call oil. The oil is produced at 3 production locations, located in the USA. It is distributed to 21 distribution locations which further distribute the oil to the 713 customers locations, also located in the USA. The production locations are named 'US-P001', 'US-P002', and 'US-P003', the distribution locations are named 'US-D' followed by a zip-code. The management believes that a large cost saving can be achieved by closing one or more distribution locations. Therefore a study is performed to make recommendations to the management team on the network configuration. For this case no investment decisions have to be made; the relevant decision is which distribution locations should be kept open and which should be closed. Appendix B gives a graphical overview of the production and distribution locations.

For this test case savings from closing a distribution location are caused by not having to pay a fixed amount, for example for renting the building. However, closing a location does not yield a fixed amount of money, as for TC Bread, from selling off facilities. Thus, the only fixed costs defined in this test case are for using a location. Furthermore, variable costs are associated with production, handling at a distribution location, transportation from the production locations to the distribution locations, and distributing the oil from the distribution locations to the customers. All fixed and variable costs differ per location. Table 5.2 gives data on capacities and costs of the three production locations and three of the 21 distribution locations. Note that since closing any of the production locations is not an option in the scope of this study, no fixed location cost are defined for these locations.

location	capacity	fixed cost	variable cost
US-P001	760,000	-	\$1.87
US-P002	330,000	-	\$2.64
US-P003	815,000	-	\$1.54
US-D5499	74,023	\$611,201	\$0.56
US-D6712	114,399	\$1,137,274	\$0.44
US-D6883	7,739	\$68,733	\$0.58

Table 5.2: Example data for TC Oil on capacities and costs corresponding to several locations.

Transport cost for one unit of oil from a production location to a distribution location varies between \$0.88 and \$3.90. For the transport of one unit of oil from a distribution location to a customer location, costs vary between \$0.46 and \$1.97. As already mentioned, Table 5.3 gives a summary of the characteristics of this test case. Note that even though all distribution locations can be closed in this test case, these closures are not defined as investment options, since closing a location does not result in a profit, only a saving from not having to pay fixed location costs.

5.2 Results of applying the methods

This section discusses for each defined goal the results of applying the corresponding proposed method. The GHSJ framework for BOSS is implemented in AIMMS 3.11 and all test results are obtained with CPLEX 12.3. For each method we start with running it for a few different values of the allowed cost deviation k. We want to find out whether the method works as expected and intended, and how many different solutions are obtained. If the first results give rise to it, we modify the method and test it again. Furthermore, we test for sensitivities of the methods, such as the chosen random seed.

	TC Bread (small)	TC Bread (large)	TC oil
Number of customers locations	791	791	713
Total demand	81,862.2	147,352.4	1,663,485
Average demand per customer	103.5	186.3	2,333.1
Number of current prod. locations	8	8	3
Total prod. capacity current locations	75,761.0	75,761.0	1,905,000
Average prod. capacity current locations	9,470.1	9,470.1	635,000
Number of potential new prod. locations	10	30	-
Total prod. capacity new locations	64,279.0	193,558.0	-
Average prod. capacity new locations	6,427.9	6,451.9	-
Number of current dist. locations	-	-	21
Total dist. capacity current locations	-	-	2,305,704
Average dist. capacity current locations	-	-	109,795.4
Number of potential closures of prod. loc.	8	8	0
Number of potential closures of dist. loc.	-	-	21 [1]
Number of potential investments to increase capacity	3	12	0
Total number of potential investments	11	20	0
Total potential increase in capacity through investments [2]	33,100.0	106,500.0	-
Total average increase in capacity per investment	11,033.3	12,383.3	-

[1] These closures are not modeled as investment options, since they do not come with a fixed profit.

[2] This amount does not equal the sum over all investments, since only one investment can be chosen per location. Therefore this amount equals the sum over all locations, of the investment that increases the capacity most.

Table 5.3: Summary of characteristics of test cases.

5.2.1 Obtain maximally different alternative solutions

To find a number of alternative solutions that are maximally different from each other, we apply the Standard HSJ method. In words, this method attempts to find a solution that uses as little as possible locations and investments that were used in any of the previous solutions. Since the restrictions of the original model have to be fulfilled, this potentially leads to the use of different locations and investments.

This section starts with discussing the number of obtained solutions and the diversity of the set of solutions for all test cases and different values of the allowed cost deviation k. Then we discuss the results of adapting the main two dependencies of the model, i.e., the initial solution and the value of the weight factor γ. Finally, we propose and test some modifications to the model to obtain more alternatives.

Obtained solutions

For each test case, we want to find out how many alternative solutions the Standard HSJ method finds for different values of the allowed cost deviation k. Furthermore, we like to know the diversity of the set of obtained solutions and whether there is a relation between the number of obtained solutions (# alt.), the diversity measure $D(S)$, and k. To test this, we run the Standard HSJ method for each test case for multiple values of k; Table 5.4 displays these results. Appendix C gives for TC Bread (small) with $k = 0.05$ and $k = 0.10$, and for TC Oil with $k = 0.05$ a detailed overview of the obtained results.

The results in Table 5.4 show that, especially for TC Bread (small) and TC Oil, the number of obtained alternative solutions is small. For these test cases for non of the choices of k, more than 3 alternatives are found. For TC Bread (large) at most 8 alternative solutions are obtained. Furthermore, the number of obtained solutions is not always increasing with the value of k. For example, for TC Bread (small) and $k = 0.01$, 2 alternatives are found, while only 1 alternative is found when k is increased to 0.02. The reason for this is that with a higher value of k, the Standard HSJ method may find a first alternative that differs by more decision variables from the initial solution, which results in a different starting position for the next iteration. It turns out that the method is from this starting position not able to set any of the decision variables in J to zero, such that the method is terminated. Note that the data in Table 5.4 does not give information on which solutions are obtained. For example, the 2 solutions obtained for TC Bread (small) with $k = 0.01$ do not include the solution obtained with $k = 0.005$, even though this is also an allowed solution. See also Appendix C, which shows that the set of solutions obtained for TC Bread (small) with $k = 0.05$ and $k = 0.10$ do not contain overlapping solutions.

From the results in Table 5.4, we see that when k is bigger, the value of the diversity measure $D(S)$ increases. The diversity measure is not only higher when the number of obtained solutions is higher, since Table 5.4 shows examples of an increasing value of $D(S)$, while the number of obtained alternatives decreases or remains equal.

Allowed cost	TC Bread (small)		TC Bread (large)		TC Oil	
increase k	# alt	$D(S)$	# alt	$D(S)$	# alt	$D(S)$
0.0001	0	-	1	0.0086	0	-
0.001	0	-	3	0.0453	0	-
0.005	1	0.0172	4	0.0579	1	0.0477
0.01	2	0.0383	4	0.0662	2	0.0529
0.02	1	0.0517	6	0.0725	3	0.1041
0.05	3	0.0905	8	0.0924	3	0.1072
0.10	3	0.1034	7	0.1099	3	0.1481

Table 5.4: Number of obtained alternative solutions with the Standard HSJ method and corresponding diversity measure $D(S)$.

When the Standard HSJ does not find any alternative solutions, we can conclude that no alternative solutions exist within the allowed cost range k. The reason for this is, that if the Standard HSJ method did not succeed in finding an alternative solution, then it is not possible to set any of the nonzero decision variables in the initial solution to zero. Also, we can safely assume that it is, due to the constraint on total cost, not possible to set any zero decision variable to 1, without simultaneously setting a nonzero decision variable to zero. Therefore, no alternative solution exists within cost bound k, if the Standard HSJ does not find any. Thus, for TC Bread (small) and TC Oil no alternative solutions exist for $k \leq 0.001$. It is useful information for the modeler to know that no alternative solutions exist within a certain cost bound k. In that case, he knows that alternatives are available only if he accepts an increase in cost higher than this value of k. If the Standard HSJ method obtains one or more solutions, we cannot conclude anything about the number of existing alternative solutions, besides that it is as least as high as the obtained number.

Effect of different initial solution

In the Standard HSJ method, the objective function that is optimized in each iteration is determined by the value of the decision variables in the previously obtained solutions. Therefore, we want to find out the extent to which the obtained solutions depend on the chosen initial

Start	sol. obtained in iteration 1	2	3
A	**B**	**C**	**D**
E	**F**	C	D
G	**H**	C	D
I	**J**	C	D
B	C	**M**	D

(a) TC Bread (small)

Start	sol. obtained in iteration 1	2	3
A	**B**	**C**	**D**
E	**F**	**G**	D
H	B	G	D
I	**J**	G	D
K	**L**	D	

(b) TC Oil

Start	solution obtained in iteration 1	2	3	4	5	6	7	8	9	10
A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**		
B	**J**	**K**	**L**	**M**	E	F	G	H	I	
N	**O**	**P**	**Q**	**R**	E	F	G	H	I	
S	**T**	**U**								
V	**W**	**X**	L	**Y**	R	E	F	G	H	I

(c) TC Bread (large)

Table 5.5: Obtained solutions with the Standard HSJ method when different starting solutions are used. A letter classifies a specific combination of decision variables and if a letter is bold then the solution is not obtained in any of the previous runs.

solution. For this experiment, we allow a 5% increase of costs ($k = 0.05$) and use 5 different initial solutions, under wich the optimal solution. All chosen initial solutions have total cost less than 5% higher than the optimal solution. There exist for TC Bread (small) at least 70 different solutions with a maximum cost increase of 5%, 315 for TC Bread (large), and 194 solutions for TC Oil, see Table 5.8. Each solution is classified by a letter that characterizes the exact combination of decision variables, where the letter 'A' refers to the optimal solution.

From the results of this experiment, displayed in Table 5.5, it is striking that for all test cases often the same solutions are obtained, while the number of available solutions that have a total cost at most 5% higher than the optimal solution is very high. For TC Bread (small) in every iteration the solutions C and D are obtained; for TC Oil a similar situation holds where D is always obtained and G in 3 of the 5 experiments; for TC Bread (large) E, F, G, H, and I are obtained for all except the fourth experiment. For this test case, it even holds that the solutions E to I are obtained in the same order. The results for TC Bread (large) show a recognizable pattern. Thus, it seems that there is a high probability of the method to converge to the same solutions. A tentative conclusion is that the obtained solutions when using different start solutions are often the same.

Effect of changing the value of γ

The objective function for the Standard HSJ can be split in two parts: a difference part and a cost part; i.e.,

$$\min \quad \underbrace{\sum_{l \in J_L} \mathit{LocOpen}\,(l) + \sum_{i \in J_I} \mathit{InvDone}\,(i)}_{\text{difference part}} \underbrace{- \gamma \cdot S}_{\text{cost part}}.$$

We want to draw conclusions on the effect of increasing the relative importance of the cost part of the objective function by increasing the value of γ. For previous tests, we have chosen γ such that the cost part of the objective function is always subordinate to the difference part, i.e., only if two solutions have equal score on the difference part, the solution with lowest cost is preferred, otherwise the solution with lowest score on the difference part is chosen. We define for the cost part

$$\gamma = \gamma' \cdot \frac{1}{k \cdot z_0}.$$

With this formulation we can better scale the relative importance of both parts by choosing the value of γ'. When $0 < \gamma' < 1$, it holds that $\gamma \cdot S < 1$. Since the difference part can only take integer values, such a choice of γ' ensures that the objective function is hierarchical, with the difference part as primary objective. Appendix D discusses this in more detail. We are interested in the obtained solutions for $\gamma' \geq 1$. Therefore, we set $k = 0.02$ and increase γ' step by step for each test case. We only draw conclusions from the solution obtained in the first iteration, since the optimization process for each iteration depends on the value of decision variables in previously obtained solutions.

For each choice of γ', Table 5.6 displays for the first obtained alternative the value of the binary difference metric (BDM) and the actual cost increase. The value of BDM gives the pairwise difference between the initial solution and the alternative solution for the decisions variables *LocOpen* (l) and *InvDone* (i). These results show, as expected, that when γ' increases, the value of BDM decreases, indicating a smaller difference between the first obtained alternative solution and the initial solution, while total cost of the alternative solution is also decreasing.

We conclude that setting $\gamma > 1$, such that a trade-off is made between degree of difference and costs of the solution, makes the process somewhat like a black box to the user. We cannot make any strong conclusions about the effect of weight factor γ' on this trade-off or give an indication on the value a user should choose for γ'. It is more transparent to the decision maker when the objectives are assumed to be hierarchical, where difference is the main objective. In that case, the user should choose $0 < \gamma' < 1$ and different solutions are obtained by using different values of k.

γ'	BDM	act. cost increase
$0 \leq \gamma' \leq 1$	6	1.71%
$\{2, 3\}$	4	0.65%
$\{4, \cdots, 17\}$	2	0.12%
$\{18, \cdots, \infty\}$	0	0.00%

(a) TC Bread (small)

γ'	BDM	act. cost increase
$0 \leq \gamma' \leq 3$	19	1.58%
$\{4, 5\}$	16	1.07%
6	14	0.67%
$\{7, \cdots, 14\}$	12	0.36%
$\{15, \cdots, 49\}$	9	0.09%
$\{50, \cdots, 574\}$	5	0.01%
$\{575, \cdots, \infty\}$	0	0.00%

(b) TC Bread (large)

γ'	BDM	act. cost increase
$0 \leq \gamma' \leq 1$	7	1.81%
$\{2, 19\}$	4	0.21%
$\{20, \cdots, \infty\}$	0	0.00%

(c) TC Oil

Table 5.6: Effect of different values of γ' on the difference and cost part of the objective function of the Standard HSJ method for (a) TC Bread (small), (b) TC Bread (large), and (c) TC Oil.

Modified Standard HSJ to obtain more alternatives

It is striking about the Standard HSJ method that it finds in most experiments only a very small number of alternative solutions. For example, for TC Bread (small) and $k = 0.02$, only 1 alternative solution is obtained, even though at least two more solutions qualify, namely the solutions obtained for $k = 0.01$. Therefore, we propose an adapted version of the Standard HSJ method, which should find more alternative solutions. However, these alternatives should still be maximally different from the current set of solutions. Therefore, also the diversity should not decrease compared to the results of the original Standard HSJ method. We test the adapted method by applying it to all test cases for k =0.01, 0.02, 0.05, and 0.10. Results can than be compared to the results displayed in Table 5.4. However, first we describe the proposed modification to the Standard HSJ model.

To prevent the Standard HSJ method from stopping after obtaining a solution that has been found before, we define a different objective function at this point. This new objective function is very similar to the original objective function of the Standard HSJ method, only now weights are added to the binary decision variables. Thus, the new objective function is:

$$\min \quad \sum_{l \in J_L} \alpha_L(l) \cdot \mathit{LocOpen}(l) + \sum_{i \in J_I} \alpha_I(i) \cdot \mathit{InvDone}(i) - \gamma \cdot S. \tag{5.1}$$

The weights $\alpha_L(l)$ and $\alpha_I(i)$ are set equal to the number of obtained solutions in which the corresponding decision variable is nonzero. By minimizing this new objective function, the highest possible weight is given to closing locations that were not closed in any of the previous solutions, while the weight associated with *LocOpen* (l) for a location l that was closed in all previous solutions still equals zero. Therefore, the new objective function maximizes the difference between the new solution and all previous obtained solutions. The weights are updated after each new solution, such that the objective is different in each iteration.

The described modifications of the Standard HSJ method ensure that in theory this method can be repeated for infinitely many iterations. Therefore, we introduce the same stopping criteria that are also used for the Random HSJ method and the Cost uncertain HSJ method, i.e., *MaxSol* and *MaxEq*. Figure 5.1 displays the complete proposed method, where *SolCount* is the number of obtained solutions including the initial solution and *EqCount* is the number of times a solution is obtained that is already in the set of solutions.

We apply the adapted version of the Standard HSJ method to all test cases, and set $MaxSol = 25$ and $MaxEq = 2$ or 5. Table 5.7 displays for different values of k the number of obtained solutions and the value of the diversity measure $D(S)$. For ease of comparison also the results for the original Standard HSJ are displayed. From these results we see that the number of obtained solutions can be increased significantly with the proposed modification. For TC Bread (small) we obtain the smallest increase, even no new alternative solutions for $k = 0.005$ and only one for $k = 0.05$, while at least for $k = 0.05$ we know that more solutions exist. This is caused by the fact that optimizing the new weighted objective function results in a solution that is already in the set of obtained solutions. We can explain this as follow: for instances in which the initial number of obtained solutions is small, the weights vary less and the resulting objective function differs less from the previous objective function. Therefore, the probability of obtaining the same optimal solution is larger. Furthermore, results show that also the diversity of the set of alternatives significantly increases by applying the adapted Standard HSJ method instead of the original Standard HSJ method.

We conclude that the adapted Standard HSJ method is able to obtain more alternative solutions, which also have a higher diversity than the solutions obtained with the original Standard HSJ method. Therefore, we recommend to use the adapted method. However, choosing $MaxEq = 5$ instead of $MaxEq = 2$ does not result in more solutions for most instances. Also, the diversity of the set of obtained solutions barely increases or even decreases by increasing $MaxEq$ from 2 to 5. Therefore, if time is an issue, $MaxEq = 2$ should be chosen. In that case, the method stops as soon as the modified method resulted in a solution that has been found in one of the previous iterations.

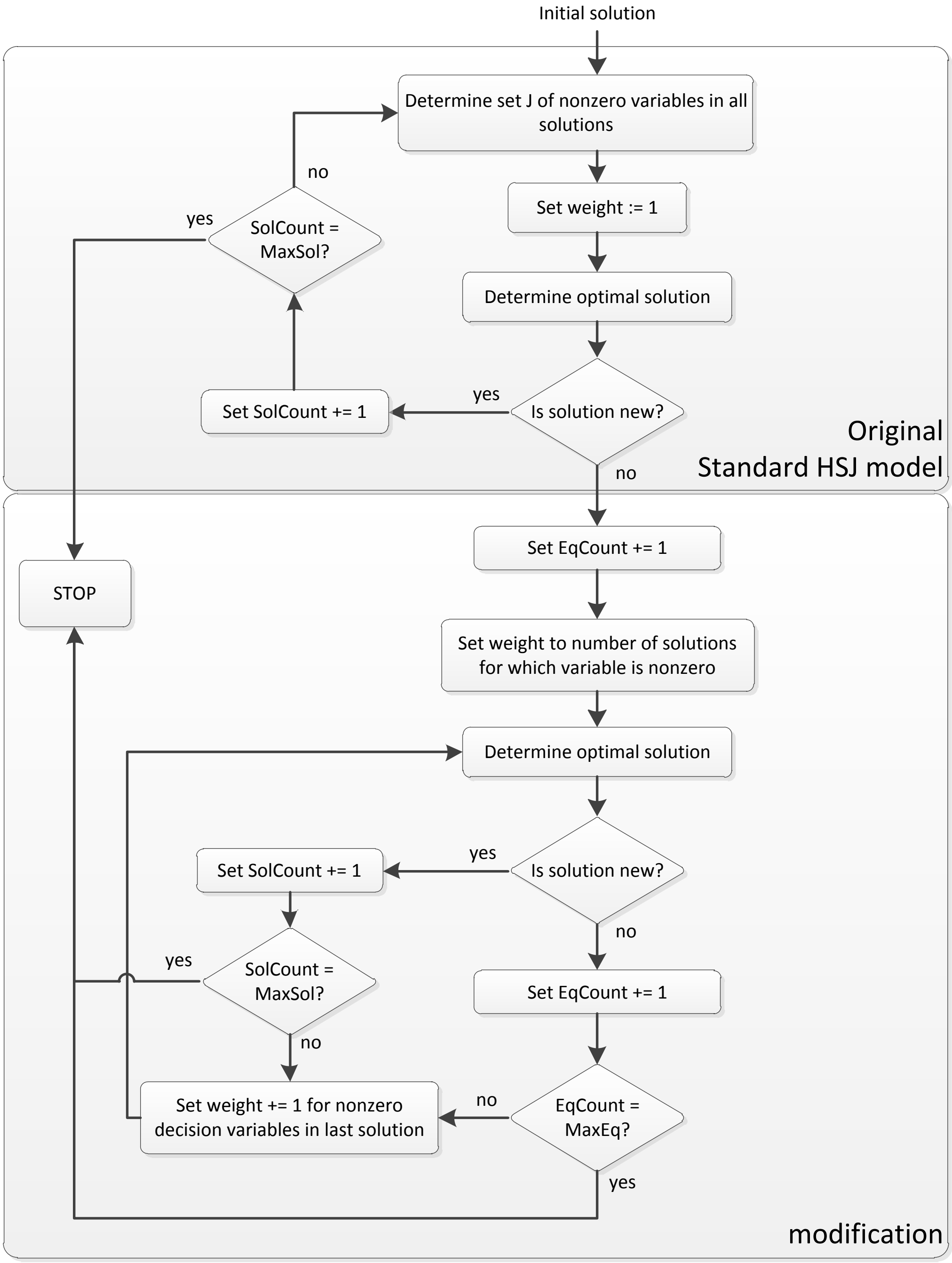


Figure 5.1: Modified Standard HSJ method to obtain more alternative solutions.

5.2.2 Obtain a large number of alternative solutions

To obtain a large number of alternative solutions, we apply the Random HSJ method. This method tries to find solutions with different decisions about the used locations and investments by generating for each iteration new random weights corresponding to these decisions and optimizing the resulting objective function.

This section first discusses the obtained number of alternatives for different values of the allowed cost deviation k and the corresponding value of the diversity measure $D(S)$. Second,

k	# obtained alternatives original	modified ($MaxEq = 2$)	modified ($MaxEq = 5$)	Diversity measure ($D(S)$) original	modified ($MaxEq = 2$)	modified ($MaxEq = 5$)
0.005	1	1	1	0.0172	0.0172	0.0172
0.01	2	4	4	0.0383	0.0524	0.0524
0.02	1	4	5	0.0517	0.0662	0.0642
0.05	3	3	4	0.0905	0.0905	0.0910
0.10	3	6	7	0.1034	0.1520	0.1530

(a) TC Bread (small)

0.0001	1	1	1	0.0086	0.0086	0.0086
0.001	3	5	6	0.0435	0.0517	0.0500
0.005	4	10	11	0.0579	0.0804	0.0800
0.01	4	21	23	0.0662	0.0946	0.0955
0.02	6	24	24	0.0725	0.1162	0.1162
0.05	8	24	24	0.0924	0.1481	0.1481
0.10	7	24	24	0.1099	0.1661	0.1661

(b) TC Bread (large)

0.005	1	4	5	0.0477	0.0648	0.0728
0.01	2	5	6	0.0529	0.0794	0.0797
0.02	3	6	8	0.1041	0.1166	0.1199
0.05	3	9	11	0.1072	0.1490	0.1491
0.10	3	9	11	0.1481	0.1662	0.1663

(c) TC Oil

Table 5.7: Results of adapted Standard HSJ for TC Bread (small) (a), TC Bread (large) (b), and TC Oil (c).

we discuss how we can decrease the number of obtained solutions, while potentially increasing the diversity of the set of solutions. Finally, we test how dependent obtained results are on the chosen random seed.

Obtained solutions

For each test case, we investigate how many alternative solutions the Random HSJ method finds for different values of the allowed cost deviation k. Therefore, we apply this method to all test cases for different values of k and set *MaxSol* sufficiently large, such that it is not restrictive. Furthermore, we choose $MaxEq = 600$ for TC Bread (small) and TC Oil, and due to large computation time $MaxEq = 10$ for TC Bread (large). To gain some first understanding of the influence of the applied random seed on the obtained solutions, this experiment is repeated twice for TC Bread (small).

Table 5.8 displays the results of these experiments. In this table, the column 'total # alternatives' gives for TC Bread (small) the total number of different obtained alternatives from both trials. For TC Bread (small) and TC Oil, no tests are performed for $k \leq 0.001$, since we concluded from the results of the Standard HSJ method that no alternatives exist. For TC Bread (large) and $k \geq 0.05$ no tests are performed, since already a large number of solutions is obtained and the process takes a significant amount of time. We see that the number of obtained solutions increases fast when k increases and can be come very large. Furthermore, the diversity measure is strictly increasing with k.

k	TC Bread (small)					TC Bread (large)		TC Oil	
	# alternatives		total #	Diversity ($D(S)$)		# alter-	diversity	# alter-	diversity
	(run 1)	(run 2)	alternatives	(run 1)	(run 2)	natives	$D(S)$	natives	$D(S)$
0.0001	-	-	0	-	-	1	0.0086	-	-
0.001	-	-	0	-	-	8	0.0485	-	-
0.005	1	1	1	0.0172	0.0172	49	0.0694	5	0.0728
0.01	4	4	4	0.0524	0.0524	112	0.0854	8	0.0764
0.02	13	12	14	0.0644	0.0641	315	0.1004	40	0.1141
0.05	54	66	70	0.1023	0.1045	-	-	194	0.1540
0.10	205	199	293	0.1387	0.1358	-	-	710	0.1674

Table 5.8: Number of obtained alternative solutions and corresponding diversity measure $D(S)$ of applying the Standard HSJ method.

Decreasing the number of solutions

Since, the total number of obtained solutions can easily become too high to be considered by a decision maker, we want to decrease the number of obtained solutions and potentially increase the diversity measure $D(S)$. There are two options to do this, i.e.,

1. introduce an extra stopping criterion;

2. remove solutions that differ too little from the current set of solutions.

An extra stopping criterion could be, besides using $MaxSol$ en $MaxEq$, to stop when the diversity of the set gets below a pre-specified value. However, it appears that $D(S)$ is not a decreasing function of the number of alternatives. For example, the diversity measure for TC Bread (large) and $k = 0.02$ equals 0.091 for the first 10 solutions and 0.100 for the entire set of 316 solutions. Therefore, using such a stopping criterion does not seem to be useful.

Removing solutions that differ too little from the current set of solutions can be done by using the binary difference measure or the capacity difference metric, discussed in Section 4.6. Appendix F displays for TC Bread (small) with $k = 0.10$ and for TC Bread (large) with $k = 0.02$ the value of the binary difference metric ($BDM(s)$) for each obtained solution. Now we present the results of applying the Random HSJ method with the binary difference metric to TC Bread (small) and $k = 0.10$. After each iteration, we calculate the value of $BDM\,(s)$ for the new obtained solution s. We remove solution s from the set of solutions if the value of $BDM\,(s)$ is smaller than a pre-specified value x. This means that a solution should differ by more than x binary variables from all current solutions to be included in the set of alternatives. We test this method with the same random seed and settings as used for the results of run 1 in Table 5.8 and set $x = 2$, 3, and 4. To be able to compare results, we terminate the method when the total number of removed solutions equals $MaxEq$, i.e., solutions that are double obtained and solutions that differ too little from the current set.

Table 5.9 displays for different values of x the number of obtained solutions and the diversity measure $D(S)$. For completeness, also the situation without removal is included, i.e., $x = 0$. From these results, we see that when a solution should differ by more than 2 variables from all current solutions, the number of obtained solutions almost halve and the diversity measure increases from 0.1387 to 0.3261. Also, increasing x further, decreases the number of obtained solutions significantly. However, increase of diversity is less significant when x is set higher than 2, compared to when x is set from 0 to 2. Thus, we conclude that using the binary difference metric to remove solutions from the set can decrease the number of obtained solutions significantly, while also increasing the diversity measure $D(S)$.

Removal from set for	# of alternatives	Diversity ($D(S)$)
No removal	205	0.1387
$BDM\,(s) \leq 2$	105	0.3261
$BDM\,(s) \leq 3$	56	0.3634
$BDM\,(s) \leq 4$	40	0.3974

Table 5.9: Results for TC Bread (small) and $k = 0.10$ of applying the binary difference metric (BDM) to the Random HSJ method.

Effect of different random seeds

We want to find out the extent to which the chosen random seed is determinant for the obtained solutions. From the results of TC Bread (small) in Table 5.8, it seems that about the same solutions are obtained with different random seeds. This is especially true for $k \leq 0.05$. To test if it holds in general that with different chosen random seeds roughly the same solutions are obtained, we select for all test cases a value of k such that around 10 different solutions are found. This means for TC Bread (small) $k = 0.02$, for TC Bread (large) $k = 0.001$, and for TC Oil $k = 0.01$. Furthermore, we set $MaxEq = 100$ for TC Bread (small) and TC Oil, and $MaxEq = 20$ for TC Bread (large).

Table 5.10 displays the results of the discussed experiment for TC Bread (large) and TC Oil. The numbers in between brackets display the iteration in which the solution is obtained. Results for TC Bread (small) show a similar pattern to results for TC Bread (large) and are therefore displayed in Appendix G. For example, for TC Bread (large) in the first run, solution B is obtained in the first iteration, solution C in the second, and in the third iteration A, B, or C is obtained again and thus it is removed from the set of solutions. For this test case, we see that the solutions C, D, E, and G are obtained in each run within 10 iterations, whereas F is only obtained in 2 of the 5 runs and K only in 1 run. Also, for TC Bread (small), the first few obtained solutions in the first run are mostly also found in the other runs. For TC Oil the variation of the iterations in which solutions are found is much bigger. However, within 25 iterations almost all solutions of the first run are also obtained in the other four runs.

The results from TC Bread indicate that at least a significant part of the obtained solutions is also found with another seed, as long as $MaxEq$ is not set too low. The results for TC Oil are less convincing, since more fluctuation in obtained solutions is present and much more iterations are needed to obtain approximately the same solutions as in the first run. Therefore, we cannot make strict conclusions on the dependency of the random seed based on the observed results. However, when $MaxSol$ and $MaxEq$ are chosen sufficiently large, approximately the same solutions are found and we can safely assume that the influence of the chosen random seed is not too significant.

5.2.3 Obtain alternatives that are robust against cost parameter uncertainties

To obtain alternative solutions that have lower cost when actual cost parameters are realized different than expected, we apply the Cost uncertain HSJ method. In this method the objective function is equivalent to the original objective function of BOSS. However, uncertain cost parameters are replaced by random numbers from a range indicated by the modeler. First, we discuss the results of applying the Cost uncertain HSJ method. We are especially interested in whether we find alternatives that are from an uncertain cost perspective preferred over the initial solution. Second, we briefly discuss the difference in obtained results between TC Bread and TC Oil.

initial solution	solution obtained in iteration 1	2	3	4	5	6	7	8
A (0)	**B (1)**	**C (2)**	**D (4)**	**E (6)**	**F (7)**	**G (8)**	**H (12)**	
A (0)	B (1)	G (2)	D (3)	E (4)	C (8)	H (13)	**I (25)**	
A (0)	E (1)	G (6)	C (7)	B (8)	**J (9)**	D (10)	I (11)	**K (22)**
A (0)	E (1)	D (3)	C (4)	I (6)	G (9)	B (22)		
A (0)	B (1)	E (2)	D (4)	G (6)	C (7)	H (12)	K (14)	F (21)

(a) TC Bread (Large)

initial sol.	solution obtained in iteration 1	2	3	4	5	6	7	8	9	10	11
A (0)	**B (1)**	**C (3)**	**D (4)**	**E (6)**	**F (7)**	**G (8)**	**H (12)**	**I (9)**	**J (20)**	**K (24)**	
A (0)	J (1)	C (2)	**L (3)**	I (4)	F (6)	K (8)	H (10)	G (21)			
A (0)	H (1)	K (2)	J (3)	C (6)	G (7)	I (9)	B (10)	F (17)	E (84)		
A (0)	B (1)	G (2)	J (4)	I (5)	K (6)	C (9)	F (10)	**M (50)**	H (54)	E (66)	D (70)
A (0)	I (1)	J (2)	H (3)	K (4)	G (5)	F (13)	C (17)	B (25)	E (38)	D (44)	

(b) TC Oil

Table 5.10: Results of using different random seeds in the Random HSJ for TC Bread (large) (a) and TC Oil (b). A letter classifies a specific combination of decision variables and if a letter is bold, then the solution is not obtained in any of the previous runs.

Obtained solutions

For each of the three test instances, we investigate whether the Cost uncertain HSJ method finds alternative solutions that are preferred over the initial solution. In order to achieve this, we define for each test case several scenarios of the uncertain cost parameters. To compare solutions with each other, the cost of each solution is calculated for a large number of realizations of the cost parameters. Subsequently, based on this data, we calculate for each obtained solution two performance indicators:

- average cost over all evaluated realizations;
- percentage of realizations in which the obtained solution has lowest cost in comparison to all other obtained solutions.

For the scenarios, we make a distinction between variable and fixed cost. For each scenario we define a variance for both types of cost. For a realization we draw the cost uniformly from the interval

$$\left[E\left[cost\right] \cdot (1 - variance)\,,\, E\left[cost\right] \cdot (1 + variance)\right]. \tag{5.2}$$

Cost parameters used in the objective function are also drawn from this same interval. Note that the initial solution is optimal if realized cost $= E[cost]$. For all tests we use the following settings: $k = 0.10$, $MaxSol = 100$, $MaxEq = 100$, and number of realizations for evaluation is 2000.

For each test case we define 4 different scenarios with increasing variance. Furthermore, variance of variable cost is for all scenarios smaller or equal to variance of fixed cost. For TC Bread (large) and TC Oil we use the same scenarios. For TC Bread (small) we choose scenarios with larger variances, since results for the scenario with smallest variance shows that this variance barely influences the optimal solution. Table 5.11 displays the results of these experiments. From the result for TC Bread (small) and TC Bread (large) we see that the initial solution in most experiments has the highest percentage of realizations with lowest cost. Even

sce-nario	variance cost fixed	variable	# obt. alt.	$D(S)$	best sol.	% this sol. lowest cost	% initial sol. lowest cost	cost compared to initial sol. orig. situation	average
1	0.05	0.05	3	0.0383	initial	80.80%	80.80%	$\approx$ 0.00%	$\approx$ 0.00%
2	0.10	0.10	3	0.0383	initial	74.90%	74.90%	$\approx$ 0.00%	$\approx$ 0.00%
3	0.25	0.10	29	0.0996	initial	33.70%	33.70%	$\approx$ 0.00%	$\approx$ 0.00%
4	0.80	0.50	100	0.1507	initial	3.40%	3.40%	$\approx$ 0.00%	$\approx$ 0.00%

(a) TC Bread (small)

sce-nario	variance cost fixed	variable	# obt. alt.	$D(S)$	best sol.	% this sol. lowest cost	% initial sol. lowest cost	cost compared to initial sol. orig. situation	average
1	0.05	0.05	100	0.0768	B	6.65%	6.00%	+0.01%	$\approx$ 0.00%
2	0.10	0.05	100	0.0905	C	4.75%	4.60%	+0.01%	$\approx$ 0.00%
3	0.10	0.10	100	0.0897	D	4.20%	2.80%	+0.14%	$\approx$ 0.00%
4	0.25	0.10	100	0.1216	initial	6.20%	6.20%	$\approx$ 0.00%	$\approx$ 0.00%

(b) TC Bread (large)

sce-nario	variance cost fixed	variable	# obt. alt.	$D(S)$	best sol.	% this sol. lowest cost	% initial sol. lowest cost	cost compared to initial sol. orig. situation	average
1	0.05	0.05	3	0.0463	B	94.85%	0.00%	+2.75%	-2.28%
2	0.10	0.05	7	0.0729	B	48.10%	4.65%	+2.75%	-1.00%
3	0.10	0.10	9	0.0772	B	39.55%	9.05%	+2.75%	-0.72%
4	0.25	0.10	28	0.1029	C	12.75%	3.35%	+0.64%	-0.64%

(c) TC Oil

Table 5.11: Results of applying the Cost uncertain HSJ method for different scenarios, with $MaxSol = 100$, $MaxEq = 100$ for TC Bread (small) (a), TC Bread (large) (b), and TC Oil (c).

if this does not hold, the initial solution is still relatively good, i.e., it has the lowest cost in almost as many realizations as the solution with highest score. Even more important, in all experiments on these two test cases, the initial solution has the lowest average cost. Also, the graphical representation of both performance indicators in Appendix H shows that the initial solution for TC Bread always belongs to the minority of solutions that have approximately the same percentage of realizations for which it has lowest cost as the best solution. Thus, for TC Bread the Cost uncertain HSJ method does not find a good alternative solution that outperforms the initial solution on both performance indicators. Therefore, there is no strong incentive to prefer one of the alternatives over the initial solution.

For TC Oil, the results are very different from those for TC Bread. From the results in Table 5.11, we see that for each scenario the Cost uncertain HSJ method finds an alternative which has lowest average cost in a higher percentage of the realizations. This table also shows that the solution in the column 'best sol.' has total cost in the initial situation only slightly higher than the optimal solution in this situation and lowest average cost. These results show that the obtained solutions B and C in the corresponding scenarios are preferred over the initial solution, since they have lower average cost and lowest cost in a higher percentage of realizations for the corresponding scenario.

To test whether the obtained results in Table 5.11 are representative or that completely different results are obtained with other random seeds, we repeat for TC Bread (small) and TC Oil for two scenarios the performed test with different random seeds. From the results of these tests, displayed in Appendix I, we conclude that the outcomes are fairly stable and no other conclusions are drawn when another random seed is used.

Difference between test cases

We conclude that for TC Bread, the Cost uncertain HSJ method did not result in an alternative solution that is preferred over the initial solution. However, for TC Oil, such an alternative is obtained for each scenario. Therefore, the Cost uncertain HSJ method is a good addition to BOSS. The fact that preferred alternatives are obtained for TC Oil and not for TC Bread, might be caused by the difference in cost structure of the test cases. An explanation could be in the difference of the fraction of fixed cost from total cost, this fraction is in the initial solution 0.23 for TC Bread (small), 0.37 for TC Bread (large), and 0.58 for TC Oil. Thus for TC Oil this fraction is significantly larger than for the other two test cases and this may cause relevant alternatives to be available for TC Oil and not for TC Bread. The reason is that total cost is only build up of a small number of fixed costs elements, while there is a very large number of variable cost elements. Variation in a large number of variable cost elements may cause effects to cancel each other out, while variation in only a small number of large cost factors may not have this effect. By this reasoning, a higher fraction of fixed cost makes it more likely to obtain alternative solutions that outperform the initial solution in other cost realizations. To be able to draw conclusions on what kind of cost structure characteristics make it likely to obtain relevant alternative solutions with the Cost uncertain HSJ method, more tests on different test cases should be performed.

5.2.4 Obtain alternatives that are robust against right-hand side uncertainties

To obtain solutions that are more robust against right-hand side uncertainties, we propose in Section 4.4.4 two versions of the Robust HSJ method. Right-hand side uncertainties for BOSS contain uncertainty about demand, production capacity, and distribution capacity. Both versions of the Robust HSJ method maximize the difference between available and used capacity. However, the Robust HSJ method (version 1) does this for individual production facilities and locations, whereas, the Robust HSJ method (version 2) maximizes the minimum production and distribution slack.

For TC Bread, only production capacity is relevant, since products are distributed directly from the production facility to the customer, without the intermediate step of a distribution location. For TC Oil, both production capacity and distribution capacity play a role. Thus, for this test case, the objective is to maximize both production and distribution slack.

This section first discusses the results of applying the Robust HSJ method (version 1) to both test cases and subsequently for the Robust HSJ method (version 2). For this last method we also propose some modifications and discuss results of applying the adapted method.

Robust HSJ method (version 1)

We want to find out whether the Robust HSJ method (version 1) is able to obtain a number of alternatives that distribute slack differently over the production facilities and distribution locations, while at the same time potentially increasing the total slack in the network. The Robust HSJ method (version 1) tries to do this by giving a higher incentive to increasing production or distribution slack for locations and facilities that are highly utilized in the previous solution than for less utilized locations and facilities. To test the Robust HSJ method (version 1), we apply it to all three test instances with different values of k.

Table 5.12 displays for each test case for the obtained alternative solutions the total slack and the actual cost increase; Appendix J gives more detailed results on the slack per location in each solution. For TC Oil, production slack is not displayed, since it is in each solution exactly

k	1st obtained alternative Total slack	actual cost increase	2nd obtained alternative Total slack	actual cost increase
0.00	40.8	-	-	-
0.01	1484.8	0.21%	-	-
0.02	1484.8	0.21%	1487.8	1.85%
0.05	4701.8	4.65%	-	-
0.07	7402.8	6.55%	-	-
0.10	9365.8	10.00%	8963.8	9.97%

(a) TC Bread (small).

k	1st obtained alternative Total slack	cost incr.	2nd obtained alternative Total slack	cost increase	3rd obtained alternative Total slack	cost increase
0.00	20.0	-	-	-	-	-
0.01	2784.6	0.98%	-	-	-	-
0.02	5563.6	1.98%	5488.0	1.96%	-	-
0.05	13407.6	5.00%	12451.6	5.00%	-	-
0.07	18516.6	7.00%	17191.6	6.99%	-	-
0.10	25852.6	10.00%	23701.6	10.00%	26939.6	9.99%

(b) TC Bread (large).

k	1st obtained alternative Total slack	cost incr.	2nd obtained alternative Total slack	cost increase	3rd obtained alternative Total slack	cost increase
0.00	15100	-	-	-	-	-
0.01	8093	0.89%	7083	0.92%	13140	0.86%
0.02	5122	1.81%	3324	1.63%	-	-
0.05	14822	5.00%	6511	3.31%	10169	4.02%
0.07	44335	5.72%	4727	2.05%	34560	5.17%
0.10	9660	8.41%	-	-	-	-

(c) TC Oil.

Table 5.12: Results of Robust HSJ method (version 1).

equally distributed over the locations. Apparently, the distribution of production slack has little influence on the total cost and even with a small allowed cost increase, total production slack can be equally allocated to the production locations. Thus, for TC Bread, in Table 5.12 slack refers to production slack and for TC Oil, it refers to distribution slack. The first line for each test case gives the situation of the initial solution. From these results, we see that the Robust HSJ method (version 1) only finds a small number of alternative solutions, mostly one or two alternatives. We can explain this in the following way: the weights $\beta_{PS}(f)$ and $\beta_{FS}(p,l)$ are calculated by the utilization of the previously obtained solution. However, if slack is relatively small, all weights are approximately equal. For example, for TC Bread (small) and $k = 0.01$ the first alternative only has slack of 1484.8 at one production location, while all others are fully utilized. Thus, the weight $\beta_{PS}(f) = 1$ for all except one location, and for this last location $\beta_{PS}(f) = 0.87$. These values of the weights result in the objective function (4.20) to be roughly the same in each iteration, such that with high probability a solution is obtained that is already in the set of alternative solutions. In that case, no new weights are calculated and the method terminates.

The results in Table 5.12 show that for TC Bread total slack increases significantly for all alternatives when k increases. For example, when $k = 0.01$, slack increases for TC Bread (small) from 0.00% for the initial solution to 1.81% of total demand for the alternative solution. For TC

Oil, total slack does not increase for most values of k. However, slack is distributed differently over the locations.

Even though we only find a small number of alternative solutions, these solutions can contribute to the decision process of choosing the solution that best fits the preferences of the decision maker. The Robust HSJ method (version 1) can give insight in possibilities to increase slack at different production facilities and distribution locations. Therefore, we recommend to apply this method if the decision maker prefers to increase production or distribution slack or is interested in options to distribute slack across the network.

Robust HSJ method (version 2)

We want to find out whether the Robust HSJ method (version 2) is able to increase minimum slack by increasing the value of k. And if this is the case, whether there is a straightforward relation between k and the minimum slack. To test this, we apply the Robust HSJ method (version 2) to all three test instances for $k \in \{0.01, 0.02, \ldots, 0.10\}$. Since, the objective function of this method is static, i.e., it does not change after an iteration, the method finds at most 1 alternative solution.

Figure 5.2 displays for all three test cases as a function of k, the minimum slack. This slack is displayed as a fraction of total demand by all customers, to make the different test cases more comparable. For TC Bread, slack refers to production slack and for TC Oil it refers to distribution slack. For TC Oil, production slack is not displayed, since minimum production slack is for all tested values of k equal to 4.84% of total demand, i.e., even for $k = 0.01$ total production slack is equally distributed over the 3 production facilities. For TC Bread (small) we see that no alternative solution is found for $k \leq 0.05$, i.e., minimum slack cannot be made positive if cost is allowed to increase by at most 5%. For TC Bread (large) it seems that slack linearly increases with k. However, this observation is not supported by the results of the other two test cases. For TC Oil minimum slack equals 0.47% of demand for all values of k between 0.07 and 0.10, and actual cost increase is 6.07%. Thus, minimum slack cannot be increased further than this 0.47% when cost may not increase by more than 10%, and therefore, the method returns the same solution for $0.07 \leq k \leq 0.10$.

The Robust HSJ method (version 2) has as a disadvantage that it is hard to interpret its results. Without context it tells very little to know that each location can produce 100 units

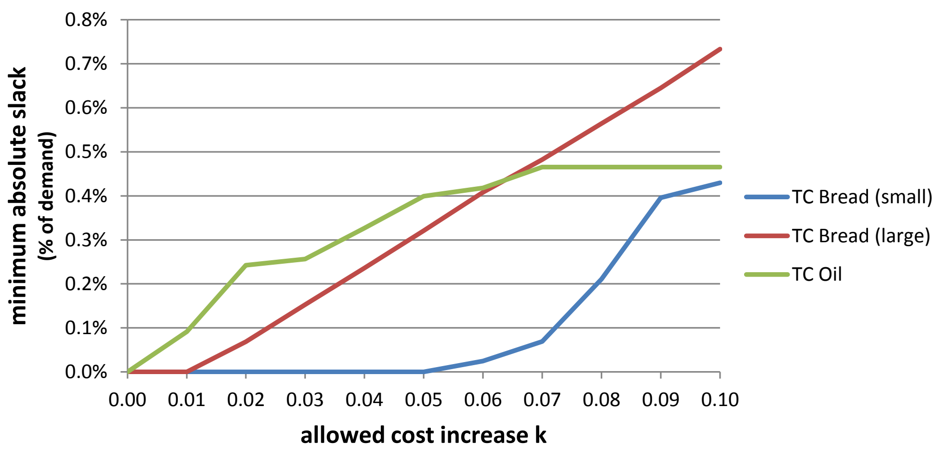


Figure 5.2: Results of Robust HSJ method (version 2), where minimum slack refers to production slack for TC Bread and to distribution slack for TC Oil. Slack is displayed as a percentage of total demand.

more. This is a very high amount, most likely even undesirable, when total capacity is 400 units, whereas, it is negligible small if total capacity is 100,000 units. Thus, we like to measure slack as a fraction of total capacity of the corresponding location or facility. We modify the Robust HSJ method (version 2) such that it maximizes minimum relative slack instead of minimum absolute slack. To this end, we replace some of the constraints, concerning slack, of the BOSS model. First, we introduce the following two new decision variables:

- *RelSmallestProdSlack*: smallest unutilized production capacity as a fraction of total capacity, over all production facilities;
- *RelSmallestFlowSlack*(p): smallest unutilized distribution capacity for product p as a fraction of total capacity, over all distribution locations.

Furthermore, we introduce the following two constraints to ensure that these two decision variables are indeed set to the corresponding values:

$$RelSmallestProdSlack \leq \frac{ProdSlack\,(f)}{prodcap\,(f)}; \tag{5.3}$$

$$RelSmallestFlowSlack\,(p) \leq \frac{FlowSlack\,(p,l)}{flowcap\,(p,l)} \qquad \forall\, p \in \mathcal{P}. \tag{5.4}$$

The value of the decision variables $ProdSlack\,(f)$ and $FlowSlack\,(p,l)$ follows from respectively the constraints (4.9) and (4.13). For the Robust HSJ method (version 1), we included constraints which ensure that slack equals zero for locations that are not in use. For this model, slack for a location that is not opened can be positive. Otherwise, *RelSmallestProdSlack* is equal to zero if at least one location is closed. The new objective function now becomes:

$$\max \quad RelSmallestProdSlack + \sum_{p \in \mathcal{P}} RelSmallestFlowSlack\,(p). \tag{5.5}$$

Note that the relative slack is defined as a fraction of the initial capacity, i.e., we do not account for investments that can increase the total capacity of a facility or location. The reason is that this makes the model nonlinear, such that it becomes much harder to solve.

Figure 5.3 displays for all three test instances, as a function of k, the minimum relative slack. The results in this figure show a similar pattern to that in Figure 5.2, i.e., the relation between k and minimum relative slack seems to be equivalent to the relation between k and minimum absolute slack. Minimum relative slack for TC Bread (small) is equal to zero for $k \leq 0.05$. For TC Bread (large), minimum relative slack almost linearly increases for $k \geq 0.01$. However, for TC Oil, the minimum relative slack increases for $k \geq 0.07$, whereas, the minimum absolute slack has the same value for $0.07 \leq k \leq 0.10$.

We now briefly discuss the impact of defining relative slack as a fraction of initial capacity instead of actual capacity. This definition biases the results, as actual relative slack is smaller than the calculated relative slack, if an investment is made to increase capacity. For TC Oil this is not an issue, since no investments are defined for this test case. For TC Bread (small) at most 1 location has a lower actual relative slack. For TC Bread (large) up to 3 locations have a lower actual relative slack. For example, for $k = 0.07$ *RelSmallestProdSlack* =2.73%, while actual relative slack is 1.50%, 1.61%, and 2.18% for 3 locations; all other locations have actual slack at least equal to 2.73%. However, the biasing is barely an issue and mostly a matter of definition.

We conclude that this adapted version using relative slack is preferred over the initial Robust HSJ method (version 2), which uses absolute slack. The reason is that relative slack is more intuitively understood and more relevant for a decision maker to steer on. Locations

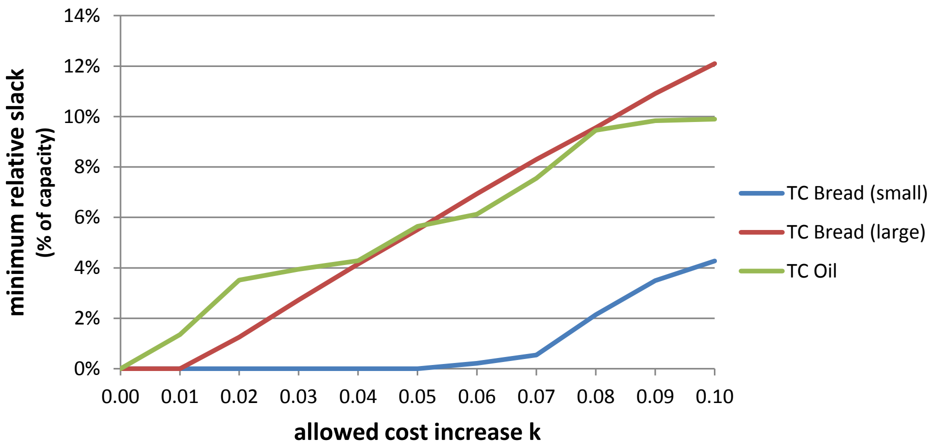


Figure 5.3: Results of modified Robust HSJ method (version 2), where relative minimum slack refers to production slack for TC Bread (small) and TC Bread (large) and to distribution slack for TC Oil. Slack is displayed as a percentage of total capacity of a location or facility.

can vary significantly in size, such that a slack of x units has a completely different effect on operations for a small facility as for a much larger facility. In contrast, if for example, the minimum relative production slack in a network is 2%, than we know that an increase of demand by 2% can be absorbed by the network. Also, every location produces at most at 98% of maximum production level. This may be an aspiration of the management team, such that failures and maintaince of machines do not disrupt the production process.

5.3 Conclusion

In this chapter, we have described the application to three test instances of four different realizations of the GHSJ framework, where each method was developed for a specific goal. We now summarize the main results and conclusions from these methods.

The Standard HSJ method was found to be well able to obtain maximally different solutions for all three test cases. However, the user has little control over the number of obtained solutions. Therefore, a modification to this model was proposed. From test results, we concluded that the number of obtained solutions is increased significantly with this modified method, while the diversity of the set of obtained solutions also increases.

The Random HSJ method can be used to obtain a large number of alternative solutions. Also, when only a very small cost deviation is allowed, this method can be used to find a number of alternatives. However, this method has no driving force to obtain solutions that are different from each other. To increase the diversity of the obtained solutions and decrease the number of alternatives, the binary difference metric ($BDM(s)$) can be used to remove solutions that differ not enough from the current set of solutions.

The Cost uncertain HSJ method was proposed to obtain solutions that are more robust against cost parameter uncertainties than the initial solution. Obtained alternatives were evaluated by the average cost over a large number of different realizations of the uncertain cost parameters and the percentage of realizations in which the solution has lowest cost. By the application of this method to the test cases and the evaluation of the obtained solutions, we found for one test case, alternative solutions that are preferred over the initial solutions, while for the other test case we did not find such alternatives. The difference between the results for both test cases may be caused by a difference in cost structure. However, without further research we cannot state this with certainty. We conclude that the Cost uncertain HSJ is a good addition to BOSS, since the method found relevant alternatives for one of the test cases.

The Robust HSJ method has two versions, with related objectives. Both methods are used to obtain solutions that are more robust against right-hand side uncertainties, such as uncertainty about demand or capacity. We found, by applying the Robust HSJ method (version 1), that a number of alternative solutions are obtained that allocate production or distribution slack differently over the facilities or locations. Even though the number of obtained solutions is small and most facilities and locations still use full capacity, results are useful and promising, i.e., with a small cost increase alternatives are found that a decision maker can consider when uncertainty in demand or capacity plays a role. By the Robust HSJ method (version 2), a single alternative solution, that maximizes minimum absolute slack is obtained. Even though this method shows some good results, it is hard to interpret the results. Therefore, we propose to adapt the method such that it maximizes minimum relative slack. From the results of applying both versions to both test cases, we conclude that the adapted version is preferred due to the more intuitive interpretation of results.

Chapter 6

Conclusion and recommendations

In this chapter we present the conclusions of this research project in Section 6.1. In section 6.2, we recommend on implementation of the discussed methods in BOSS. Finally, we gives some suggestions on further research in Section 6.3.

6.1 Conclusions

The subject of this thesis is BOSS, a strategic and tactical supply chain optimization tool, which solves a mathematical program to find an optimal supply chain configuration. We focus on the modeling of the location-allocation problem in BOSS; in this type of problem the most important decisions are which locations should be used and which investments should be made. Chapter 2 describes the mathematical formulation of this problem. For BOSS we identify two types of uncertainties that play a role:

- Data uncertainty, i.e., uncertainty about, among others, actual demand, capacities, and costs.
- Model uncertainty, i.e., uncertainty about whether the model accurately describes the realistic situation.

We have done literature research on dealing with both types of uncertainty in mathematical programming. Literature on robustness in mathematical optimization defines two types of data uncertainty, i.e., uncertainty about objective function parameters and uncertainty about right-hand side parameters. From the methods discussed in literature to obtain robust solutions, only fuzzy mathematical programming can handle both types of uncertainties. Since this method, in contrast to many other methods, does not introduce nonlinearity into the model, we expect it to be useful for dealing with data uncertainty in BOSS. However, we have decided not to implement this method. Instead we focus on methods to deal with model uncertainty and on adapting these methods such that they can also deal with data uncertainty.

Literature on model uncertainty proposes to generate alternative solutions that are good with respect to the modeled objectives and different from each other in the critical decisions taken. When presented with these different alternative solutions, the decision maker can decide which solution best fits his preferences with respect to both modeled and unmodeled objectives. Most of the methods discussed in literature generate such alternative solutions by applying mathematical optimization to the original problem, with a different objective function compared to in the initial model. We define a framework, which we refer to as Generalized Hop, Skip, Jump (GHSJ), to generate alternative solutions. In this framework, the objective function has a general form, but the implementation can be chosen based on the objective of the decision maker with respect to the obtained alternatives. Furthermore, we use the constraints of the original model to ensure that an obtained solution fulfills all requirements. Finally, we add an extra constraint to ensure that total cost of an alternative solution does not deviate more than a pre-specified amount from the cost of the optimal solution.

Based on the GHSJ framework, we propose two different methods to generate alternative solutions to the location-allocation problem in BOSS that make different critical decisions. First, we propose the Standard HSJ method to obtain a number of maximally different alternative solutions. This method uses an objective function that minimizes the sum of the binary decision variables that were nonzero in any of the previously obtained solutions. Second, we propose to apply an objective function with randomly generated weights to find a large number of alternative solutions; we refer to this method as Random HSJ.

We extend the applicability of the GHSJ framework to account for data uncertainty in BOSS. To this end, we define two methods to generate solutions that are more robust against data uncertainty. The Cost uncertain HSJ method aims to find alternative solutions that are more robust against cost parameter uncertainty than the initial solution. This is done by using the original objective function, where uncertain cost parameters are replaced by random realizations. The Robust HSJ aims to find alternative solutions that are more robust against right-hand side uncertainties. To this end, we introduce slack variables measuring for each location and facility the difference between available and used capacity. We propose two versions of the Robust HSJ, where the first one maximizes individual slack and the second maximizes minimum slack.

Chapter 5 discusses the application of the four proposed methods to two test cases based on studies performed for customers of ORTEC. These tests show that each proposed method is indeed able to generate the required alternatives. For the Standard HSJ method, we propose and test a modification that provides the user with more control over the number of obtained solutions, since this number can be very small with the original Standard HSJ. The Random HSJ can obtain a very large number of alternatives. Since this method has no driving force to obtain solutions that are different from each other, we apply the binary difference metric ($BDM(s)$) to remove solutions that differ too little from the current set of solutions. Tests show that this indeed results in a significant decrease of the number of solutions, while the diversity of the set of solutions increases significantly. Solutions obtained with the Cost uncertain HSJ method are evaluated based on a large number of realizations of the uncertain cost parameters. This method is a good addition to BOSS, since we found for one of the test cases alternative solutions that perform better in case of cost parameter uncertainty. For the Robust HSJ method we test two versions. Version 1 was found to obtain a number of alternative solutions that allocate production or distribution slack over different locations. For version 2 we propose to adapt the method to maximize minimum relative slack instead of minimum absolute slack, since results are in that case more intuitively interpreted.

6.2 Recommendations

We recommend ORTEC to implement all four discussed methods in BOSS. Note that for the Standard and the Robust HSJ method (version 2) we advise to implement the modified methods, since they find more or better alternatives. We now describe the recommendations for each of the four developed methods separately.

1. Standard HSJ

When doing a strategic network study for a customer with BOSS, we advise ORTEC to always use the Standard HSJ method. This method takes less time than the Random HSJ method and finds a small number of maximally different solutions. The Standard HSJ method can be used for a number of different values of k, for example 0.01, 0.02, 0.05, and 0.10. Obtained solutions can then be discussed with the customer, such that his preferences are better understood. Based on these results, ORTEC can either adapt the original model to include certain preferences or recommend one of the alternative solutions. The Standard HSJ gives insight to the decision maker in the existence of solutions that are close to optimal, but make very different strategic decisions.

2. Random HSJ

We recommend to use the Random HSJ method when cost may only increase from optimality by a small percentage, for example 0.5%, and the Standard HSJ method finds too few solutions. In that case, the Random HSJ may be able to obtain more and different alternatives than the Standard HSJ finds, especially when $MaxEq$ is set to a high value. Even for a higher allowed cost deviation it might be that more alternative solutions are requested. Also, in that case the Random HSJ method should be used. Furthermore, when a set of solutions with a very high diversity is required, we recommend to use the Random HSJ method with removal of solutions that differ too little by the binary difference metric.

3. Cost uncertain HSJ

When cost uncertainty plays a role, even if there is only a slight presumption that actual cost may be different than expected, we recommend to apply the Cost uncertain HSJ method. Whether this method finds solutions that are preferred over the initial solution for cost uncertain parameters, depends on the structure of the data. However, we cannot conclude beforehand whether this method finds relevant alternatives. Therefore, it is best to apply the Cost uncertain HSJ method for a small number of iterations and conclude from these results whether the method finds good alternatives, or that the initial solution is already good. Solutions should be evaluated based on a large number of randomly generated realizations of the uncertain cost parameters, and compared on average cost and percentage of realizations in which the solution has lowest cost. When this method does not find alternatives that score better on any of these criteria, this gives the decision maker more confident in the initial solution. So also in that case, it is relevant to apply the Cost uncertain HSJ method.

4. Robust HSJ

Finally, we recommend to apply the Robust HSJ method when the customer indicates that he is not sure about future demand or capacity. The Robust HSJ method (version 1) can be applied to find alternative solutions that allocate unused capacity differently over production facilities and distribution locations. These alternative solutions also have potentially more total unused capacity, thereby making it easier to absorb a growing demand or loss of capacity due to failure. This results in a supply chain that is more robust and flexible. When the customer indicates to prefer capacity to be not fully utilized for all production or distribution locations, we recommend to apply the Robust HSJ method (version 2). With the modified method that maximizes minimum relative unused capacity, an alternative can be obtained that has for each production facility or distribution location the same percentage of slack. For example, if this percentage equals 1%, it means that an increase of total demand by 1% can still be fulfilled by the production or distribution capacity. It can also be relevant to have a certain percentage of slack, such that maintaince can be scheduled easily and failures do not disturb production too much.

6.3 Further research

Based on the assumptions made in this research and the knowledge gained on methods to generate meaningful alternative solutions for BOSS, we give some suggestions for further research.

1. Influence cost structure on solutions from Cost uncertain HSJ

With the Cost uncertain HSJ method, we find no relevant alternative solutions for one of the test cases, while for the other test case in each scenario we obtain at least one alternative that is preferred over the initial solution. More research is needed, by applying the Cost uncertain HSJ method to test cases with different cost structures, to be able to draw conclusions on what kind of cost structure makes it likely for this method to obtain relevant alternatives.

2. Distribution uncertain cost parameters

Throughout this research we assume that no information on the distribution of the uncertain cost parameters is available, and the uniform distribution best matches this situation. Therefore, it might be worth testing the Cost uncertain HSJ method on TC Bread with different distributions of cost parameters. For example, a normal distribution gives a higher probability to the expected value of a cost parameter. Also, in that case, a modeler does not have to specify a hard lower and upper limit. Nevertheless, a normal distribution requires indicating the standard deviation of the data, and this may also be difficult to specify. If a consultant of ORTEC wants to apply this method in a study for a customer and enough historical data is available, he might be able to base the assumed distribution on actual data and test the Cost uncertain HSJ method with this distribution. Such tests result in a better understanding of the assumptions on the influence of uncertain cost parameters on this method.

3. More alternatives from Robust HSJ method

For the Robust HSJ method (version 1) we found for our test cases only one or two alternative solutions, that allocate slack over a small number of locations. However, we like to find more alternatives, and distribute slack over more locations. A modification to this method that is worth testing is to introduce binary variables measuring whether a location or production facility has unused capacity. Then we can force the slack to be distributed over more locations by taking these decision variables into the objective function. However, we should also enforce a pre-defined minimum slack, since otherwise a very small amount of slack already qualifies.

4. New methods and better fitting methods to customer wishes

Finally, in general, the discussed methods should be applied to more test cases to increase confidence of the users. If consultants of ORTEC use one or more methods in case studies they performs for customers with BOSS, they can use feedback of those customers to improve the search for the type of alternatives a customer requests. Based on this information new realizations of the GHSJ framework can be developed and tested. For example, a customer with an environmental focus, may be interested in decreasing CO_2 emission. Section 4.4.5 discusses a number of such methods that might be business wise relevant.

Bibliography

[1] ORTEC company profile (n.d.). Retrieved September 28, 2011 from *http://www.ortec.com.*

[2] Balakrishnan, A. and Altinkemer, K. (1992), Using a hop-constrained model to generate alternative communication network design. *ORSA Journal on Computing*, Vol. 4, pp. 192-205.

[3] Bellman, R. and Zadeh, L. (1970), Decision-making in a fuzzy environment. *Management Science*, Vol. 17, pp. 141-161.

[4] Ben-Tal, A. and Nemirovski, A. (2000), Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, Vol. 88, No. 3, pp. 411-424.

[5] Bertsimas and Sim (2003), Robust discrete optimization and network flows. *Mathematical programming*, Vol. 98, pp. 49-71.

[6] Birge, J. and Louveaux, F. (1997), Introduction to stochastic programming. *Springer*, New York.

[7] Brill, E., Chang, J and Hopkins, L. (1982), Modeling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning. *Management Science*, Vol. 28, No. 3, pp. 221-235.

[8] Brooks, S. (1958), A discussion of random methods for seeking maxima. *Operations Research*, Vol. 6, No. 2, pp. 244-251.

[9] Camm, J., Raturi, A., and Tsubakitani, S. (1990), Cutting Big M down to size. *Interfaces*, Vol. 20 , No. 5, pp. 61-66.

[10] Chang, S. (1981) Methods for generating alternative solutions to water resource planning problems. *PhD dissertation, University of Illinois at Urbana-Champaign.*

[11] Chang, S, Brill, E., and Hopkins, L. (1982), Use of mathematical models to generate alternative solutions to water resource planning problems. *Water Resources Research*, Vol. 18, No. 1, pp. 58-64.

[12] Chang, S., Brill, E., and Hopkins, L. (1983), Modeling to generate alternatives: a fuzzy approach. *Fuzzy sets and systems*, Vol. 9, pp. 137-151.

[13] Charnes, A. and Cooper, W. (1959), Chance-constrained programming. *Management Science*, Vol. 6, No. 1, pp. 73-79.

[14] Danna, E., Fenelon, M., Gu, Z., and Wunderling, R. (2007), Generating multiple solutions for mixed integer programming problems. *Integer Programming and Combinatorial Optimization*, Vol. 4513 of LNCS, pp. 280-294.

[15] Dantzig, G. (1955), Linear programming under uncertainty. *Management Science*, Vol. 1, pp. 197-206.

[16] Huang, G., Huaicheng, G. and Guangming, Z. (1997), A mixed integer linear programming approach for municipal solid waste management. *Journal of Environmental Sciences*, Vol. 9, No. 4, pp. 431-445.

[17] Kataoka, S. (1963), A stochastic programming model. *Econometrica*, Vol. 31, No. 1/2, pp. 181-196.

[18] Kolesar, P., Rider, K., Crabill, T. and Walker, W. (1975), A queuing-linear programming approach to scheduling police patrol cars. *Operations Research*, Vol. 23, No. 6, pp. 1045-1062.

[19] Kouvelis, P. and Yu, G. (1997), Robust discrete optimization and its applications. *Kluwer Academic Publishers*, Norwell, MA.

[20] Kripakaran, P. and Gupta, A. (2006), MGA - a mathematical approach to generate design alternatives. *Intelligent Computing in Engineering and Architecture*, Lecture Notes in Computer Science, vol. 4200, pp. 408-415.

[21] Loughlin, D., Ranjithan, R., Brill, E. and Baught, J. (2001), Genetic algorithm approaches for addressing unmodeled objectives in optimization problems. *Engineering Optimization*, Vol. 33, pp. 549-569.

[22] Mulvey, J., Vanderbei, R., Zenios, S. (1995), Robust optimization of large-scale systems. *Operations Research*, Vol. 43, No. 2, pp. 264-281.

[23] Nakamura, M. and Brill, E. (1979), Generation and Evaluation of Alternative Plans for Regional Wastewater Systems: An Imputed Value Method. *Water resources research*, Vol. 15, No. 4, pp. 750-756.

[24] Sahinidis, N. (2004), Optimization under uncertainty: state-of-the-art and opportunities. *Computers and Chemical Engineering*, Vol. 28, No. 6-7, pp. 971-983.

[25] Tsiakis, P., Shah, N. and Pantelides, C. (2001), Design of multi-echelon supply chain networks under demand uncertainty. *Industrial & Engineering Chemical Research*, No. 40, pp. 3585-3604.

[26] Zechman, E. and Ranjithan, R. (2004), An evolutionary algorithm to generate alternatives (EAGA) for engineering optimization problems. *Engineering Optimization*, Vol. 36, No. 5, pp. 539-553.

[27] Zimmerman, H. (1978), Fuzzy programming and linear programming with several objective functions. *Fuzzy sets and Systems*, Vol. 1, pp. 45-55.

Appendix A

BOSS model description

This appendix summarizes the BOSS model described in Chapter 2. Section A.1 gives the sets and parameters of the model. Section A.2 describes the decision variables and Section A.3 indicates the objective function. Finally, Section A.4 discusses the restrictions that hold for the model.

A.1 Sets and parameters

The BOSS model uses the following main data sets:

1. $\mathcal{L}$: set of all physical locations in the network;
2. $\mathcal{F}$: set of all facilities;
3. $\mathcal{F}^l \subseteq \mathcal{F}$: set of all facilities located at physical location l, i.e., $\mathcal{F}^l : \emptyset$ if no production can take place at location l, for example if l is a customer location;
4. $\mathcal{M}$: set of commodities;
5. $\mathcal{P}$: set of products;
6. $\mathcal{G}$: set of ingredients for the production process, i.e., $\mathcal{G} : \mathcal{M} \cup \mathcal{P}$;
7. $\mathcal{I}$: set of potential investments. Each investment is related to either a location in its entirety or to a single facility;
8. $\mathcal{I}^{fl(l)} \subseteq \mathcal{I}$: set of potential investments which increase or decrease the flow capacity of location l;
9. $\mathcal{I}^{pr(f)} \subseteq \mathcal{I}$: set of potential investments which increase or decrease the capacity of production facility f.

Furthermore, we have the following data available:

1. $demand\,(p, l)$: demand for product p by location l;
2. $prodcap\,(f)$: maximum production capacity of production facility f;
3. $investprodcap\,(i, f)$: maximum production capacity of production facility f if investment i is done;
4. $flowcap\,(p, l)$: maximum throughput of product p at location l;

5. $investflowcap\,(i,p,l)$: maximum throughput of product p at location l if investment i is done;

6. $recipe_f\,(g)$: number of units of ingredient g used or produced at facility f;

7. $maxsupply\,(p,l)$: maximum supply of product p at location l;

8. $commavail\,(m,l)$: available number of units of commodity m at location l;

9. $bigMFlow\,(p,l) \;=\; \max\left(\max_{i\in\mathcal{I}^f l(l)} (investflowcap\,(i,p,l))\,,flowcap\,(p,l)\right).$

A.2 Decision variables

The decision variables in the BOSS model are:

1. $$LocOpen\,(l) = \begin{cases} 1 & \text{if location } l \text{ is in use;} \\ 0 & \text{otherwise.} \end{cases}$$

2. $$InvDone\,(i) = \begin{cases} 1 & \text{if investment } i \text{ is made;} \\ 0 & \text{otherwise.} \end{cases}$$

3. $Production\,(f)$: production level at production facility f;

4. $ProdIn\,(p,f)$: number of units of product p used in the production process of one production unit at facility f;

5. $ProdOut\,(p,f)$: number of units of product p produced from one production unit at facility f;

6. $Flow\,(p,l_1,l_2)$: number of units of product p transported from location l_1 to location l_2;

7. $Supply\,(p,l)$: number of units of product p supplied to location l;

8. $CommUse\,(m,f)$: number of units of commodity m used at facility f.

A.3 Objective function

The objective function is to minimize the total costs of the supply chain design, i.e.,

$$\begin{aligned} \min Cost = {} & TransportCost + LocationCost + ProductionCost \\ & + SupplyCost + InvestmentCost, \end{aligned} \tag{A.1}$$

where

$$TransportCost = \sum_{\substack{p\in\mathcal{P}\\ l_1,l_2\in\mathcal{L}|l_1\neq l_2}} Flow\,(p,l_1,l_2)\cdot vartransportcost\,(p,l_1,l_2); \tag{A.2}$$

$$LocationCost = \sum_{l\in\mathcal{L}} LocOpen\,(l)\cdot fixedlocationcost\,(l); \tag{A.3}$$

$$\begin{aligned} ProductionCost = &\sum_{l\in\mathcal{L}}\sum_{\substack{m\in\mathcal{M}\\ f\in\mathcal{F}}} CommUse\,(m,f)\cdot varcommoditycost\,(m,l)\\ &+\sum_{f\in\mathcal{F}^l} Production\,(f)\cdot varproductioncost\,(f); \end{aligned} \tag{A.4}$$

$$SupplyCost = \sum_{\substack{p\in\mathcal{P}\\ l\in\mathcal{L}}} Supply\,(p,l)\cdot varsupplycost\,(p,l); \tag{A.5}$$

$$InvestmentCost = \sum_{i\in\mathcal{I}} InvDone\,(i)\cdot fixedinvestmentcost\,(i). \tag{A.6}$$

The meaning of the cost parameters is as follows:

- $vartransportcost\,(p,l_1,l_2)$: variable cost of transporting one unit of product p from location l_1 to location l_2;
- $fixedlocationcost\,(l)$: fixed cost of opening location l;
- $varcommoditycost\,(m,l)$: variable cost of one unit of commodity m at location l;
- $varproductioncost\,(f)$: variable cost of one production unit at facility f;
- $varsupplycost\,(p,l)$: variable cost of one unit of product p supplied at location l, which includes transportation cost and purchase price;
- $fixedinvestmentcost\,(i)$: fixed cost of making investment i.

A.4 Constraints

The following restrictions should hold:

1. All variables should be nonnegative, i.e., all variables ≥ 0;
2. $LocOpen\,(l), InvDone\,(i) \in \{0,1\}$;
3. $$\sum_{l_1\in\mathcal{L}|l_1\neq l} Flow\,(p,l,l_1) \leq LocOpen\,(l)\cdot bigMFlow\,(p,l) \quad \forall\, l\in\mathcal{L}; \tag{A.7}$$
4. $$CommUse\,(m,f) = Production\,(f)\cdot recipe_f\,(m) \qquad \forall\, m\in\mathcal{M},\ f\in\mathcal{F}; \tag{A.8}$$
5. $$ProdIn\,(p,f) = -1\cdot Production\,(f)\cdot recipe_f\,(p) \qquad \forall\, p\in\mathcal{P},\ f\in\mathcal{F}|\,recipe_f\,(p)<0; \tag{A.9}$$

6.

$$ProdOut\,(p,f) = Production\,(f)\cdot recipe_f\,(p) \qquad \forall\, p\in\mathcal{P},\ f\in\mathcal{F}|\, recipe_f\,(p)\geq 0; \tag{A.10}$$

7.

$$\sum_{l_1\in\mathcal{L}|l_1\neq l} Flow\,(p,l_1,l) + \sum_{f\in\mathcal{F}^l} ProdOut\,(p,f) + Supply\,(p,l) = \\ \sum_{l_2\in\mathcal{L}|l_2\neq l} Flow\,(p,l,l_2) + \sum_{f\in\mathcal{F}^l} ProdIn\,(p,f) + demand(p,l) \qquad \forall\, p\in\mathcal{P},\ l\in\mathcal{L};$$

8.

$$Supply\,(p,l)\leq maxsupply\,(p,l) \qquad \forall\, p\in\mathcal{P},\ l\in\mathcal{L}; \tag{A.11}$$

9.

$$\sum_{f\in\mathcal{F}^l} CommUse\,(m,f)\leq commavail\,(m,l) \qquad \forall\, m\in\mathcal{M},\ l\in\mathcal{L}; \tag{A.12}$$

10.

$$\sum_{i\in\mathcal{I}^{fl(l)}} InvDone\,(i)\leq 1 \qquad \forall\, l\in\mathcal{L}; \tag{A.13}$$

11.

$$\sum_{i\in\mathcal{I}^{pr(f)}} InvDone\,(i)\leq 1 \qquad \forall\, f\in\mathcal{F}; \tag{A.14}$$

12.

$$AvailProdCap\,(f) = prodcap\,(f)\cdot\left(1-\sum_{i\in\mathcal{I}^{pr(f)}} InvDone\,(i)\right) \\ + \sum_{i\in\mathcal{I}^{pr(f)}} investprodcap\,(i,f)\cdot InvDone\,(i) \quad \forall\, f\in\mathcal{F}. \tag{A.15}$$

13.

$$AvailFlowCap\,(p,l) = flowcap\,(p,l)\cdot\left(1-\sum_{i\in\mathcal{I}^{fl(l)}} InvDone\,(i)\right) \\ + \sum_{i\in\mathcal{I}^{fl(l)}} investflowcap\,(i,p,l)\cdot InvDone\,(i). \quad \forall\, p\in\mathcal{P},\ l\in\mathcal{L}. \tag{A.16}$$

14.

$$Production\,(f)\leq AvailProdCap\,(f) \qquad \forall\, f\in\mathcal{F}. \tag{A.17}$$

15.

$$\sum_{l_1\in\mathcal{L}|l_1\neq l} Flow\,(p,l,l_1)\leq AvailFlowCap\,(p,l) \qquad \forall\, p\in\mathcal{P},\ l\in\mathcal{L}. \tag{A.18}$$

Appendix B

Visualization data test cases

Figure B.1 displays the current production locations of the Bread test cases in blue and also the potential new production locations for TC Bread (small) in red. For TC Bread (large) 20 more potential new production locations are defined; these are not displayed. Figure B.2 displays for TC Oil the three production locations in red, and the 21 distribution locations in green. The little dark blue dots on both maps are customer locations.

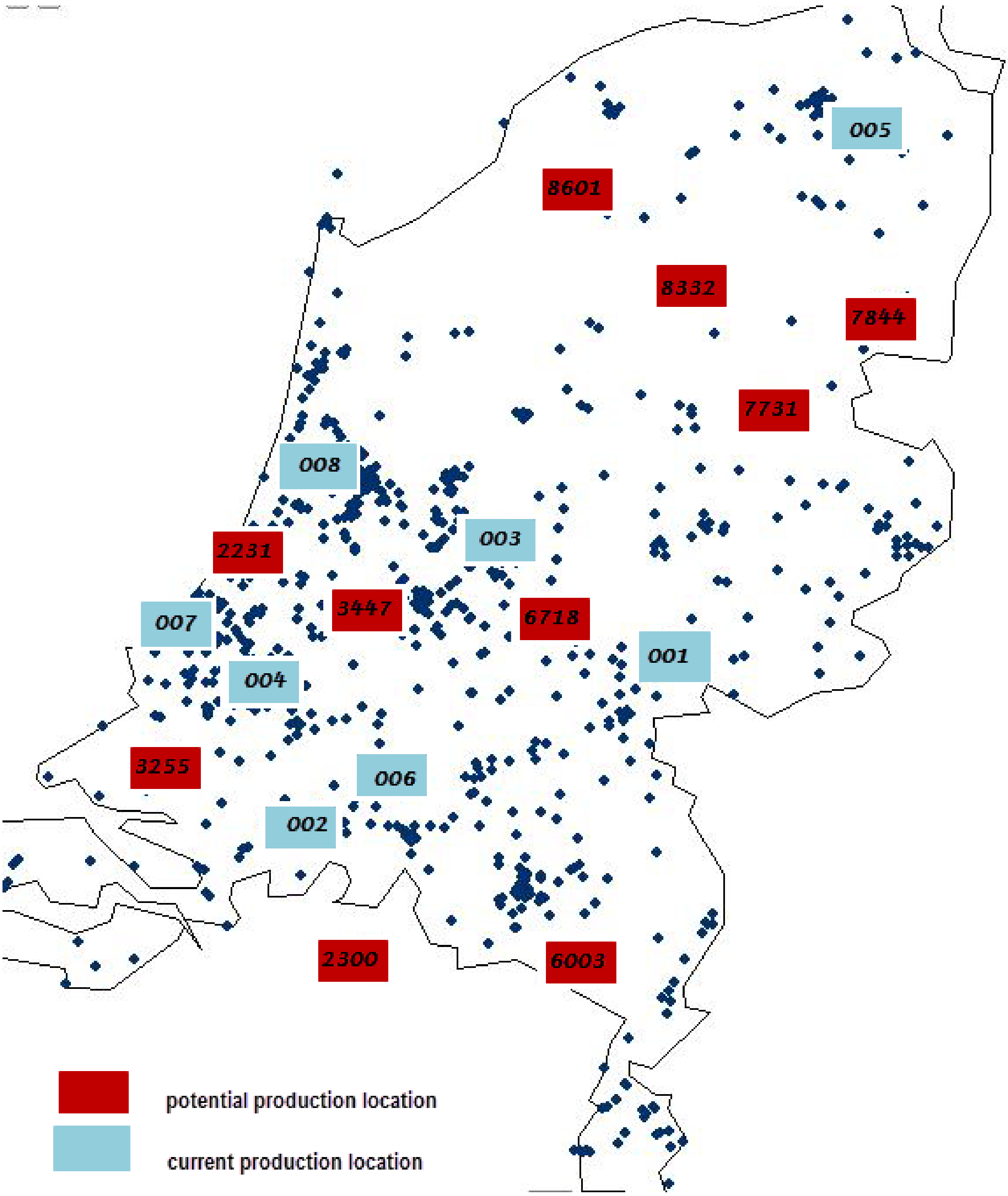


Figure B.1: Graphical overview of TC Bread (small).

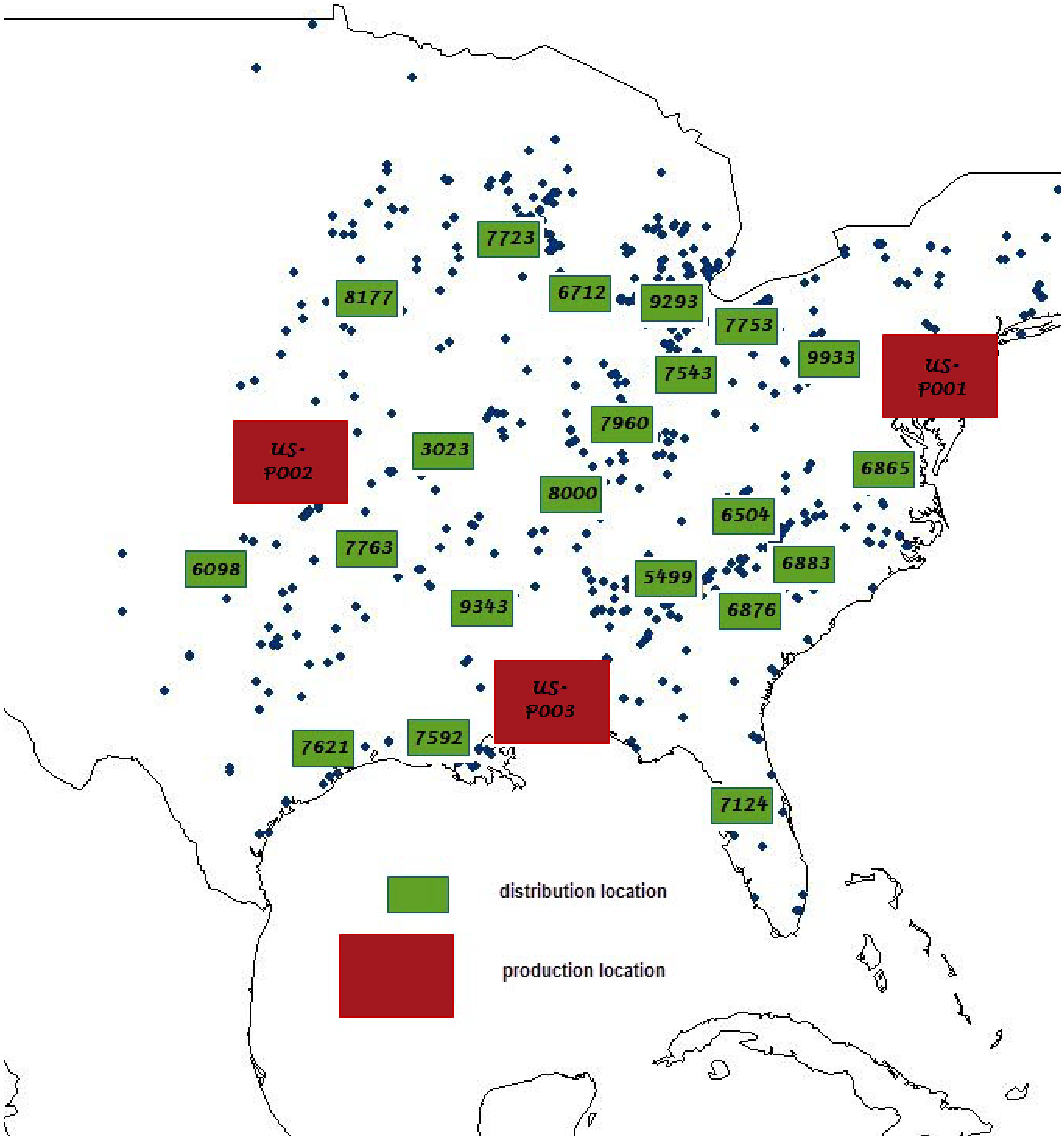


Figure B.2: Graphical overview of TC Oil.

Appendix C

Standard HSJ results

Table C.1 displays the obtained solutions for TC Bread (small) and an allowed cost increase of 5%. For example, alternative 3 has only 2.70% higher cost than optimal, while total transport distance is 2.47% lower. There could also be other reasons to prefer one alternative over another. For example, if a decision maker prefer to keep NL005 open, he might prefer alternative 2 over the initial solution. Table C.2 displays these same results for an allowed cost increase of 10%. From comparing these two tables, we notice that the obtained solutions for an allowed cost increase of 10% have no overlap with the solutions obtained when the allowed cost increase is 5%, even though these solutions are also within the allowed cost range. Results for TC Bread (large) are similar, but less easy overseen due to the higher number of locations and investments. Table C.3 displays the results for TC Oil for an allowed cost increase of 5%.

	Initial solution	Alternative 1	Alternative 2	Alternative 3
BOSS objective value	€961,221.43	€1,000,505.45	€1,002,193.70	€987,203.97
Cost increase %		4.09%	4.26%	2.70%
Difference objective value		7	10	11
Binary difference		10	4	4
Capacity difference		60575	68315	25324
Total transport distance	264,171,048	263,180,907	260,996,446	257,657,410
Fixed cost / total cost	0.2341	0.2892	0.2770	0.2792
Locations				
NL001	X	X	X	X
NL002	X	X	X	X
NL003		X		X
NL004	X	X	X	X
NL005			X	
NL006	X	X	X	X
NL007	X	X	X	X
NL008	X	X	X	X
BE2300	X		X	
NL2231				
NL3255				
NL3447	X		X	
NL6003				
NL6718				
NL7731	X		X	X
NL7844		X		
NL8332				
NL8601	X			X
Investments				
NL001 closure				
NL001 invest		X		X
NL002 closure				
NL002 invest		X		
NL003 closure	X		X	
NL004 closure				
NL005 closure	X	X		X
NL006 closure				
NL006 invest	X			
NL007 closure				
NL007 closure				

Table C.1: Results Standard HSJ for TC Bread (small) and $k = 0.05$.

	Initial solution	Alternative 1	Alternative 2	Alternative 3
BOSS objective value	€961,221.43	€1,047,074.65	€1,036,186.61	€1,033,678.27
Cost increase %		8.93%	7.80%	7.54%
Difference objective value		6	8	9
Binary difference		10	6	4
Capacity difference		67459	40701	19480
Total transport distance	264,171,048	270,838,554	254,659,369	257,665,671
Fixed cost / total cost	0.2341	0.3305	0.3475	0.3268
Locations				
NL001	X	X	X	X
NL002	X	X	X	X
NL003		X	X	X
NL004	X	X	X	X
NL005		X		X
NL006	X	X	X	X
NL007	X	X	X	X
NL008	X	X	X	X
BE2300	X			X
NL2231				
NL3255				
NL3447	X			
NL6003			X	
NL6718				X
NL7731	X			
NL7844		X		
NL8332			X	
NL8601	X			
Investments				
NL001 closure				
NL001 invest			X	
NL002 closure				
NL002 invest				X
NL003 closure	X			
NL004 closure				
NL005 closure	X		X	
NL006 closure				
NL006 invest	X			
NL007 closure				
NL008 closure				

Table C.2: Results Standard HSJ for TC Bread (small) and $k = 0.10$.

	Initial solution	Alternative 1	Alternative 2	Alternative 3
BOSS objective value	€17,825,599.47	€18,694,181.96	€18,202,618.56	€18,080,330.14
Cost increase %		4.87%	2.12%	1.43%
Difference objective value		9	12	13
Binary difference		9	6	4
Capacity difference		1071172.17	748246.50	338619.53
Total transport distance	1,276,904,993	1,351,437,472	1,339,595,843	1,314,290,037
Fixed cost / total cost	0.5782	0.5937	0.5829	0.5856
Locations				
US-D5499	X			X
US-D6098		X		
US-D6712			X	
US-D6865	X		X	X
US-D6876		X		
US-D6883				X
US-D7960	X	X	X	X
US-D8000	X	X	X	X
US-D3023	X		X	
US-D7543	X	X	X	X
US-D7553	X	X	X	X
US-D7723	X			
US-D7763	X	X	X	X
US-D6504		X		
US-D7592	X	X	X	X
US-D7621	X	X	X	X
US-D9293		X	X	X
US-D9343	X	X	X	X
US-D9933	X	X	X	X
US-D2011	X			X
US-D8177				

Table C.3: Results Standard HSJ method for TC Oil and $k = 0.05$.

Appendix D

Discussion on cost part of objective function

In this appendix we explain how we can make the Standard HSJ and Random HSJ model hierarchical, with difference as primary objective. A hierarchical objective is realized in linear programming by applying weight factors to the different objectives in the objective function. Therefore, we discuss how γ should be chosen to achieve a hierarchical objective function.

The objective for the Standard HSJ and Random HSJ can be split in two parts. For example for the Standard HSJ this gives

$$\min \quad \underbrace{\sum_{l \in J_L} \mathit{LocOpen}\,(l) + \sum_{i \in J_I} \mathit{InvDone}\,(i)}_{\text{difference part}} \underbrace{- \gamma \cdot S}_{\text{cost part}}.$$

For the cost part we define

$$\gamma = \gamma' \cdot \frac{1}{k \cdot z_0}, \tag{D.1}$$

where z_0 is the objective value of the optimal solution. Thus the cost part can be written as

$$\gamma' \cdot \frac{S}{k \cdot z_0}. \tag{D.2}$$

The decision variable S achieves its maximum when the cost of the alternative solution is equal to the cost of the optimal solution; in that case $S = k \cdot z_0$. Thus, the cost part is at most equal to γ'. We can choose γ' such that the term $\gamma \cdot S$ is always smaller than the smallest impact of the binary decision variables, in that case the objective function is hierarchical; i.e., the difference part is the primary objective, and only if two solutions have equal score on the difference part, the solution with lowest cost is preferred.

For the Standard HSJ method we should choose $0 < \gamma' < 1$, since the variables $\mathit{LocOpen}\,(l)$ and $\mathit{InvDone}\,(i)$ are binary decision variables. Thus, if $\gamma' < 1$, the cost part is always smaller than 1, and no trade-off can exist between difference and cost.

Due to the random weight factors in th objective function of the Random HSJ method, it is less clear how to choose γ' for this method. The random numbers are drawn from the interval [0,1], which means that if we choose $\gamma' = 0.01$, the probability of the occurrence of a trade-off situation is sufficiently small. To see this, we assume that there is only one decision variable *LocOpen* in the objective function, i.e., the objective function is

$$\min \alpha \cdot \mathit{LocOpen} - 0.01 \cdot \frac{S}{k \cdot z_0}. \tag{D.3}$$

Only if $\alpha \leq 0.01$, which happens with probability 0.01, a trade-off situation could exist between *LocOpen* and cost, since $\frac{S}{k \cdot z_0} \leq 1$. However, whether this trade-off situation actually occurs depends on the data structure and whether it is possible to decrease cost by setting $LocOpen\,(l)$ to 0, such that all restrictions are still fulfilled. When there are more decision variables in the objective function a similar situation holds.

Appendix E

Adapted Standard HSJ: minimum binary difference

The figures in this appendix display the minimum binary difference for each solution obtained with the modified Standard HSJ method, for each test case and $k = 0.10$. The lines below the horizontal axis show if the solution is found with the standard HSJ method or in an iteration of the modified method.

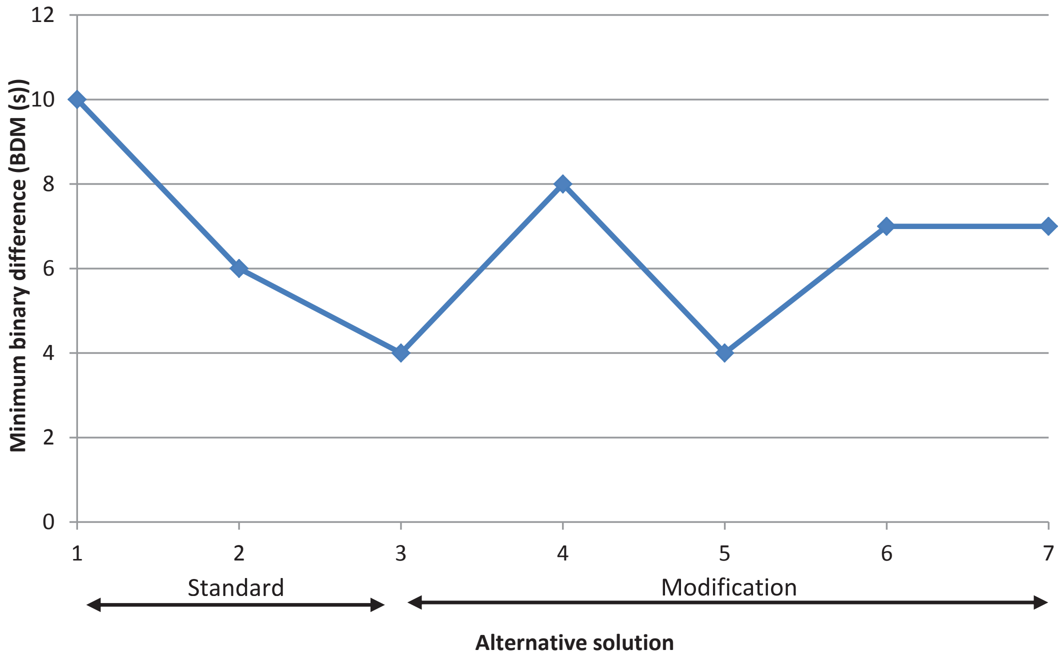


Figure E.1: Binary difference metric (BDM(s)) for solutions obtained with the modified Standard HSJ for TC Bread (small) and $k = 0.10$.

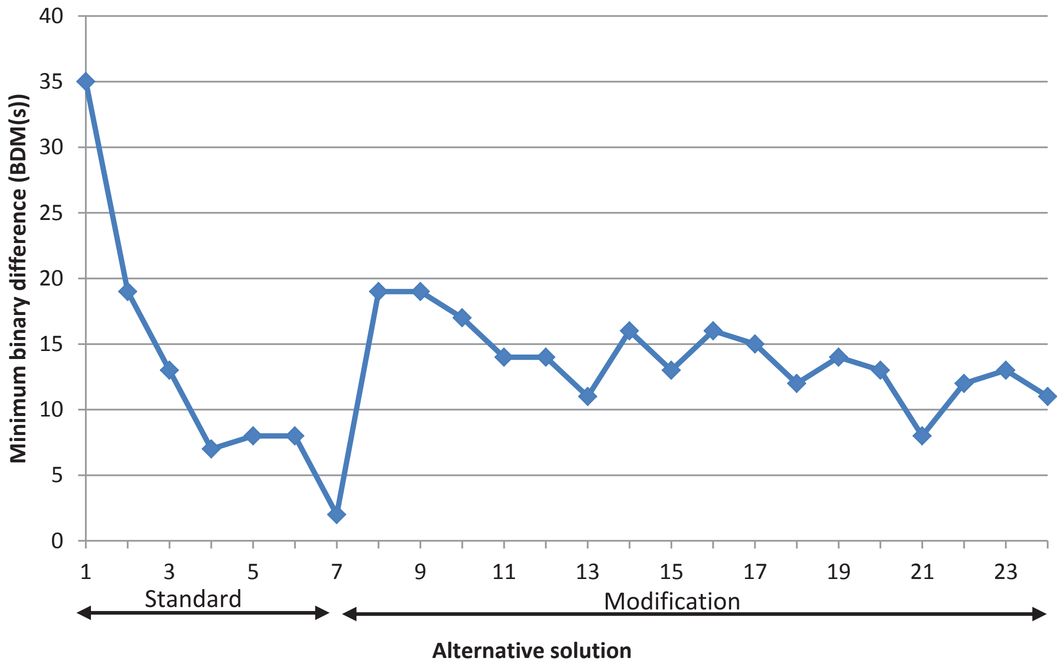


Figure E.2: Binary difference metric (BDM(s)) for solutions obtained with the modified Standard HSJ for TC Bread (large) and $k = 0.10$.

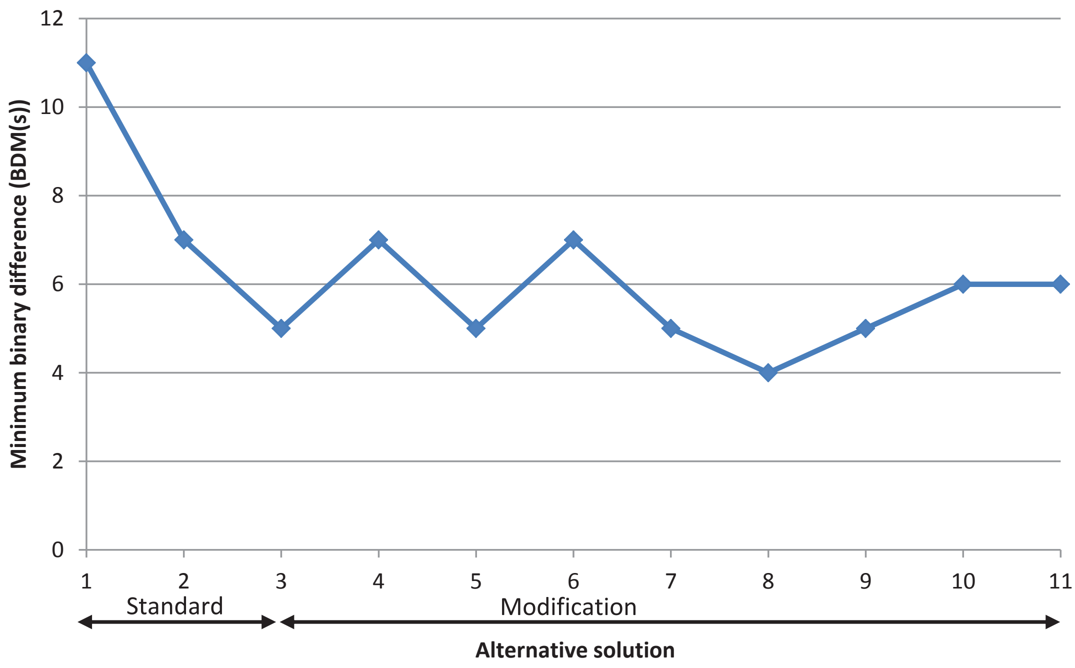


Figure E.3: Binary difference metric (BDM(s)) for solutions obtained with modified Standard HSJ for TC Oil and $k = 0.10$.

Appendix F

Random HSJ: minimum binary difference

The Figures F.1 and F.2 display for respectively TC Bread (small) and $k = 0.10$, and TC Bread (large) and $k = 0.02$ the minimum binary difference of each solution obtained with the Random HSJ method.

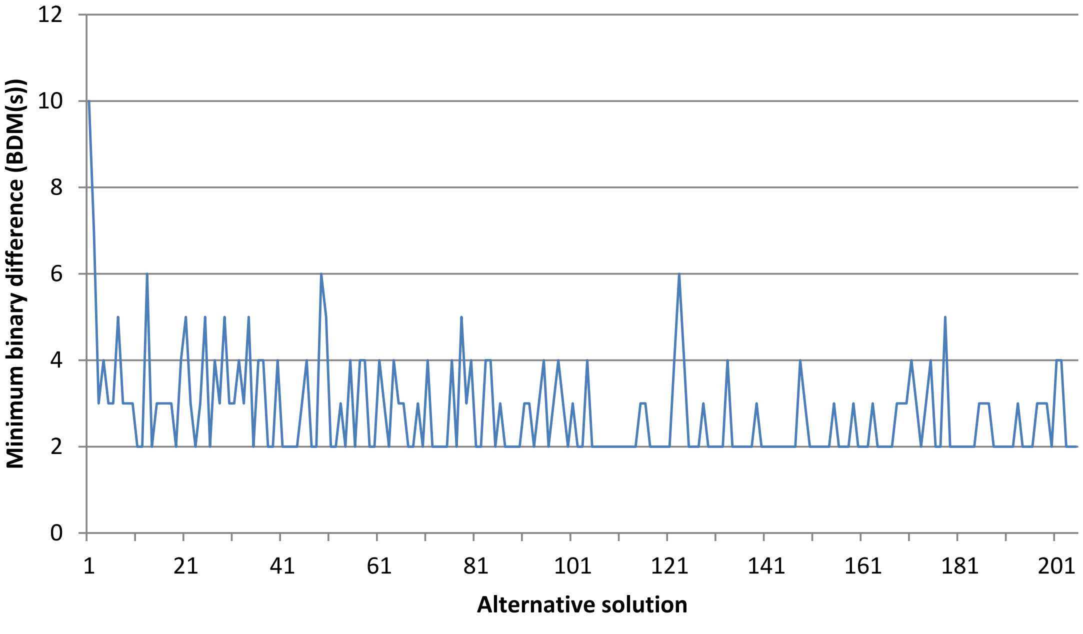


Figure F.1: Binary difference metric (BDM(s)) for solutions obtained with Random HSJ for TC Bread (small) and $k = 0.10$.

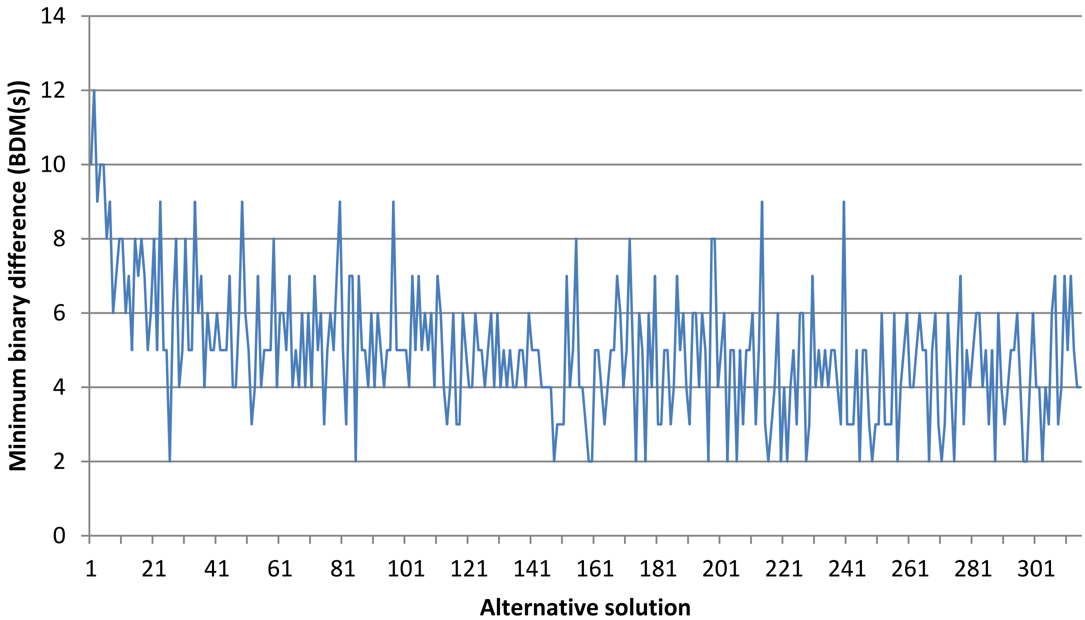


Figure F.2: Binary difference metric (BDM(s)) for solutions obtained with Random HSJ for TC Bread (large) and $k = 0.02$.

Appendix G

Random HSJ method: results different random seeds

Table G.1 displays the results of the Random HSJ method for different random seeds for TC Bread (small) and an $k = 0.02$. A letter classifies an exact combination of values for decision variables $LocOpen\,(l)$ and $InvDone\,(i)$. In bold are solutions not obtained in previous runs and in between brackets is the iteration in which the solution is obtained.

Number alternative	run 1	run 2	run 3	run 4	run 5
initial solution	**A (0)**	A (0)	A (0)	A (0)	A (0)
alt. 1	**B (1)**	F (1)	K (1)	M (1)	C (1)
alt. 2	**C (4)**	I (3)	G (2)	I (2)	I (2)
alt. 3	**D (5)**	G (4)	M (4)	D (3)	F (3)
alt. 4	**E (6)**	K (5)	E (6)	F (4)	E (7)
alt. 5	**F (7)**	E (6)	F (7)	K (6)	K (8)
alt. 6	**G (10)**	H (8)	B (13)	C (10)	M (10)
alt. 7	**H (16)**	M (9)	L (17)	B (11)	G (20)
alt. 8	**I (22)**	N (13)	C (22)	E (12)	L (22)
alt. 9	**J (23)**	D (20)	I (26)	L (15)	B (31)
alt. 10	**K (35)**	B (25)	D (34)	N (19)	D (45)
alt. 11	**L (46)**	C (52)	H (57)	G (30)	N (51)
alt. 12	**M (62)**	L (71)	J (63)	J (47)	O (55)
alt. 13	**N (82)**	**O (79)**	O (89)	O (60)	
alt. 14				H (66)	

Table G.1: Results of Random HSJ for TC Bread (small) and $k = 0.02$, for different random seeds.

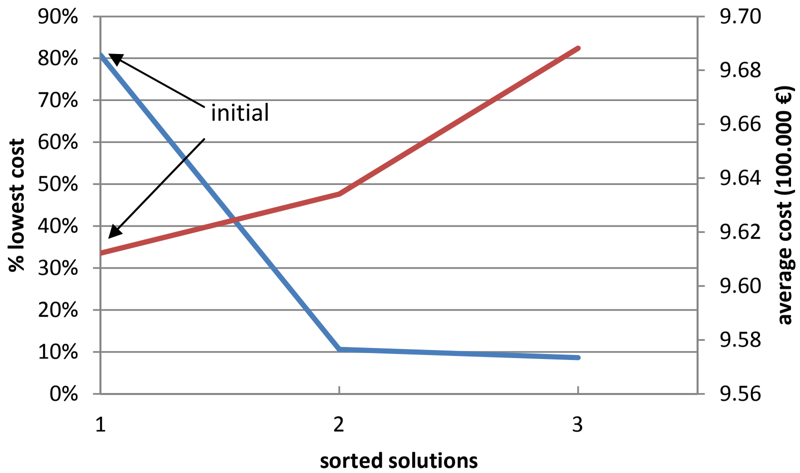
Appendix H

Cost uncertain HSJ results

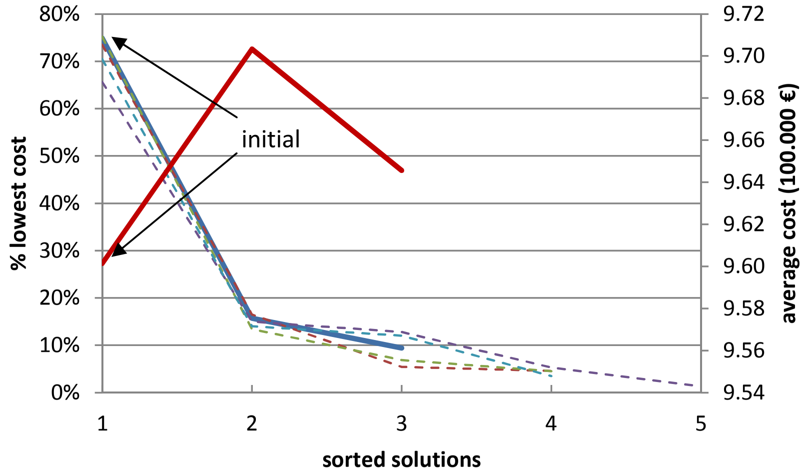
For the figures in this appendix, the solutions have been sorted based on the percentage of realizations in which the solution has lowest cost, from high to low. Then the sorted solutions are plotted against these percentages with a blue line. The shape of the graph tells us something about the distribution of the solutions and the relative performance. When the line starts very horizontal before decreasing faster, many solutions perform similar, whereas when the line directly decreases fast before stabilizing, a small group of solutions perform significantly better than most solutions. The red line gives the corresponding average cost of the solutions. In all figures also the initial solution is indicated with an arrow, such that we can analyze the relative performance of the initial solution. The dotted lines correspond to more experiments.

From the results in Figure H.1 we see that for TC Bread (small) the initial solution is for all experiments by far the best, i.e., the initial solution is the solution with the highest percentage of realizations for which this solution has the lowest cost compared to the other solutions. Figure H.2 displays the results for TC Bread (large). In this figure we see that the initial solution is not always the best solution, but it at least has lowest cost in a percentage of realizations close to the solution with the best score. Also, the two Bread test cases have lowest average cost over all realizations for the initial solution.

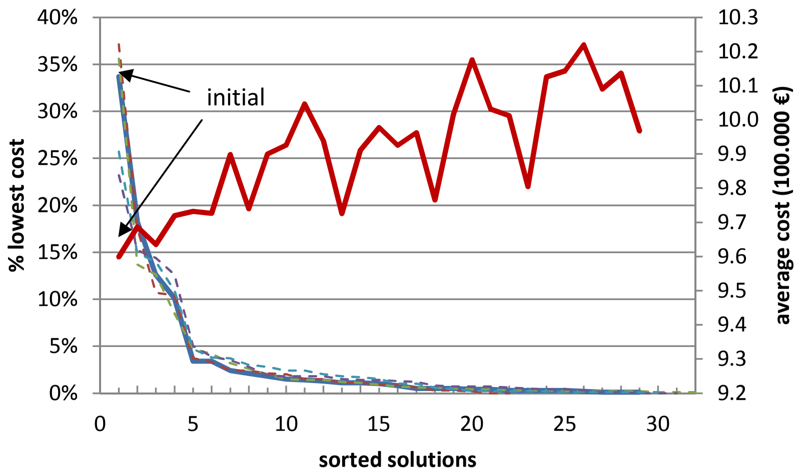
For TC Oil the results are displayed in Figure H.3. For this test case, the initial solution is in non of the experiments the best. It even holds that this solution is not even close to the best solution, i.e., the initial solution has lowest cost in much less scenarios than one or more obtained alternatives have. Also, average cost for the initial solution are higher than for the relevant alternative solutions.



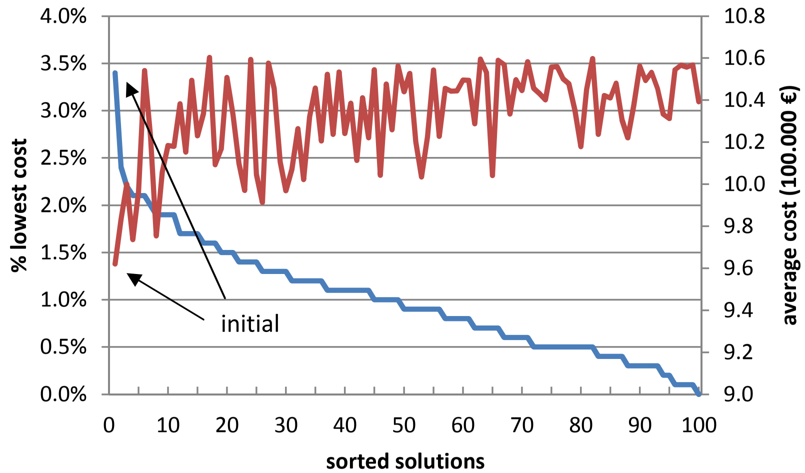
(a) Variance fixed cost: 0.05, variance variable cost: 0.05.



(b) Variance fixed cost: 0.10, variance variable cost: 0.10.

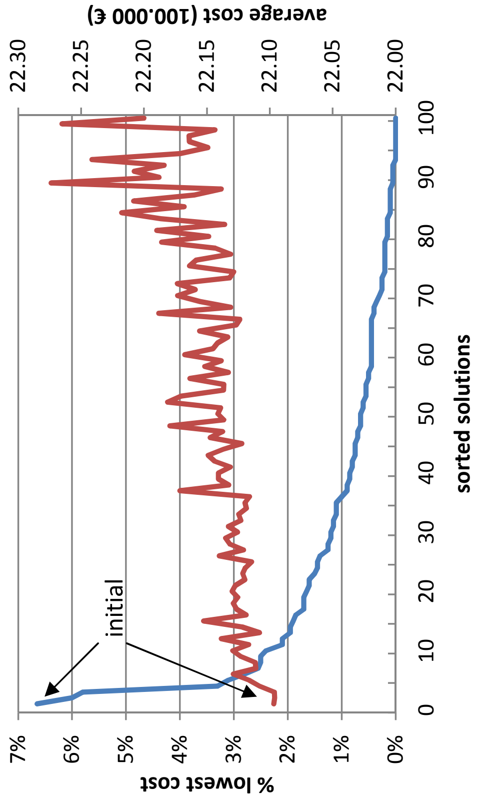


(c) Variance fixed cost: 0.25, variance variable cost: 0.10.

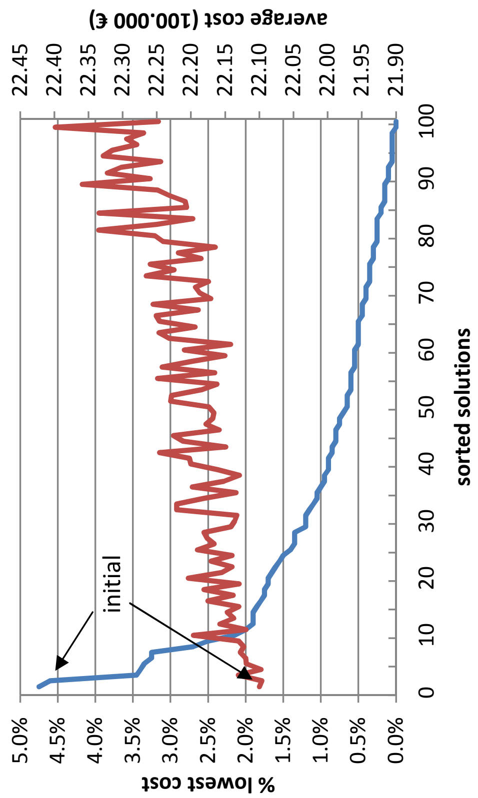


(d) Variance fixed cost: 0.80, variance variable cost: 0.50.

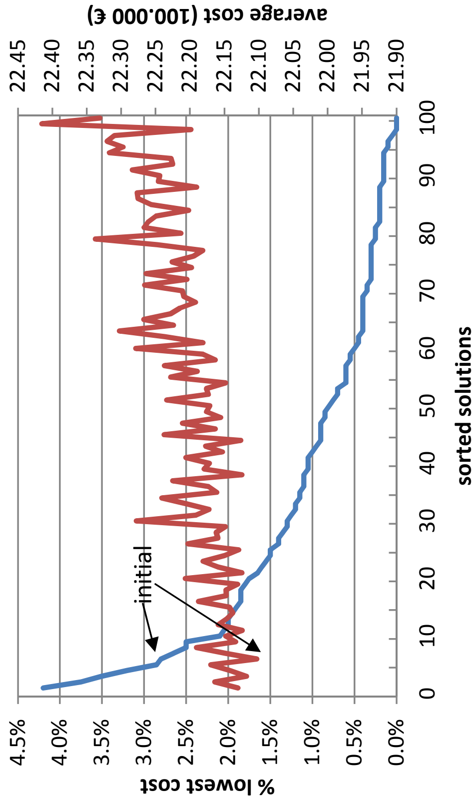
Figure H.1: Overview of results Cost uncertain HSJ method for TC Bread (small).



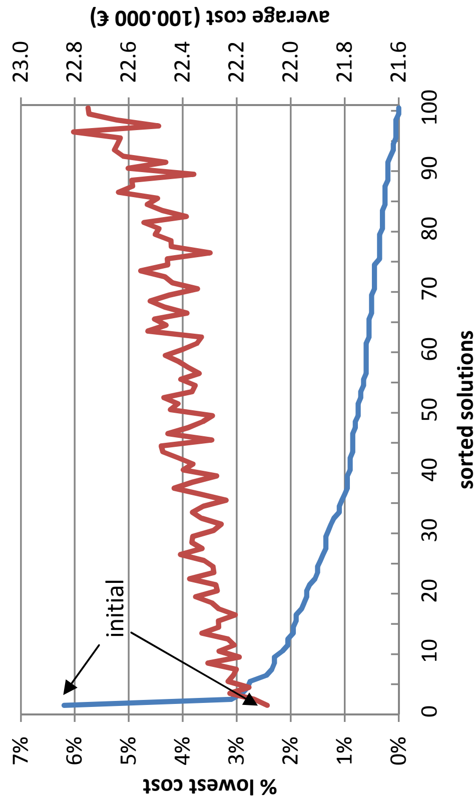
(a) Variance fixed cost: 0.05, variance variable cost: 0.05.



(b) Variance fixed cost: 0.10, variance variable cost: 0.05.

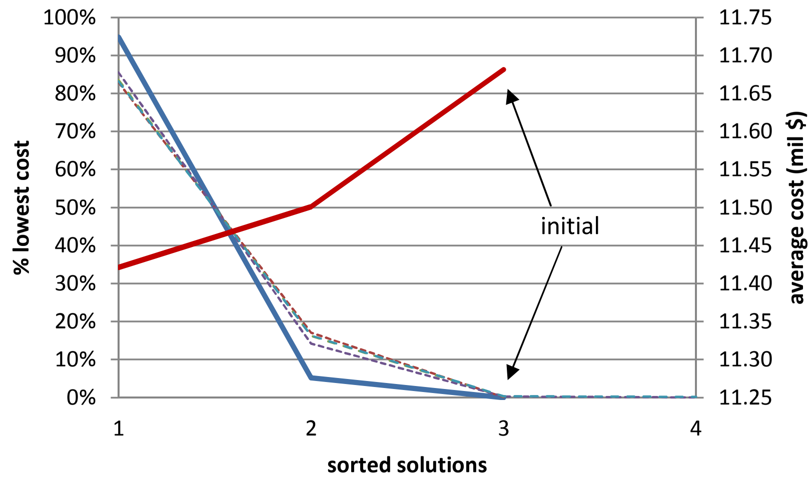


(c) Variance fixed cost: 0.10, variance variable cost: 0.10.

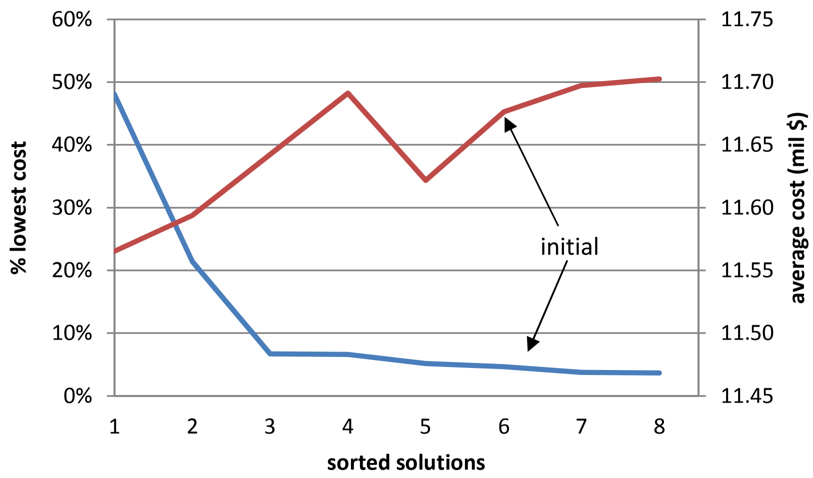


(d) Variance fixed cost: 0.25, variance variable cost: 0.10.

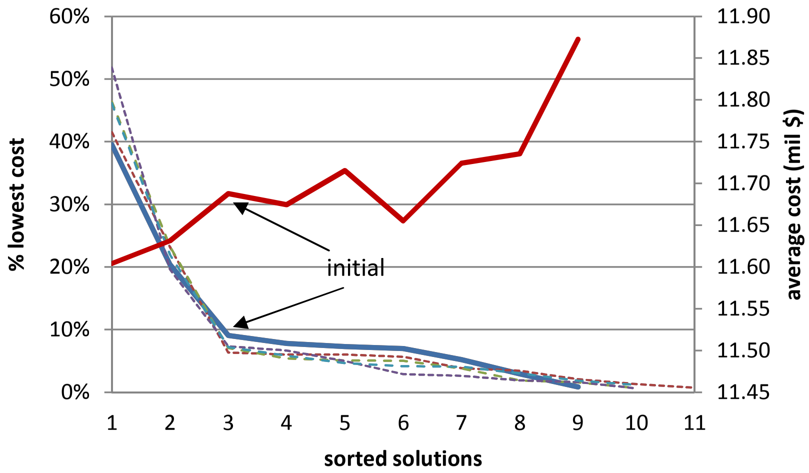
Figure H.2: Overview of results Cost uncertain HSJ method for TC Bread (large).



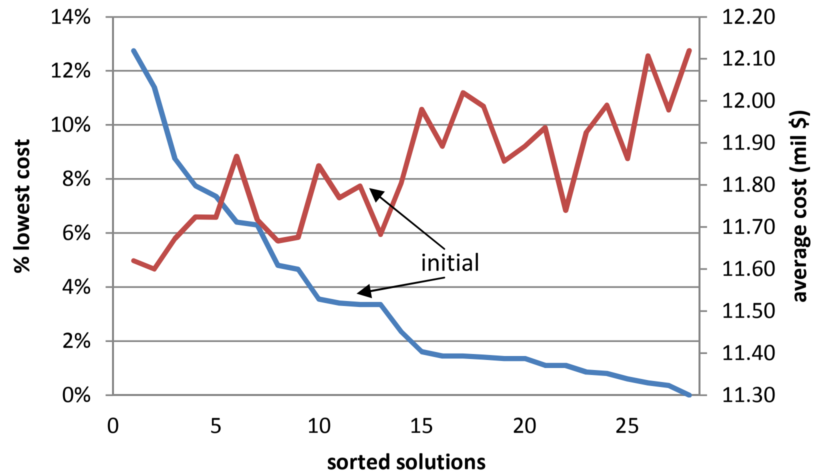
(a) Variance fixed cost: 0.05, variance variable cost: 0.05.



(b) Variance fixed cost: 0.10, variance variable cost: 0.05.



(c) Variance fixed cost: 0.10, variance variable cost: 0.10.



(d) Variance fixed cost: 0.25, variance variable cost: 0.10.

Figure H.3: Overview of results Cost uncertain HSJ method for TC Oil.

Appendix I

Cost uncertain HSJ: results different random seeds

The input of the Cost uncertain HSJ method is besides the assumed variances, also a large amount of pseudo-random numbers, based on a random seed. We want to test the influence of the chosen random seed on the obtained results. By repeating the experiments of Table 5.11 with different random seeds, we can determine whether these results are representative or that the chosen random seed highly influences the outcomes and completely different results are obtained with other random seeds. We test this by repeating for TC Bread (small) and TC Oil for two scenarios the performed tests with other random seeds. Altering the random seed, influences the process in two ways; the optimized objective function differs for each iteration and also the evaluated realizations differ.

Table I.1 shows the results of these experiments for TC Bread (small), including the results of the initial run and Table I.2 displays these results for TC Oil. From Table I.1 we see that results for TC Bread (small) are fairly stable, i.e., for both scenarios similar results are obtained with all five random seeds. The same holds for the results of TC Oil, where with all five random seeds the same solution is found to have the highest percentage of realizations in which it has lowest cost. From Table I.2 we see that although there is some variation in the percentages of realizations in which solution B has the lowest cost, in every run the same solution is obtained. Furthermore, this solution is always obtained within 10 iterations. Also, the number of obtained alternatives and the diversity measure are roughly equal for each of the five different random seeds.

scenario	variance cost fixed	variance cost variable	# obt. sol.	Diversity $D(S)$	best sol.	% this sol. lowest cost
2	0.10	0.10	3	0.03831	initial	74.90%
2	0.10	0.10	4	0.04741	initial	73.50%
2	0.10	0.10	4	0.04740	initial	75.20%
2	0.10	0.10	5	0.04414	initial	65.60%
2	0.10	0.10	4	0.03920	initial	70.50%
3	0.25	0.10	29	0.09964	initial	33.70%
3	0.25	0.10	22	0.09298	initial	37.10%
3	0.25	0.10	32	0.10281	initial	35.60%
3	0.25	0.10	28	0.10365	initial	37.60%
3	0.25	0.10	27	0.10420	initial	36.40%

Table I.1: Results of applying the Cost uncertain HSJ method on TC Bread (small) for two scenarios, with $MaxSol = 100$, $MaxEq = 100$ with different random seeds.

scenario	variance cost fixed	variance cost variable	# obt. sol.	$D(S)$	best sol.	% this sol lowest cost	% initial sol lowest cost
1	0.05	0.05	3	0.04630	B	94.85%	0.00%
1	0.05	0.05	3	0.05556	B	82.85%	0.10%
1	0.05	0.05	3	0.05556	B	83.55%	0.25%
1	0.05	0.05	4	0.05990	B	85.55%	0.25%
1	0.05	0.05	4	0.05469	B	83.15%	0.20%
3	0.10	0.10	9	0.07716	B	39.55%	9.05%
3	0.10	0.10	11	0.07989	B	41.50%	7.55%
3	0.10	0.10	10	0.07917	B	46.35%	7.05%
3	0.10	0.10	10	0.07792	B	51.85%	7.35%
3	0.10	0.10	10	0.07792	B	46.15%	7.20%

Table I.2: Results of applying the Cost uncertain HSJ method for different scenarios, with $MaxSol = 100$, $MaxEq = 100$, for TC Oil with different random seeds.

Appendix J

Robust HSJ (version 1) results

Table J.1a, J.1b, and J.2 display the results of the Robust HSJ method (version 1) for respectively TC Bread (small), TC Bread (large), and TC Oil. Note that, Table J.2 displays for TC Oil only distribution slack. For the production slack problem we have not found any alternative solutions, i.e., all solutions distribute production slack perfectly. Therefore, only results for distribution slack are displayed in Table J.2.

	initial solution	all. cost incr. 1%	all. cost incr. 2%		all. cost incr. 5%	all. cost incr. 7%	allowed cost increase 10%	
		alt. 1	alt. 1	alt. 2	alt. 1	alt. 1	alt. 1	alt. 2
Act. cost incr.	0.00%	0.18%	1.21%	1.85%	4.65%	6.55%	10.00%	9.97%
locations:								
NL001		1083					3116	
NL002	41		744	1488	4232	7403	5477	
BE2300								4513
NL3447								4451
BE7844					470			
NL8601							773	

(a) TC Bread (small)

	initial solution	allowed cost incr. 1%	allowed cost increase 2%		allowed cost increase 5%		allowed cost increase 7%		allowed cost increase 10%		
		alt. 1	alt. 1	alt. 2	alt. 1	alt. 2	alt. 1	alt. 2	alt. 1	alt. 2	alt. 3
Act. cost incr.	0.00%	0.98%	1.98%	1.96%	5.00%	5.00%	7.00%	6.99%	10.00%	10.00%	9.99%
locations:											
NL001						268		30		5569	
NL002			423		1574		5568		5880		1759
BE2300										1382	
BE2960											9398
BE3600						308		308		771	
NL1689	20			5489	1829	57	3285		4793		6097
NL3316								5874	3986		
NL3447						2333		2585		4407	
NL3845		2785	4879		7136		7245		7425		6817
NL4841						8282		7190		8545	
NL8211			262		2869		2418		2869		2896
NL8601						1204		1204	988		

(b) TC Bread (large)

Table J.1: Results of Robust HSJ (version 1) for TC Bread.

	initial	allowed cost increase 1%			all. cost incr. 2%		allowed cost increase 5%			allowed cost increase 7%			allowed cost. incr. 10%
	solution	alt. 1	alt. 2	alt. 3	alt. 1	alt. 2	alt. 1	alt. 2	alt. 3	alt. 1	alt. 2	alt. 3	alt. 1
Act. cost incr.	0.00%	0.89%	0.92%	0.86%	1.81%	1.63%	5.00%	3.31%	4.02%	5.72%	2.05%	5.17%	8.41%
locations:													
US-D2011	15100						6042						
US-D5499										7739		7739	
US-D6098								5209					
US-D6865										7739		7739	
US-D7543				985						7739		7739	
US-D7553													9660
US-D7723		8093								1249		3604	
US-D7763				12155		3324	8780				4727		
US-D7960			7083		5122				10169	19869		7739	
US-D8000								1302					

Table J.2: Results of Robust HSJ (version 1) for TC Oil.