SUPPLY CHAIN PLANNING WITH THE MACRO PLANNER

Erasmus School of Economics
Erasmus University Rotterdam

Master Thesis in Operations Research and Quantitative Logistics

Author:
Chiel van Oosterom BSc.

Supervisors:
Dr. Wilco van den Heuvel (EUR)
Drs. Peter Teijgeman (Capgemini)

February 17, 2010

Preface

This thesis is written to finish the Master Operations Research & Quantitative Logistics at the Erasmus School of Economics of the Erasmus University Rotterdam. As supervisor from the university I asked Wilco van den Heuvel due to his affinity with deterministic operations research and production problems in particular. His frequent constructive feedback was always a firm support to conduct this research. Also I would like to thank PhD student Twan Dollevoet for his help with the ILOG Concert Technology.

The subject of this thesis is a practical problem which came across at my internship at Capgemini. I would in general like to thank all colleagues at the Advanced Scheduling and Planning solution center for the work climate at Capgemini. For the possibility of writing this thesis at Capgemini I would like to thank Willem de Paepe who also triggered my interest in Capgemini with his presentation on the 'Landelijke Econometristendag'. Moreover I would like to thank Peter Teijgeman, my supervisor at Capgemini, for his time and support when I faced problems and his practical view on advanced planning and scheduling. Then I would like to thank the members of the algorithm expert group for their interest and coming up with the idea of robust planning which appeared to be a very interesting field. Also the implementation tips of Quintiq specialists Edwin de Caluwe and Luite van Zelst and the support with processing the shoe box dataset of Floris Beltman proved very valuable.

From the university I would like to thank the teachers but also my fellow students Harm Bossers and Luuk Veelenturf for our teamwork at assignments and their support in writing this thesis. At last I want to thank my parents for their support.

About the form of this thesis it should be mentioned that 'we' is used frequently and this is not meant as majestic plural but to indicate that the thesis is meant as dialog between author and reader. Everywhere when she is used in this thesis it could equally well be replaced by he.

Abstract

The Macro Planner is a Quintiq application which is among others implemented as supply chain planning tool by Capgemini. Given a certain sales forecast an optimal demand planning is made. When making a planning for a long horizon there is a lot of uncertainty for the actual demand. Currently only one sales forecast is used but in this thesis multiple sales forecast scenarios can be specified and a planning is created which is 'good' for all sales forecasts.

We create three models based on different definitions of good, such that for each type of risk-seeking there is a useful model. All our models belong the class of two-stage recourse models in robust optimization and instead of solving the deterministic model by CPLEX we could also apply decomposition methods. Therefore the two most important parts of the thesis are the implementation of our models in the Macro Planner on the one hand and the comparison of solution times for the solution methods for this kind of problems on the other hand.

Test for a real dataset turned out that extra expected profit could be made when a planning would be made in the expected goal function model compared to the normal supply chain planning. Also it was seen that it could be possible to raise the worst case profit with 10% when the worst case robustness measure was applied compared to normal supply chain planning.

In JAVA we implemented other solving techniques of our models based on decomposition methods with ILOG Concert Technology. For all our models we proposed some new options in the decomposition methods using the fact that there is only right-hand uncertainty in the models we considered. Although we could only generate a limited amount of results, there is an indication of the viability of our new methods, especially the ideas in the worst case model.

Thereafter we will briefly review other possible extensions to increase the attractiveness of the Macro Planner as in principle this was the goal. For capacity extension and CO_2 emission first suggestions are made, but nothing is implemented.

Contents

Chapter 1

Introduction

1.1 Company Information

1.1.1 Capgemini

Capgemini is a multinational company in IT consulting services with 91000 employees (in 2008) headquartered in Paris. It offers services in Consulting, Technology and Outsourcing businesses. In the Netherlands there are about 7000 employees. The Dutch headquarter is in Utrecht. Capgemini also has a subsidiary company, Sogeti, which offers local services.

1.1.2 Quintiq

Quintiq is a company founded in 1997 when five people splitted off Bolesian, a filial of Capgemini, to develop their own software to support advanced planning and scheduling decisions. In 2008 this company from Den Bosch had 180 employees and focuses on advanced planning and scheduling solutions for logistics, production and workforce.

1.1.3 APS Solution Center

The APS Solution Center is a group in the sector of products of Capgemini concentrating on Advanced Planning and Scheduling problems. This is set up by former employees of Bolesian, by then taken over by Capgemini, to work together with their former colleagues and implement their Quintiq software. Nowadays Capgemini evaluates planning and scheduling of customers and implements advanced solutions if necessary. The Quintiq software is now used less often to develop solutions.

1.2 General Problem Description

1.2.1 Problem definition

The APS unit of Capgemini implements among others the Macro Planner, a software application of Quintiq to support supply chain planning decisions. This application will be the subject of this thesis. The motivation of this thesis is that there are probably still a lot of possibilities to make the Macro Planner more attractive for customers by adjusting or expanding it.

One possible interesting option is to make it able to support robust planning. Currently in the Macro Planner (for this thesis version 4.5.0 is studied) only one specific sales forecast is used and an optimal supply chain planning is a planning in which the goal function is optimized to that sales forecast. However, in reality one

forecast is just an expectation and will normally turn out to be wrong. We would like to create a supply chain planning which also turns out to be 'good' for other realizations of the sales demand and we want to be able with several definitions of good.

Furthermore it is useful to consider other possibilities for the Macro Planner. Therefore the main question of this thesis is: *How can a more robust supply chain planning be created and what are the differences with the current supply chain planning, and what else can be suggested to increase the attractiveness of the Macro Planner?*

1.2.2 Problem description and thesis objective

The Macro Planner is an advanced planning software application which can be used both for constructing a sales and operational plan and to make strategic decisions at most a few times per year. The generated high level strategic decisions will probably have an impact on a long horizon for which there is still a lot of uncertainty in the sales demand and it will cost effort to reverse them.

We can distinguish the decisions which are made before information on the demand is revealed and decisions which are made after the information is revealed. The first category consists for example of planning the inventory levels, allocating quantities to subcontractors and determining what the product mix will be. We know the optimal supply chain planning with these first-stage decisions is excellent for the given input of sales forecasts, but now we recognize the possibility of a different demand we want also those first-case decisions to be good in that case. This will be the main subject of study of this thesis.

For each planner the notion of good may have a different meaning, depending on her risk aversion. Therefore we have to consider more robustness measures and therefore also more models. We will do a literature review to explore the options and choose the most likely for Macro Planner users. Then a comparison is needed with the normal supply chain planning. Only if the outcome is considerably different, it is worth the effort for planners to adopt robust planning.

Besides there are also other suggestions to improve the attractiveness of the Macro Planner. Recently reducing the amount of carbon dioxide emission became a business goal for many companies. According to [16], maybe an already outdated survey in these days, more than 34% of the managers have sustainability on their agenda. Greening of supply chains is an interesting topic with for example much research on reverse supply chains. However little attention is paid to the effect of the planning/design decisions on the emission of carbon dioxide. It is likely that in some cases a supply chain planning affects this amount, for example by the means of transportation chosen in the supply chain. Therefore the possibilities of containing measures on emission have to be considered. The question how to measure and present the emission is of course essential for making a planning based on CO_2 emission. This question in itself is actually not an econometric issue, but just very important for the position of the Macro Planner.

In the Macro Planner it is currently possible to create multiple scenarios for the supply chain design and there also the capacities of stocking points and units are given as input. If a planning for a long horizon is made it may be a wise decision to increase or decrease the capacity during the horizon. However, currently this has to be stated manually. As a supply chain planning itself mostly triggers the needs for capacities it seems very natural to try to include the choice for capacities in the optimizer. Possibly this yields extra benefits when CO_2 emission is taken into account as maybe capacity expansion of clean machines can be observed.

1.2.3 Relevance of problem description

For Capgemini it is interesting to study these questions related to the Macro Planner as they have some projects in which the Macro Planner is implemented and are still interested in new projects in which it can do so. This thesis could help Capgemini to offer additional value to the customer in such projects. Furthermore the ideas on robustness may be inspiring for similar models.

There are a lot of related problems which deal with decision making under uncertainty. All production problems represented by mathematical programming formulations in which the right-hand side parameters representing demand are uncertain with its uncertainty represented by a discrete sample space for example. All proposed solution methods, and thus also the options which are not earlier studied in literature, can be applied to those problems as well.

1.3 Outline

In the next chapter, chapter 2, a description of the Macro Planner is given. It is important to read this chapter as all other topics are related to the Macro Planner and the terminology is introduced in this chapter. With the Macro Planner at hand it is easier to read this thesis, but with this chapter and if necessary the appendix it should also be possible. Chapter 3 makes clear which decisions in the supply chain are made by the optimizer and which supply chain design decisions are made manually. Furthermore the planning decisions and the business goals are related in a matrix. This will also give more insight in which business goals are affected when the supply chain planning process is revised.

Chapter 4 is the core of the thesis. Here the main ideas of the thesis are worked out. Instead of taking one sales forecast for granted, we want to consider multiple possibilities for the sales demand when creating the supply chain planning. By a literature survey the possible goals to strive for in robust planning are examined. By making the assumptions that we regard as plausible and selecting from all possibilities the robustness goals we think planners working with the Macro Planner strive for we present the three models mentioned under approach in this proposal. These models can be solved by feeding it to CPLEX or solving it by a decomposition method. For a decomposition method there are a number of options and it is not clear which method will lead to the lowest computation time observing that all methods lead to an exact solution.

Chapter 5 will then give the results of our robust planning methods in the Macro Planner framework. As it will be very important to make customers enthusiastic on robust planning we will also elaborate on the graphical representation. Before studying the results a number of checks are performed to verify the implementation.

In chapter 6 the consequences of the decomposition methods are examined. For various datasets we compare normal solutions with robust solutions and evaluate their solution time on the possible solution methods.

In chapter 7 we will propose ways to work out the ideas of capacity extension and CO_2 emission. These ideas are not implemented and therefore this chapter can be seen as a large chapter to describe recommendations for future research to increase the attractiveness of the Macro Planner except from enabling robust planning.

Chapter 8 gives the conclusions of this thesis and recommendations for further research. In the appendix A all details on the model of the Macro Planner are given.

Chapter 2

Macro Planner Environment

2.1 General Information

The Macro Planner is an advanced planning software application of Quintiq which can be used both for constructing a sales and operational plan and to make strategic decisions at most a few times per year. Most Macro Planners are currently implemented for metal producers. As Quintiq was already a market leader for scheduler products in the metal industry, this is the logical industry to be an early adaptor. However the Macro Planner is suited for all production industries. To make a supply chain planning, input data is needed to get the planning as output.

The inputs of this software application are:

- Forecasts for the finished products of the supply chain, usually on a monthly or yearly basis, and the current outstanding orders which are accepted in the past. This is processed by demand netting to so called sales demands for the planning. The goal of the planning is to react on this demands as good as possible.
- The supply chain design, which can consists of plant-units, machines existing in plants, stocking points, transport-units, routings from stocking point(s) via a unit to stocking point(s), shift times for machines, current inventories and so on. It is the total current structure of the supply chain. This is framework the company has disposal of to support the reactions on demands.

For each Quintiq application, including the Macro Planner, there is, according to the Quintiq way of working, also a Quintiq Business Analysis document made: [33], which will be denoted by the abbreviation QBA. It is a technical description of the application and it serves as input for the next modeling phase done by the Quintiq specialist when implementing the solution. To get insight in the framework in which we operate, the supply chain design, we quote the QBA where the design is described in 6 steps for a metal manufacturer:

1. "Define supply chain skeleton: defining all production units (factories, suppliers, transport), the stocking points and the connections between the production units via stocking points.
2. Define sales hierarchy.
3. Define unit availability: defining the availability in the various periods for all units with constrained capacity.
4. Define products hierarchy: defining the high level products, e.g. Hot rolled coils
5. Define product routings: define product routings.

Figure 2.1: Supply Chain Planning in the Macro Planner

6. Define capacities of units and stocking points: external production units such as suppliers may have a maximum capacity (e.g. in number of tons per month), as well as a minimum capacity (long term contracts). These capacity restrictions are defined in this decision."

An example of a supply chain network will be presented for a problem instance in figure 6.1 later in this thesis. Although some terminology will only be explained later in this chapter, it is already clear that many decisions in the supply chain optimization process have already been made in the supply chain design. In advance of the relation between planning decisions and business goals as explained in the next chapter, we can already reveal that the design of the supply chain influences the key performance indicator (KPI) total fixed costs directly which is important for the business goal of profitability. This includes costs of opening and closing plants for example.

The performance which can be reached in the planning is also influenced by the design. To compare the influence of the design on the planning it is possible to create manually different scenarios in which a supply chain design is defined. When a planning is made for two scenarios that only have a very small difference caused by a supply chain design decision, the quality of that decision can be evaluated.

For a scenario the output can be created by manual actions or by a planning algorithm which is the most interesting for this thesis. The output is:

- A planning that determines the quantity to be produced, the way to produce that quantity (use own plants or hire a subcontractor) and which sales demands are fulfilled with that quantity. Also planned inventory levels are a result of this planning.

Once a planning algorithm creates this output it can still be adjusted manually, the final responsibility is always for the planner. In this way the effect of each decision can be perceived by the planner. See 2.1 for an example output of a planning generated by the Macro Planner. A Gantt chart shows demand for products in stocking points in periods and the colors indicate whether the demand or inventory targets are fulfilled by the planning.

From the point of view of the planner the input-output information of the Macro Planner is used as decision support tool for multiple decisions. One could name *optimal inventory policy*, determining an inventory policy which can probably be viewed as determining a safety stock safety stock; *source efficiency*, where to get how many of the raw materials and semi-finished products indicates how contracts with suppliers should be agreed; *product mix optimization*, the planning indicates which products will become the most produced products and possibly the marketing can be adjusted to that result; (*effective supply chain design*), by trying out more scenarios for the supply chain design with for example different fulfillment goals the effect of the fulfillment goals on the cost can be examined.

2.2 Terminology Macro Planner

In the application several terms are used to describe the supply chain. Here the most important terms are listed and explained with the purpose to make the rest of the thesis readable without having the application at hand. A list of terms in alphabetical order is presented below. In some explanations terms are used which are explained itself in another place.

External parties The actual term is quantity-based units, as external parties like suppliers or subcontractors have in each period a minimum and a maximum amount of goods to be processed. The intuition is that it is the result of a contract. Also a minimum or maximum capacity per product can be defined.

Fulfillment goals For each sales segment it is possible to define for a period what percentage of demand should be fulfilled and this is called a fulfillment goal. For a fulfillment goal also a cost per percentage of demand less fulfilled has to be specified. This will be taken into account in the optimizer. It is also possible to specify such an fulfillment goal specifically for one product.

Machines The actual term is time-based units as the capacity of machines is measured in operating time in each period. Like external parties machines are an example of units. In all periods for machines fixed costs, maintenance time, a shift pattern defining the number of hours the machine is available per week and the number of identical copies of this type of machine is defined. The used capacity in hours is derived from the quantities of supply routings planned on the machine. On basis of the used capacity and the available capacity the utilization per period can be obtained.

Periods The periods (years/quarters/months/weeks/days) which are considered in the planning. All periods have a number of days, a start date and end date. Also a number of periods before the 'present period' can be taken into account in the planning as satisfying the demand for the start periods may require production steps in earlier periods. The start of the planning horizon is defined in the supply chain design.

Plants Actually this is an non-capacitated unit. For each plant a set of machines which it houses is given. When a product routing is defined on a plant obviously only machines from the plant can be used for the operations. There are connections between plants and stocking points and these connections are used to check whether it is possible for a product-in-stocking point to be an input or output for a product routing.

Products In the supply chain several products are used from raw material to finished products. It is possible to classify the products in this way (raw materials - intermediate products - finished products as example of a product hierarchy). Each product is assigned a set of stocking points in the supply chain in which it can be stored.

(Product) Routings This non-standard term is very important for the Macro Planner. A routing is the representation of the process in which from certain products in stocking points (PISPs) certain products in stocking points are created in a plant or by a subcontractor. So a routing is defined on a unit being a plant or an external party. A routing normally has one ore more product-in-stocking points as input and one or more product-in-stocking points as output. For each product-in-stocking point that is used as input the share of its quantity relative to the total input quantity should be given. Similarly for each product-in-stocking point that is used as output a share should be given. In a production process the total input quantity does not have to be equal to the total output

quantity. There can be some waste in the process. Therefore each product routing has a yield, the proportion of output quantity in relation to the input quantity. One side note is that there can be product routings base on external suppliers that have no input but have as output the supplied product in a stocking point.

Inside the process represented by the product routing for a plant there are product routing steps, steps that have to be made sequentially to transform the input to the output. In these steps product routing operations are defined, operations that have to be performed simultaneously. Most frequently each product routing step has only one product routing operation. For the product routing operation the machines on which they can be performed are known and a production speed is assigned to perform the product routing operation on each of those machines. Maybe more insight in the concept of a product routing can be given by figure 2.2.

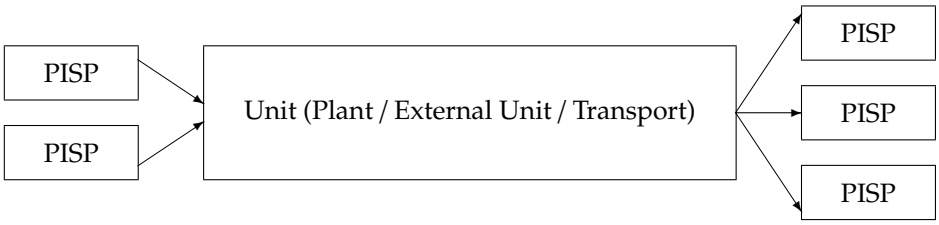

Figure 2.2: Product Routing Diagram

Sales demands For each sales segment and each period and each stocking point there is a sales demand with a quantity, a priority and a revenue per fulfilled quantity. This is the result of sales forecasts and orders, by the demand netting procedure explained in section 2.3. Fulfilling these sales demands is the idea of making a supply chain planning.

Sales segments Selling segments can be defined as the sales forecasts of the sales department are sometimes made for a specific selling area. These areas are called sales segments and can also be non-geographical determined, large customers for example. Outstanding orders, fulfillment goals en sales targets are also all defined on a selling segment.

Sales targets For a period there can be a target of the number of products to sell in a sales segment. This is specified by a starting date, an ending date, a minimum capacity and a maximum capacity. Maybe sales could be undesirable from a marketing point of view to rise above or drop below a certain level in a sales segment and it can be taken in account in the optimizer settings.

Stocking points Products are stored in stocking points. Stocking points are the start and end of a product routing and sales demand is defined on a stocking point. For a stocking point in each period there is a starting inventory and ending inventory measured in number of goods. Also a maximum capacity can be defined and this can be split out to a maximum inventory per product. There is also a 'planned capacity' which is the inventory target for the stocking point in a period. In the optimization parameters, the weight for a penalty on the deviation from this planned capacity can be defined.

Supply routings A supply routing is a copy of a product routing but now also an output quantity (and with that the input quantity), an end date (and with that

Quantity	Day
20	2
20	4
30	6
30	9
15	12

Table 2.1: Orders

Quantity	Startday	Endday
45	1	5
35	6	10
10	11	12

Table 2.2: Sales forecasts

the start date) are defined. The output of a supply routing creates a supply. A diagram of a supply routing is given in the appendix on the mathematical model, see figure A.1.

Stocking points All stocking points have per period a starting inventory and an ending inventory measured in number of goods. They also have a parameter planned capacity, the inventory target, and a maximum capacity in number of goods. Also for each period a maximum and a planned capacity per product can be defined. In the optimization parameters, the weight for a penalty on the deviation from this planned capacity can be defined.

Now we will give in the next section a description of demand netting which deduces the sales demands and in the next chapter we will look at the planning decisions.

2.3 Demand Netting

At the supply chain design sales forecasts and orders are imported from ERP-systems. Orders are already known and have one day at which they should be delivered. Sales forecasts are from the forecasting department and they have an end and a start date and a sales segment in which the sales would take place. As mentioned in the model itself discrete time periods are used, so these orders and forecasts should be converted into demands per period in the model, the so-called sales demands.

An illustration of the process can be seen by tables 2.2, 2.1 and 2.3 and figure 2.3. Sales segments are ignored, but the procedure described here should be done for each sales segment. In the example each period consists of 3 days. An important concept in demand netting is the planning feedback horizon. Here the planning horizon consists of two periods. It means that the two coming periods are so close to this moment that it is not possible anymore, or at least it is not expected, that new orders will be made in this period. The once made sales forecasts are useless for these two periods and we take the order quantity as demand quantity. In the example this can be seen for the first 6 days. For the remaining two periods the maximum of the quantity of the confirmed orders and the quantity of the sales forecasts is taken. When the ordered quantity exceeds the sales forecast, then it is clear the sales demand will be at least the ordered quantity. Otherwise the sales forecast is still taken as valid. See for this process the remaining 2 periods.

The revenue per unit of a sales demand is determined in this demand netting process by first summing the sales forecast quantities times the revenues per quantity

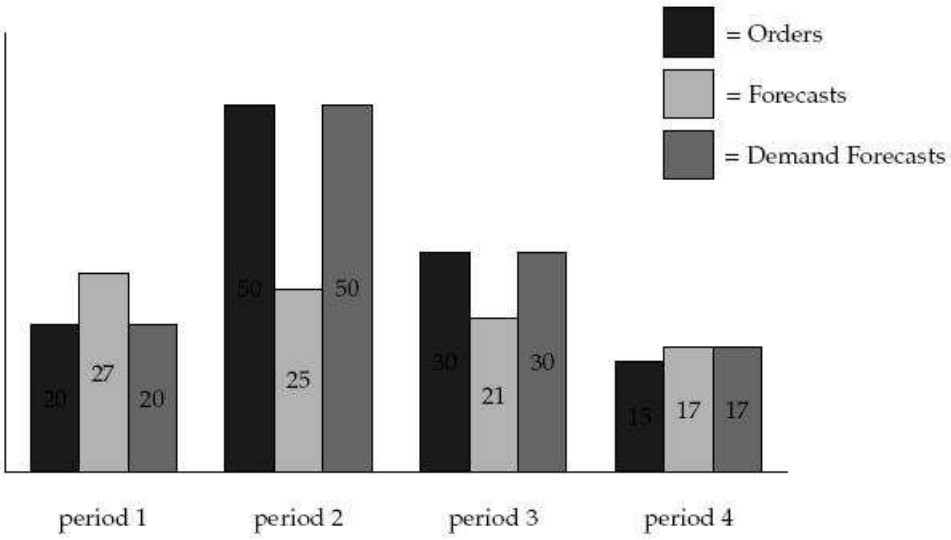

Figure 2.3: Demand Netting Example

Period	Days	Orders	Forecasts	Sales demand
1	1-3	20	$(3/5) \times 45 = 27$	20
2	4-6	$20 + 30 = 50$	$(2/5) \times 45 + (1/5) \times 35 = 25$	50
3	7-9	30	$(3/5) \times 35 = 21$	30
4	10-12	15	$(1/5) \times 35 + 10 = 17$	17

Table 2.3: Netting results

of these sales forecast if the total sales forecast quantity is higher or by the order quantities times the revenue per unit of these orders otherwise. The revenue per unit follows by dividing this revenue by the total quantity of the sales demand. This procedure is a bit unnatural as will be shown in chapter 5 when we will do the implementation of a robust planning method.

By intuition it can already be felt that the way demand netting is taking place influences the results of the Macro Planner. In theory it would be best for example if the sales forecast of the sales department is perfectly up-to-date such that a forecast given the current ordered quantity is made. For example when there is already a period outside the planning feedback horizon in which orders exceed the sales forecast, the expectation that sales demand will be the now ordered quantity may be inaccurate.

Chapter 3

Planning and Business Goals

3.1 Supply Chain Planning

As mentioned in the previous chapter, a supply chain planning itself can be obtained by manual actions or an optimizer. A manual action in the planning is to set a quantity of a supply for a product in a stocking point in a period, which will be denoted as 'pispip'. This is also the term used in the mathematical model. Setting such a quantity is usually done when there is some sales demand for that pispip. Creating supplies gives costs from the supply routings used to create the supply from other products but fulfilling those sales demands will give revenue. Also it could be reasonable to create supplies such that an inventory level at the stocking point will be reached.

When setting such a supply one or more routings of which with some factor larger or equal than zero the output goes to that pispip ('with output in pispip') have to be created. This should satisfy that the sum of supply routings output for that pispip equals the quantity of the supply. The quantity of these supply routings is not unlimited as their operations have to be planned on a finite number of machines which have a maximum available time in a period. Luckily the user is guided to choose a quantity for supply which can be realized by supply routings by an algorithm which calculates the maximum quantity supply routings can offer. See the screenshot in figure 3.1.

When a planning is automatically created and supplies are determined by the optimizer, the maximum capacity of machines will of course be taken into account. Once supply routings have been created they require inputs from pispip's and this generates dependent demand. In the optimizer these dependent demands automatically have to be fulfilled by creating other supply routings which have output in that pispip as otherwise all supplies depending on the supply routings which use this dependent demand are still worthless. This backward chain in the ends uses the product routings based on external product (usually raw material) suppliers which do not generate dependent demand.

The mathematical model for the automatic supply chain planning is discussed in appendix A. Those wanting more mathematical insight are referred to that appendix. In the end the supply chain planning part can be summarized as the decision what quantity of supply is produced, from which sources (supply routings) the supply comes, how supply operations are divided over machines and for which demands (sales demand, dependent demand, inventory demand) the supplies is used for each pispip. Note that it would also be desirable in certain circumstances to determine in the supply chain planning for a machine in a period if it is switched on or switched off, because usually costs are coupled with it. This can now only be simulated by opening and closing a machine for a certain period in the supply chain design. Currently all these decisions are captured by continuous variables and linear expressions such that

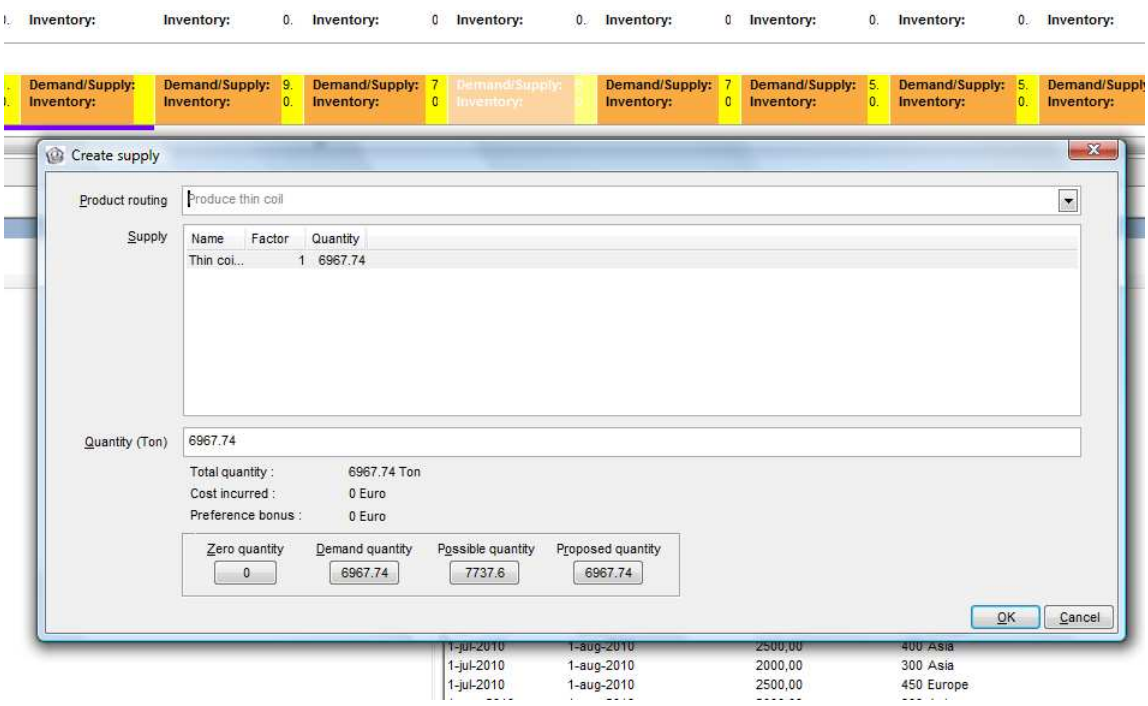

Figure 3.1: Creating a new supply

the problem can be solved with a simplex method in the optimizer.

3.2 KPI Matrix

In section 4.2 of the QBA a KPI matrix is presented. Recall that KPI stands for key performance indicator. This matrix tells which planning decisions affect which business goals and how these relations are captured in performance indicators. We give an extended version here in table 3.1 to get insight in the effects of planning decisions on the business goals. The numbers stand for the following KPI's:

1. Total profit: Total revenue of finished products - Total fixed cost - Total cost of supply
2. Total revenue: Sum of revenue of fulfilled demand

	profit	inventory	satisfaction	productivity
Supply Chain Planning				
Quantity of supply to produce	(1),(2),(3),(4)	(4),(5)	(6)	(8)
Sources used for supply	(1),(3)	-	-	-
Operation allocation to machines	(1),(3)	-	-	-
Demand served by supply	(1),(2),(4)	(4),(5)	(6)	-
Inputs for Supply Chain Planning				
Demand netting	-	-	-	-
Fulfillment goal, sales target	-	-	(6)	(8)
Inventory target decision	-	(4)	-	-
Supply Chain Design	(7)	-	-	-

Table 3.1: Planning Decisions versus Business Goals

3. Total cost of supply: Sum of cost of the supply routings for the supplies.
4. Total inventory cost: Sum of all cost for having inventory in a pispip. This is calculated per pispip as

$$\begin{aligned} CostOfInventory &= InventoryLevelStart * (YearlyInterestRate/100 * DaysOfPeriod/365) \\ &+ 0.5 * (InventoryLevelEnd - InventoryLevelStart) \\ &*(YearlyInterestRate/100 * DaysOfPeriod/365) \end{aligned}$$

   So the total inventory cost is related to cost of inventory for finished products and inventory for work in progress. When the optimizer is making the planning decisions it uses a simplification to represent inventory cost in it's goal function.
5. Fulfilled target levels: The percentage of inventory target fulfillment.
6. Fulfillment: The percentage of demand for finished goods fulfilled.
7. Total fixed cost: Sum of fixed cost per period of units and the opening/closing cost of units.
8. Total volume: Total quantity of fulfilled sales demand.

From the division of supply chain planning decisions it can be seen that even making separate small decisions in the supply chain planning business goals are affected. If we set the quantity of supply to be produced we affect many business goals but leave also still a lot of possibilities open to affect these goals.

3.3 Non-supply-chain-planning Decisions

The supply chain planning does not deal with some of the planning decisions for the supply chain. These other planning decisions can be supported with the Macro Planner by creating different scenario's in which, usually while keeping other things equal, an aspect of the inputs for supply chain planning or the supply chain design is changed. The KPI's of the different scenario's should be compared.

For the planning decision of inventory target it is clear that costs will decline when the inventory target declines. In the Macro Planner the difference in costs and emission for lowering the inventory target can be estimated. However no information about the advantage of such a safety stock can not be obtained because the randomness of demand is not taken into account. Therefore another calculation should be made for this aspect to take the right decision. This note is meant to emphasize that the application is primary meant for supply chain planning and it can only help in supply chain design.

Another planning decision in the supply chain design is whether capacity should be expanded and at what time moment this should take place. Now it is possible to try some scenario's in which in the knowledge base some change in capacity is made. The difference in costs and other KPI's of the supply chain planning can than be compared with the costs of such a capacity change which the user should estimate. Later in the thesis some ideas are presented about considering this planning decision in the optimizer of the supply chain planning.

Chapter 4

Robust Planning Methodology

In this chapter we will first give a literature overview of robust optimization. Discussed are among others types of uncertainty, robustness measures and types of robustness measures. Then we consider the assumptions to be made to use robust planning in the Macro Planner with a critical view on their sense of reality. After that we will propose three models. These models are given in a general form and we specify how this general form should be interpreted in the Macro Planner environment. A number of solution methods are proposed subsequently on the models in their general form. After we presented our models and solution methods we give a summary of the benefits for the customer at the end of the chapter.

4.1 Literature

Robust optimization is a branch of mathematical programming with the goal to find solutions which perform well under any realization of the yet uncertain parameters, though not necessarily optimal in any. Note the difference with conducting a sensitivity analysis for a LP-solution to measure the sensitivity to the uncertain parameters. A sensitivity analysis is only a post-optimization tool while robust optimization takes the uncertainty into account during the optimization. Sensitivity analysis is reactive instead of proactive. Moreover in general it will only give information about a very small interval of a parameter. In supply chains there is a lot of uncertainty in parameters. Consider for example demand, transport prices, revenues, raw material prices, inventory costs, possible new production technologies and management decisions. Therefore it is not surprising to see many articles in robust optimization considering a problem in supply chains. First we will elaborate on the different forms of robust optimization.

At first the subfields of robust optimization can be split on the type of uncertainty for parameters. At the one hand there is *continuous*, interval-based uncertainty and at the other hand there is *discrete* uncertainty when only a finite number of realizations are possible for a parameter which is also called the scenario approach. For continuous uncertainty usually there is no possibility to evaluate an expectation of the goal function value or other possible robustness values. Such an expectation would be $\int_{\omega \in \Omega} F(x, \omega) \mathrm{d}\mathbb{P}(\omega)$ if x denote the decision variables. Techniques based on Monte Carlo sampling are therefore common in stochastic programming. In [28], a standard work on stochastic programming, multiple techniques available for continuous uncertainty can be found. Note that when there is a correlation between the realization of parameters, it is not possible to define intervals for every single parameter but ellipsoids to model the sample space of two variables. An intersection of many ellipsoids can then give the total uncertainty region. However in this thesis only discrete uncertainty is considered because it is most relevant for

the Macro Planner. Throughout this chapter we will use the term goal function for the goal function of the problem for one specific scenario, one specific realization of parameters. The term objective function is reserved for the goal function of the overall robust optimization problem, using results from all scenarios.

Although frequently mixed up, in principle also a distinction can be made between *risk* and *uncertainty*. Risk is meaning that a probability distribution is available for the sample space of the uncertain parameter, while under uncertainty there is no information about probabilities. It is clear that in an uncertainty situation there are less possible robustness measures. The most natural suggestion in that situation is always to use a measure based on worst-case objective value. While in this thesis a distinction is made between situations with or without probabilities, the term uncertainty is always used in favour of risk.

[34] gives an easily readable overview of stochastic linear programming models, robust optimization models in which no integer variables are allowed. Emphasized is that under uncertainty wait-and-see approaches, when a solution is determined for each possible scenario and from that a first-stage response is constructed, do not bring anything in the form of a decision-making framework. Those solutions may all be infeasible for other scenarios. Under uncertainty you have to use a here-and-now approach for a subset of the variables. In this tutorial treated decision-making frameworks are amongst others *recourse models* and models with *probabilistic constraints*.

Stochastic programming with recourse is said to be introduced in [11]. In these (two-stage) recourse models first-stage and second-stage decision variables are used, also called design variables and recourse variables or control variables. By definition are first-stage decisions made before information is available and second-stage decisions after information is available. Second-stage variables are also called recourse variables as after the information is available the decision maker usually has recourse to an adaptation of the initial policy, for example demand is backlogged. These decisions can be made on wait-and-see basis. When it is not possible to set the first-stage variables such that for every scenario there is a feasible choice of second-stage variables, the recourse model is said to be infeasible. On the contrary a recourse model is said to be *complete recourse* when for every choice of the first-stage variables there is in every scenario a feasible choice of recourse variables. A weaker but still important characteristic is relatively complete recourse, in which there is in every scenario a feasible choice of recourse variables for every feasible choice of the first-stage variables.

In models with probabilistic constraints instead of defining control actions with certain cost for every scenario in which control constraints have to be satisfied, here the possibility of infeasibilities is just accepted as long as it does not occur too much over the scenarios. This is appropriate for example in emergency medical services when a call has to be answered within a certain period of time by a certain probability. The definition already indicates that the control constraint can be violated by a certain probability. In the Macro Planner there are hardly any hard constraints such that this approach is not useful in this thesis.

These different models cause us to separate robustness into *model robustness* and *solution robustness*. A solution that performs well under all possible realizations means in solution robustness that the solution is always 'nearly' optimal. In model robustness it means that the solution is nearly feasible in all realizations. An example to show that a solution can be infeasible: A solution in which a certain amount will be sold does not hold anymore when the demanded quantity is lower than that amount when no control action is available. In recourse models solution robustness is considered and in models with probabilistic constraints the model robustness. A combination of these robustness measures can lead to a new model. In [23] a so-called RO (robust optimization) framework is introduced, a model in which the goal function promotes solution robustness as well as model robustness. That last term

is based on the infeasibilities in the scenario-dependent control constraints. This is different from models with probabilistic constraints in the sense that there is no hard constraint for the number of infeasibilities but a term in the goal function and it is different from recourse models in the sense that it is not possible to take control actions which lead to a certain cost. Instead it is not clear how undesirable an infeasibility is. By varying the weights in the aggregate objective function a set of solutions can be obtained in which a trade-off is made between solution quality and model robustness.

Considering the solution robustness there is a range of robustness measures to which the model can be optimized. As we will study a recourse model in this thesis, it is interesting to list which robustness measure possibilities the literature offers. In the next paragraphs an enumeration is made and some advantages and disadvantages are examined.

The most straightforward is the *expected* goal function, in which a weighted average is taken over the possible realizations of the parameters. Here it is important to note that the solution for this model when the expected goal function is taken over the possible parameter realizations is in general very different from the goal function value of the problem when the expected parameter values are taken as input. This straightforward measure may not be totally adequate as usually utility is considered as primary goal for the homo economicus and a Von Neumann-Morgenstern utility-function is not necessarily risk-neutral. A possibility based on the expected goal function measure to allow for non risk-neutral preferences of the decision maker is to introduce a *mean-risk* robustness objective. In the objective function the expected goal function and a dispersion measure of the goal function over the demand realizations are taken into account. This is also very usual for interval-based uncertainty, see for example [3], where different measures for risk are considered and their computational suitability is evaluated. Variance as measure for risk may be computationally impractical, but it also does not take skewness of the goal function values into account, which may be interesting. Besides it is hard to give a clear interpretation to such a mean-risk objective.

Another possibility is max(min) the *worst-case* objective value for a maximization (minimization) problem. As the worst-case objective value has to be found by maximizing or minimizing over the possible realizations of uncertain parameters. Therefore such a problem becomes a maximin (minimax) problem. After finding a solution, it is possible to say that the solution will be at least as good as some value over all scenarios, which gives a concrete interpretation of the goal of robust optimization, finding a solution that performs well under all parameter realizations. There are both advantages and disadvantages to this approach. The advantage of this approach is that there is no need for probability distributions. A disadvantage is that the solution is dominated by the worst-case scenario.

A closely related robustness measure is *absolute regret*. For each parameter realization the objective value of the solution is compared with the optimal objective value possible when the information about the parameters would already have been available. The difference between these two values is called the regret. The objective is to minimize the largest regret, which also makes it a minimax problem. The idea behind this measure is that the performance of the decision maker can be measured in this way. After the realization of demand it is possible to see what the profit is and to calculate what the profit could have been. The lower the difference, the higher is the perception of the performance of the decision maker. The value per scenario over which the maximum is taken in fact only differs by a constant, the optimal solution under perfect information, from value per scenario over which the maximum is taken in case of max(min) worst-case objective value, such that solution techniques are similar. Related to absolute regret is the robustness measure of *relative regret*. For this measure the regret is divided by the optimal value under perfect information to get the relative regret. The largest relative regret is minimized. This is more interesting

when the performance of the decision maker is measured by the percentage of sub optimality of the chosen solution. The best robustness measure to choose depends on the information available, some measures need probabilities, and the goals and degree of risk-aversion of the decision maker.

All these maximin or minimax problems can be completely dominated by the worst scenario. This may lead to undesirable effects for example when the worst scenario is very different from the other scenarios and the performance of the solution at the other scenarios is very bad. Therefore in [12] an approach was introduced called *α-reliability* in which not all scenarios are taken into account. Under the assumption of the availability of probabilities the minimum regret value over an endogenously selected set with combined probability larger or equal than α is maximized. The point of Daskin et al. is that for example an airport is never sized for an average day or for a peak day. It does not treat the possibility that regrets in the tail are excessively higher. In this thesis a recommendation of that paper in the case of unavailability is used, *N-reliability*, and not applied on regrets but instead just on the goal function. This is the same idea, but now the endogenously selected set should contain exactly N scenarios. This set and the solutions for the variables are chosen such that the minimum goal function of this set is maximal. In a further paper [8], Daskin made another proposal based on endogenously selected scenario sets to eliminate worst case scenario. The mean-excess regret model takes the mean of the regret. This is clearly easier to calculate but the reason why to exclude worst-case scenarios in this case is not completely obvious anymore.

Also proposed in [25] is a robustness measure which combines expected goal function value and the sensitivity of the solution to changes in demand for a model in which the expected demand is taken. The dual values for the variables in the optimal solution of the constraints involving the uncertain parameters can be taken as measure for sensitivity. However the general idea is that taking the expected demand, still with this extra term in the goal function, does not result in very robust solutions.

For the recourse models usually the solution robustness and model robustness are the criteria were a robust optimization focuses on. However sometimes it is also useful to have robustness in the solution itself, i.e. the second-stage variables should be 'nearly' equal over all scenarios. In [9] and [10], papers from 1987 and 1990, the differences of the solution of the scenarios are reactively evaluated after the optimization. In the years from then on several papers were written in which this robustness in the second-stage variables was taken into account in the objective function, mostly with a multi-criteria objective as a result resembling to [23].

In 1997 [38] the concept of *restricted recourse* was introduced. This is an objective that stands apart from model robustness and solution robustness and could be seen as related to a new business goal called recourse robustness. In a constraint the differences in second-stage variables is can be measured by the mean of the Euclidian norm per scenario taken over the difference between the second-stage variables and the mean second-stage variables over all scenarios. One of the alternatives is to take the maximum of that Euclidian norm. These measures were restricted to be below a certain threshold. When this threshold is lowered, the constraint is more strict and consequently the cost of the solution (if the objective deals with costs) will become higher. By solving the program for multiple decreasing values for the threshold, a Pareto-curve can be made to compare costs and robustness in the solution. The model can be solved by an iterative procedure in which a solution is made without restrictions on recourse differences and based on that solution new bounds are added for the recourse variables after which a new solution is made and so on. They proposed multiple procedures and [5] tried to improve on that. In 2003 [6] appeared in which the concept of *limited recourse* is introduced. There the focus is on models in which in the first-stage an estimation is made and in the second-stage this estimation

is adapted. In reality here only the variables of the first-stage and the variables of the second-stage for this scenario will be observed. Now it would be very desirable if the observed second-stage estimation is still very similar to the first-stage estimation. No constraints on the difference between the second-stage variables is required, but a constraint on the difference between first-stage variables and second-stage variables would be very useful. This is the concept of limited recourse. Note that this will usually imply that the second-stage variables are still not very different from each other as they all are close to the first-stage variables. The procedure to tighten the recourse limitations more and more is similar as in the previous articles. This article seems to be the latest state-of-the-art.

Note that many optimizations are multi-objective optimizations. The usual way to deal with multiple objectives is to construct a single aggregate objective function. For each objective a scalar weight is defined. Those scalar weights are determined by the optimizer-user and therefore this kind of objective functions are subjective. Another disadvantage is that in this way not all non-dominated, or Pareto optimal, solutions are found. Pareto optimal solutions, named after the Italian economist Vilfredo Pareto, can be split in strong and weak Pareto optima. A strong Pareto optimum is a solution in which no objective can be improved without worsening any of the other objectives, while in a weak Pareto optimum no objective can be improved without not improving any of the other objectives. As can be seen in chapter 4 of [26] only solutions on the convex part of the Pareto front can be found. To overcome these disadvantages sometimes other ways of optimization are tried, belonging to the more general class of vector optimization. Thereby vector optimization under uncertainty can be seen as a more general class of optimization under uncertainty. Recently this topic gained interest, see for example [13]. It is a very interesting field, but this thesis will deal with a single aggregate objective function.

4.2 Assumptions

At implementing methods for robust optimization in the Macro Planner a few assumptions are made. These assumptions will be listed here and also a remark is made on the practical quality of these assumptions. In general the assumptions are chosen in a way to hold as much as possible for Macro Planner users such that the proposed robust planning options will also have practical value. In that we focus on the application of the Macro Planner to support long term strategic decisions, because then a planning horizon is considered for which the parameters can not be established with certainty. When the Macro Planner is used to find a Sales and Operations plan, a lot more information is available and robust planning makes less sense.

An assumption regarding the data uncertainty is that we only study uncertainty in the demand forecasts. While, as said, in supply chains there are more sources of uncertainty, we decided to focus only on demand, because about that probably most information is available at companies. Another assumption is that the uncertainty is in the form of scenarios, like HIGH, MEDIUM and LOW, not necessarily meaning a scenario contains realizations that are strictly higher or lower than realizations in another scenario. In every scenario there is a quantity for the expected demand per period and sales segment, the sales forecasts. According to consultants with Macro Planner experience this is more realistic than intervals for the demand. Moreover correlation in demand is expected and it is easier to represent this with discrete uncertainty. We will present both robustness measures which require risk and which also suite uncertainty. So it is not really necessary to have probabilities for the scenarios.

In case there would be independence among the uncertain parameters and there is a so-called full-factorial scenario design of data uncertainty, a set of total scenarios

have to be created. The size of this set grows exponentially with the number of uncertain parameters. Therefore this is not really the ideal situation, we rather would like to have only a limited number of total scenarios. However we still try to design our solution methods such that it can handle many scenarios.

Furthermore we assumed that the object value has a 1-to-1 relation with the quality of the decisions made on the supply chain planning solution. Implicitly this assumes that the planner used exactly the right parameters and weights in the goal function. On average this assumption may be valid as sometimes parameters, the 'certain' parameters, are overestimated and sometimes underestimated. For individual cases this assumption will probably not be true, but without this assumption no model would be possible.

We assume that the Macro Planner is run once and the decisions made for the planning horizon cannot be reversed or cancelled, a so-called here-and-now approach. In this way our models are two-stage recourse models. In practise the Macro Planner is run once or a few times per year to support decisions at the highest strategic level. This means that maybe multi-stage recourse models could model situations with a planning horizon of multiple years better. This would be a topic for further research. For more information about multi-stage recourse models we refer to [35].

The assumption that the decisions cannot be reversed or cancelled is unrealistic from practical perspective. In some sectors investments with a very long cost-recovery-time are made, like building oil rigs in the oil sector, and then this is of course very valid. But in fact in most businesses companies always have their possibilities with an active management style to react by adaptations in the strategy on the highest level when demand changes. Contracts can be adapted, extra subcontractors can be hired, new infrastructure can be made etcetera. That is also the reason that in business often the policy is to make a planning only on the most likely scenario and analyze the consequences when demand is different. When there is an indication that the demand is different from the most likely scenario new measures can be taken. Still our model with this assumption is interesting from practical perspective as the scrap prices we use can also be interpreted as costs for adapting strategies.

In fact this possibility for adaptations by the management hints at the appropriateness of restricted recourse to model robustness when variables which are now marked as scenario-independent variables will be marked as recourse variables. However a model based on it would require more information about which strategy decisions can be adapted in what way. It is unlikely that accurate information is available on that and therefore we still see this model as a very valuable model.

Also about the decisions which can be made after information is revealed assumptions have to be made. When taking all variables on which the high level strategic decisions have to be made equal across all scenarios it means that the quantity of a product in each stocking point in period is equal across the scenarios. At the stocking points having sales demands which are scenario-dependent now decisions have to be taken in each scenario. The assumption is that the sum of quantity of all sales demand of a product that is fulfilled at a stocking point in a period equals in every scenario the supply plus the quantity that was carried forward from the previous period minus the quantity that is carried forward towards the next period and the quantity of dependent demand to be fulfilled (all first-stage variables). So that total sum is always equal, but the division, how that quantity is allocated to the different sales demands, can be chosen differently in each scenario. The consequence from choosing an amount of a sales demand that is fulfilled is that in case the actual parameter for the quantity of the sales demand is lower than the fulfilled quantity, the difference is called the over produced quantity. The constraint indicating that fulfilled quantity should be lower or equal than the demand quantity is therefore changed such that the fulfilled quantity minus the over produced quantity should be lower or equal than the demand quantity. The over produced quantity is simply

scrapped.

The scrapping possibility for over produced quantity is introduced to overcome feasibility problems. Otherwise the fulfilled demand should always have been lower or equal to the smallest demand over all scenarios. Now the recourse problem has relatively complete recourse which can be used in our solution procedures. It cannot be called complete recourse because if a produced quantity per pispip would be chosen negatively in the first stage, which is actually prevented by restrictions, no feasible choice can be made for the fulfilled quantities any more.

An alternative option would be to introduce variables for extra carried forward quantity per scenario and charge that with the inventory cost. However in the planning application the end inventory is not valuated. When it is completely uncertain whether there will exist demand after the planning horizon this sounds reasonable. Now it is far from imaginary that once carried forward inventory will also be in inventory at the end of the planning horizon, while in reality these products would be scrapped probably. On the other hand when there is a low demand in the current period and a higher demand than expected in the next period it is completely unnatural to scrap your products. This makes the chosen assumption quite debatable and probably a combination of the possibility to scrap and the possibility to store over produced products in inventory would be best. Finally note that we also assume it is always possible to sell your products against a scrap price that also possibly is 0 or negative.

4.3 Models

For the Macro Planner in accordance with our assumptions we will work with a two-stage recourse model, in which fulfilling the demands and scrapping over produced products are the recourse actions. Then we chose three robustness measures to model and implement the maximization of the expected goal function, maximizing the minimum goal function over the scenarios and N-reliability. Note that the first measure only makes sense when probabilities are available, while the other two do not need them. First the general form of the models is shown.

For the case of optimizing to the robustness measure of expected goal function value, we have this model:

$$
\begin{aligned}
&\max\ cx + \sum_{\omega\in\Omega}(p_\omega * d_\omega y_\omega) \\
&\quad \text{st } A_1 x \le b \\
&\qquad A_2 x + B_\omega y_\omega \le b_\omega \qquad\qquad \forall \omega \in \Omega \\
&\qquad x, y_\omega \ge 0
\end{aligned}
$$

Here x represents the scenario independent variables and y_ω are the scenario dependent variables. These are multiplied by parameters in the objective function. Later in this section we will point out why the coefficients of the variables y_ω are not equal for all forecast scenarios. In the constraints we have b_ω as right hand side, parameters which are scenario dependent. This is natural as the constraint that fulfilled quantity minus overfulfilled, to be scrapped, quantity should be lower or equal than parameter $\text{Quantity}_{sd,\omega}$. Obviously this does not mean that every parameter in vector b_ω is scenario dependent. An example is the quantity of a sales target. Some coefficients of scenario dependent variables in the constraints are also scenario dependent as will be explained later. From this model we will get the optimal first-stage decision variables x and the optimal second-stage decision variables y_ω for all scenarios.

When the worst-case goal function value is maximized using the same terminol-

ogy the model will look like this:

$$
\begin{aligned}
\max \ & \min_{\omega \in \Omega}(cx + d_\omega y_\omega) \\
\text{st } & A_1 x \leq b \\
& A_2 x + B_\omega y_\omega \leq b_\omega && \forall \omega \in \Omega \\
& x, y_\omega \geq 0
\end{aligned}
$$

Note that it is possible and useful to rewrite this into a standard LP-formulation by introducing an extra variable δ. This is the formulation we will use.

$$
\begin{aligned}
\max \ & cx + \delta \\
\text{st } & \delta \leq d_\omega y_\omega && \forall \omega \in \Omega \\
& A_1 x \leq b \\
& A_2 x + B_\omega y_\omega \leq b_\omega && \forall \omega \in \Omega \\
& \delta, x, y_\omega \geq 0
\end{aligned}
$$

If we maximize this model, this objective function subject to these constraints, we get the optimal first-stage decision variables x, the optimal artificial variable δ and the optimal variable $y_{\tilde{\omega}}$ for the scenario(s) endogenously identified as the worst-case scenario(s). However we do not obtain the optimal values for y_ω for all $\omega \in \Omega/\tilde{\omega}$. As long as $A_2 x + B_\omega \bar{y}_\omega \leq b_\omega$ and $d_\omega \bar{y}_\omega \geq d_\omega y_{\tilde{\omega}}$, $\bar{y}_\omega$ is in the set of optimal solutions. As we want to present not only the first-stage decisions but also the consequences for all forecast scenarios, we have to find also the optimal solutions y_ω for those scenarios that are not the worst-case scenarios. We can find those optimal values by solving for all scenarios, $\forall \omega \in \Omega$

$$
\begin{aligned}
\max \ & d_\omega y_\omega \\
\text{st } & A_2 x + B_\omega y_\omega \leq b_\omega \\
& y_\omega \geq 0
\end{aligned}
$$

which will also come back as subproblem of solution methods presented in section 4.4.

When an optimization is made with N-reliablity, the model will look like this with M being a very large value compared with the other possible values for the goal function:

$$
\begin{aligned}
\max \ & \min_{\omega \in \Omega}(cx + d_\omega y_\omega + (1 - z_\omega)M) \\
\text{st } & A_1 x \leq b \\
& A_2 x + B_\omega y_\omega \leq b_\omega && \forall \omega \in \Omega \\
& \sum_{\omega \in \Omega}(z_\omega) = N \\
& x, y_\omega \geq 0 \\
& z_\omega \in \{0, 1\}
\end{aligned}
$$

Similar to the former robustness measure, also now this model can be rewritten, this time to a mixed integer programming form due to the presence of binary variables.

$$
\begin{aligned}
&\max\ cx + \delta \\
&\quad \text{st } \delta \le d_\omega y_\omega + (1 - z_\omega)M && \forall \omega \in \Omega \\
&\qquad A_1 x \le b \\
&\qquad A_2 x + B_\omega y_\omega \le b_\omega && \forall \omega \in \Omega \\
&\qquad \sum_{\omega \in \Omega} z_\omega = N \\
&\qquad x, y_\omega \ge 0 \\
&\qquad z_\omega \in \{0, 1\}
\end{aligned}
$$

The observations of the worst-case goal function do also hold here. Only for the Nth worst scenarios the optimal value for y_ω can be obtained. Be aware that for the N scenarios which have a z_ω of 1 the value of M is not added such that they are the N scenarios relevant for determining δ. For the other scenarios the model

$$
\begin{aligned}
&\max\ dy_\omega \\
&\quad \text{st } A_2 x + B y_\omega \le b_\omega && \forall \omega \in \Omega \\
&\qquad y_\omega \ge 0
\end{aligned}
$$

has to be solved to obtain their optimal values. Again this is necessary to present the decision maker information on the consequences for all possible sales forecast realizations.

To give these abstract models further interpretation we have to decide which variables in the CapacityPlanningAlgorithm of the Macro Planner are first-stage variables and which variables are second-stage variables. To read the rest of this section, the description of the Macro Planner given in chapter 2 does not suffices. It is required to read appendix A.

The chosen division is presented in table 4.1. According to our assumptions most spill variables, like the total quantities supplied and the quantities of a supply routing are structure decisions. On basis of the outcomes of these variables, high level strategic decisions will be taken. For example the variables for carried forward inventory are important for decisions on the design of stocking points. Therefore they need to be fixed over all scenarios.

The other variables which will be chosen after the information about demand becomes clear do not initiate strategic decisions and are therefore free to be chosen optimally per scenario. $\text{FulfilledQuantity}_{\text{sd},\omega}$ is the only spill variable which is made scenario dependent. For strategic decisions it is not important which demand is actually fulfilled if it is for the same product and from the same sales segment and the quantity is dependent of the quantity of demand. A newly introduced variable is $\text{OverProduced}_{\text{sd},\omega}$, which is a recourse variable. If in the second-stage it becomes clear that too much is produced for a stocking point in a certain period it is possible to scrap the product against a scrapping price. That quantity is called the over produced quantity.

Note that the other second-stage variables are derived from the $\text{FulfilledQuantity}_{\text{sd},\omega}$ variables and for them higher demands can have a negative influence on the goal function value. This does not allow us to label a scenario for which another scenario exist with a lower or equal quantity for each sales forecast as trivial when optimizing the goal function value of the worst-case scenario.

First-stage variables	Second-stage variables
$\text{Quantity}_{\text{sr}}$	$\text{FulfilledQuantity}_{\text{sd},\omega}$
$\text{Quantity}_{\text{so}}$	$\text{OverProduced}_{\text{sd},\omega}$
$\text{Quantity}_{\text{ns}}$	$\text{UnfulfilledPercentage}_{\text{fg},\omega}$
$\text{Quantity}_{\text{op}}$	$\text{UnfulfilledPercentage}_{\text{fgp},\omega}$
$\text{MinimumQuantityNotMet}_{\text{ut}}$	$\text{OverFulfilledQuantity}_{\text{st},\omega}$
$\text{MaximumCapacityOverloaded}_{\text{ut}}$	$\text{UnfulfilledQuantity}_{\text{st},\omega}$
$\text{MinimumProductQuantityNotMet}_{\text{ut,p}}$	$\text{OverFulfilledQuantity}_{\text{stp},\omega}$
$\text{MaximumTimeOverloaded}_{\text{ut}}$	$\text{UnfulfilledQuantity}_{\text{stp},\omega}$
$\text{CarriedForwardInventory}_{\text{pispip}}$	$\text{RealFulfilledQuantity}_{\text{id},\omega}$
$\text{SPCarriedForwardInventory}_{\text{spip}}$	
$\text{UnallocatedSupply}_{\text{pispip}}$	
$\text{FulfilledQuantity}_{\text{d}}$	

Table 4.1: Variable division with respect to information

The concrete constraints in these two-stage recourse models are now:

$$UnfulfilledQuantity_{st,\omega} \geq MinimumQuantity_{st} - \sum_{sd \in SD_{sals}} (FulfilledQuantity_{sd,\omega} - OverProduced_{sd,\omega}) \tag{4.1}$$
$$\forall sals \in SalS, st \in ST_{sals}, \omega \in \Omega$$

$$OverfulfilledQuantity_{st,\omega} \geq \sum_{sd \in SD_{sals}} (FulfilledQuantity_{sd,\omega} - OverProduced_{sd,\omega}) - MaximumQuantity_{st} \tag{4.2}$$
$$\forall sals \in SalS, st \in ST_{sals}, \omega \in \Omega$$

$$UnfulfilledQuantity_{stp,\omega} \geq MinimumQuantity_{stp} - \sum_{sd \in SD_{sals} | Start_{sd} \leq Start_{stp} < End_{sd} \text{ and } \sum_{p \in P}(STPonP_{stp,p} \times DinPISPIP_{sd,p,sp,t}=1)} (FulfilledQuantity_{sd,\omega} - OverProduced_{sd,\omega}) \tag{4.3}$$
$$\forall sals \in SalS, stp \in STP_{sals}, \omega \in \Omega$$

$$OverfulfilledQuantity_{stp,\omega} \geq -MaximumQuantity_{stp} + \sum_{sd \in SD_{sals} | Start_{sd} \leq Start_{stp} < End_{sd} \text{ and } \sum_{p \in P}(STPonP_{stp,p} \times DinPISPIP_{sd,p,sp,t}=1)} (FulfilledQuantity_{sd,\omega} - OverProduced_{sd,\omega}) \tag{4.4}$$
$$\forall sals \in SalS, stp \in STP_{sals}, \omega \in \Omega$$

$$UnfulfilledPercentage_{fg,\omega} \geq FulfillmentPercent_{fg} - \sum_{sd \in SD_{sals}} (FulfilledQuantity_{sd,\omega} - OverProduced_{sd,\omega})/Quantity_{sd,\omega} \tag{4.5}$$
$$\forall sals \in SalS, fg \in FG_{sals}, \omega \in \Omega$$

$$UnfulfilledPercentage_{fgp,\omega} \geq FulfillmentPercent_{fgp} - \sum_{sd \in SD_{sals} | Start_{sd} \leq Start_{stp} < End_{sd} \text{ and } \sum_{p \in P}(FGPonP_{fgp,p} \times DinPISPIP_{sd,p,sp,t}=1)} (FulfilledQuantity_{sd,\omega} - OverProduced_{sd,\omega})/Quantity_{sd,\omega} \tag{4.6}$$
$$\forall sals \in SalS, fgp \in FGP_{sals}, \omega \in \Omega$$

$$(1 - IsEarlySupplyAllowed) \sum_{ns \in NS_{pispip}} Quantity_{ns} \leq$$

$$\sum_{d \in Dd_{pispip} \cup Id_{pispip}} Fulfilled Quantity_{d} + \sum_{d \in Sd_{pispip}} Fulfilled Quantity_{d,\omega} \tag{4.7}$$

$$\forall \omega \in \Omega, pispip \in PISPIP$$

$$Unallocated Supply_{pispip} = \sum_{ns \in NS_{pispip}} Quantity_{ns} + \sum_{is \in IS_{pispip}} Quantity_{is}$$

$$+ Carried Forward Inventory_{(p,sp,t-1)} - (\sum_{d \in Dd_{pispip} \cup Id_{pispip}} Fulfilled Quantity_{d} + \sum_{d \in Sd_{pispip}} Fulfilled Quantity_{d,\omega}) \tag{4.8}$$

$$\forall pispip = (p, sp, t) \in PISPIP, \omega \in \Omega$$

$$0 \leq Fulfilled Quantity_{sd,\omega} \leq Quantity_{sd,\omega} + Over Produced_{sd,\omega} \qquad \forall sd \in Sd, \omega \in \Omega \tag{4.9}$$

$$Unfulfilled Percentage_{fgp,\omega} \geq 0 \qquad \forall fgp \in FGP, \omega \in \Omega \tag{4.10}$$

$$Unfulfilled Percentage_{fg,\omega} \geq 0 \qquad \forall fg \in FG, \omega \in \Omega \tag{4.11}$$

$$Unfulfilled Quantity_{stp,\omega} \geq 0 \qquad \forall stp \in STP, \omega \in \Omega \tag{4.12}$$

$$Over Fulfilled Quantity_{stp,\omega} \geq 0 \qquad \forall stp \in STP, \omega \in \Omega \tag{4.13}$$

$$Unfulfilled Quantity_{st,\omega} \geq 0 \qquad \forall st \in ST, \omega \in \Omega \tag{4.14}$$

$$Over Fulfilled Quantity_{st,\omega} \geq 0 \qquad \forall st \in ST, \omega \in \Omega \tag{4.15}$$

$$Real Fulfilled Quantity_{id,\omega} \leq Fulfilled Quantity_{id} \qquad \forall \omega \in \Omega, id \in ID_{pispip} | \sum_{sd \in SD_{pispip}} (Quantity_{sd,\omega} > 0) \tag{4.16}$$

$$Real Fulfilled Quantity_{id,\omega} \leq Quantity_{id,\omega} \qquad \forall \omega \in \Omega, id \in ID_{pispip} | \sum_{sd \in SD_{pispip}} (Quantity_{sd,\omega} > 0) \tag{4.17}$$

From these constraints it can be observed that the only reason that B_ω is scenario dependent is the presence of fulfillment goal constraints in which fulfilled quantity and overfulfilled quantity are divided by the quantity of the sales demand under that scenario. This is the direct consequence of the fact that unfulfillment is measured as percentage. When the model would be a little bit adapted such that a penalty on unfulfillment is given per quantity B_ω reduces to B. The coefficients d_ω are scenario dependent as in the goal function the fulfilled sales demand under a scenario is rewarded with the revenue per unit of the sales demand under that scenario. As explained in chapter 2 that revenue per unit is dependent on the quantity of the sales forecasts netted to the sales demand. When another revenue per unit calculation would be used it might very well be possible to reduce d_ω to d.

Returning to the division of variables we note that as the inventory demand can be defined as a number of days times the sales quantity per day, the quantity is sometimes scenario-dependent. For those inventory demands we still want to have a fixed decision variable as it can form the basis of high-level strategic decisions, but in the goal function only the real fulfilled quantity can be rewarded.

Finally note that these restrictions could be relaxed. Now per scenario a certain customer service level, and a productivity level is required. It is also possible to say that the expected customer service or the expected productivity over all scenarios should be on a certain level. The constraints would then become for the example of

a fulfillment goal:

$$ExpectedUnfulfilledPercentage_{fg} \geq FulfillmentPercent_{fg} - \sum_{\omega \in \Omega} p_\omega * \sum_{sd \in SD_{sals}} (FulfilledQuantity_{sd,\omega} - OverProduced_{sd,\omega})/Quantity_{sd,\omega}$$
$$\forall sals \in SalS, fg \in FG_{sals}$$

where p_ω is the probability of forecast scenario ω. This implies that such a relaxation is only possible for situations when probabilities are known. Relaxations for the product fulfillment goal, sales target and product sales target works likewise. The term $ExpectedUnfulfilledPercentage_{fg}$ is in the goal function penalized by the old WeightFulfillmentGoal, but is taken outside the summation over the scenarios. It is an relaxation in the sense that for every solution in total a lower or equal penalty is given, such that a higher or equal objective value is assigned to the solution. This holds especially for solutions in which the fulfillment goal under the old rules were hardly ever reached in order to score better on the other business goals. Therefore it is quite well possible that after this relaxation an optimal solution will be found with a better score on the other business goals than in the old solution. In our models we assume the planner wants to reach the fulfillment goal in every forecast scenario and we will not elaborate on this relaxation further.

4.4 Solution Methods

In this section for the three models a number of solution methods are proposed. These solution methods are largely based on methods described in the literature. But here some new possibilities are proposed for problems, where only right-hand uncertainty occurs.

4.4.1 Model I

The model which optimizes on the expected goal function can be solved by a LP solver, for example using the simplex method because this so called deterministic equivalent of the stochastic problem is a LP formulation. In fact this is the most normal solution and the solution implemented in the developed Robust Macro Planner. However, this can still lead to larger computation times when the original deterministic problem is very large or the number of scenarios is large. After all the size of the model is equal to the original size times the number of scenarios. In those cases it can be interesting from computational point of view to investigate alternatives. However current LP solvers of CPLEX are already very efficient and therefore it seems that try to implement an own LP solving method designed for such large problems as for example in [21] has a low chance to give an improvement.

But the structure of the problem, the scenario-dependent variables are used in separate constraints, gives causes to investigate decomposition methods. To fully exploit possibilities of decomposition methods we decided that unlike in the Macro Planner implementation, we will focus on models in which unfulfillment is measured by number of units unfulfilled and revenue per unit is obtained from taking the average of the revenues per quantity of orders and sales forecasts instead of taking the weighted average. The difference of the approximation of reality, would it be worse, is for many cases minimal and by allowing d_ω and B_ω to be reduced to d and B the solution time could be lower for decomposition methods. However we do acknowledge that it is slightly more natural to measure fulfillment as a percentage and therefore this is a small weak point in the decomposition methods analysis.

Furthermore to enforce relative complete recourse we introduced a variable $\text{ToBeSpend}_{\text{pispip}}$, which is equal to the amount flowing in to a pispip. In this way the scenario independent inflow can be separated from the scenario dependent outflow considered in the first-stage constraints and the second-stage constraints. From the deterministic point of view this change does not change anything as the variables are substituted in the solution process. Needless to say, when comparing the solution times the decomposition methods obviously are only compared against deterministic methods on the same problem, with the same assumptions for d and B.

Due to the special structure of the problem a stochastic Benders decomposition (also called L-shaped decomposition), equivalent with Dantzig-Wolfe decomposition on the dual version of the problem, may be possible. This method is originally introduced in [37] and since then an important tool in stochastic programming. The idea of these decomposition methods is to split the problem into many subproblems which can be solved a factor more than many quicker than the overall problem by an LP-solver, thus reducing computation time when there are many scenarios. As can be seen in [2] other optimizing software solutions recognize the possible benefits of decomposition methods in robust optimization. In AIMMS you have the possibility to convert a deterministic problem into a stochastic problem and as solution method Benders decomposition is one of the two options. Also in literature Benders decomposition for recourse models is much discussed. See for example section 3.2 of [18]. In [14] nested Benders decomposition designed for multi-stage problems is discussed.

In the context of multi-stage problems there are apart from the nested Benders decomposition approach which decomposes the problem to all the stages, leading to problems per stage which are made of sub problems per possible parameter realization, more decomposition methods of which the scenario decomposition method is the most prominent one. The scenario decomposition method identifies a scenario as one vector of parameter realizations from stage 1 until stage T. The problem is now decomposed into problems per scenario in which the nonanticipativity constraints are dropped. That are the constraints to ensure that variables should be the same up to stage i for scenarios that have the same parameter realizations up to stage i. In [22] this approach is studied extensively. Possibly the attractiveness of this alternative is mostly the possibility of avoiding a nested structure. For a two-stage recourse model this way of decomposition is actually never proposed, possibly because the lack of nested structure in the stochastic Benders decomposition. Therefore we will only focus on the stochastic Benders decomposition as alternative to solving the large linear program. Other decomposition methods which are interesting for a multi-stage case but are not interesting for our two-stage model can be found in [20].

For this model I to apply stochastic Benders decomposition the form should now first be rewritten to:

$$\begin{aligned} &\max\ cx + \theta \\ &\quad \text{st}\ A_1 x \leq b \\ &\qquad \theta \leq Q(x) \\ &\qquad x \geq 0 \end{aligned}$$

with $Q(x) = \sum_{\omega \in \Omega}(p_\omega * Q(x, \omega))$, a probability-weighted aggregation of subproblems

$$Q(x, \omega) = \max\{dy_\omega | A_2 x + B y_\omega \leq b_\omega, y_\omega \geq 0\}$$

Note that the subproblem $Q(x, \omega)$ can also be formulated in it's dual form:

$$Q(x, \omega) = \min\{\pi_\omega (b_\omega - A_2 x) | \pi_\omega B \leq d, \pi_\omega \geq 0\}$$

by the strong duality in linear programming. Remark that in this dual formulation the feasible region is the same for every ω and every possible first-stage value x. The

first-stage and second-stage decisions are clearly separated now and the decomposition possibilities appear.

A single-cut stochastic Benders decomposition, the most common form, now works as follows. First constraint $\theta \leq Q(x)$ can be dropped to get the master problem (MP) of the Benders decomposition. The (MP) is solved multiple times, each time leading to a candidate solution $(\bar{x}, \bar{\theta})$. In the first iteration $\bar{\theta}$ is obviously chosen as some value but the idea is to generate optimality cuts in each iteration thus restricting θ depending on the choice of x in further iterations. The objective function value which can be reached by maximization non-increases in the number of added restrictions for θ and therefore it is natural to set θ to infinity when generating the first candidate solution. Observe that stochastic Benders decomposition is therefore a form of a cutting plane method.

For each solution $(\bar{x}, \bar{\theta})$ the optimality cuts are generated by solving $Q(\bar{x})$. The cut that can be introduced for candidate solution $\bar{x}$ is then

$$\theta \leq \sum_{\omega \in \Omega} (p_\omega * \pi(\bar{x})_\omega (b_\omega - A_2 x))$$

Here $\pi(\bar{x})_\omega$ represent the optimal dual variables found for forecast scenario ω and candidate solution $\bar{x}$. The correctness of this new optimality cut can be understood by observing that for $x \neq \bar{x}$, all other possible solutions for the first-stage variables, there exists an optimal $\pi(x)$ in every scenario for which the value of the dual problem is equal or lower than for $\pi(\bar{x})$ as this is still a feasible solution of the subproblem for that x but not necessarily an optimal one. Risking repeating it too much, we would like to stress that this problem has relatively complete recourse and therefore no feasibility cuts are needed.

After a finite number of iterations this method will converge to an optimal solution for (x, θ). For this candidate solution the generated optimality cut is already satisfied as equality. By remarking that the candidate solution always gives an upper bound on the optimal objective value and that an $\bar{x}$ with θ such that $\theta = \sum_{\omega \in \Omega} (p_\omega * \pi(\bar{x})_\omega (b_\omega - A_2 x))$ is a lower bound on the optimal objective value, it can also be stated as the situation in which this lower bound and upper bound are equal. As there is a finite number of feasible bases for each subproblem, only a finite number of cuts can be generated and the procedure will converge in a finite number of steps. Then the variables y_ω can be found again by solving the subproblems for this given x or by looking it up from memory as the subproblems have already been solved for this x to establish convergence.

Summarizing in steps the stochastic Benders decomposition works as follows:

Step 1 Solve (MP) generating a new candidate solution $(\bar{x}, \bar{\theta})$. In the first iteration $\bar{\theta}$ is set as infinity.

Step 2 Solve $Q(\bar{x})$ to generate the optimality cut $\theta \leq \sum_{\omega \in \Omega} (p_\omega * \pi(\bar{x})_\omega (b_\omega - A_2 x))$. If this constraint is already satisfied, stop; the candidate solution is the optimal solution for (x, θ) and the solution for y_ω can be derived by solving the subproblems. Else add that constraint to (MP) and go back to step 1.

As mentioned this procedure of the stochastic Benders decomposition using an aggregated θ is called a single cut approach. In [7] another way to execute the same Benders decomposition was suggested. It is equally well also possible to write the problem as

$$\begin{aligned}
&\max cx + \sum_{\omega \in \Omega} p_\omega * \theta_\omega \\
&\quad \text{st } A_1 x \leq b \\
&\qquad \theta_\omega \leq Q(x, \omega) && \forall \omega \in \Omega \\
&\qquad x \geq 0
\end{aligned}$$

with $Q(x, \omega)$ still being defined as

$$Q(x, \omega) = \max \{dy_\omega | By_\omega \leq b_\omega - A_2 x, y_\omega \geq 0\}$$

A similar decomposition procedure can be used. Now all the constraints $\theta_\omega \leq Q(x, \omega)$ are dropped to get the Master problem. In each iteration a candidate solution $(\bar{x}, \bar{\theta}_1, ..., \bar{\theta}_{|\Omega|})$ is generated and after solving the subproblems a total number of $|\Omega|$ constraints can be added:

$$\theta_\omega \leq \pi(\bar{x})_\omega (b_\omega - A_2 x)$$

for every scenario $\omega \in \Omega$. Contrary to the previous method this is called the multicut procedure.

There are pros and cons for both the single cut procedure and the multicut procedure. It cannot be said that in general one method performs better. In the single cut procedure aggregation leads to a loss of information and in general (a counter example was also given in [7]) less iterations with candidate solutions have to be tried to obtain the optimal solution. On the other hand in the multicut procedure the size of the (MP) will be larger and increasingly larger in the number of iterations. If m_1 represents the initial number of constraints in the (MP) and n_1 is the size of decision vector x, the size of (MP) expressed in number of constraints times number of variables at iteration k+1 in a multicut approach will be $(m_1 + k * |\Omega|) * (n_1 + |\Omega|)$, while for a single cut this would be $(m_1 + k) * (n_1 + 1)$. Therefore the (MP) in the multicut approach has in general a larger solution time. These effects are conflicting. A suggested rule of thumb is that the multicut approach is preferable to a single cut approach if the number of scenarios is not much larger than the number of decisions variables of the original (MP). The intuition should be that when the number of scenarios is much larger the increase in size of the (MP) after adding the cuts is too large.

Traditionally the choice was to use a multicut procedure or a single cut procedure, but in [36] an approach is suggested in which falls in between. In their algorithm the level of aggregation is determined endogenously in each iteration. Their results suggest this approach may lead to better results and therefore this is interesting for our problem as well.

Their procedure called adaptive multicut is initialized as follows. Introduce for each iteration k a partitioning set $\Omega(k) = \{\Omega_1, \Omega_2, ..., \Omega_{l_k}\}$ with the elements of Ω_i for $1 \leq i \leq l_k$ being scenarios. This partitioning satisfies by definition $\Omega = \Omega_1 \cup ... \cup \Omega_{l_k}$ and $\Omega_i \cap \Omega_j = \emptyset$ if $i \neq j$. So every scenario is included in one set in each iteration. Define the aggregate probability as $p_{\Omega_i} = \sum_{\omega \in \Omega_i} (p_\omega)$. A (MP) with added optimality cuts in iteration k now looks like:

$$\begin{aligned}
&\max cx + \sum_{d=1,...,l_k} \theta_d \\
&\quad \text{st } A_1 x \leq b \\
&\quad\quad \theta_d \leq \sum_{\omega \in \Omega_d} (p_\omega * Q(x, \omega)) \qquad\qquad \forall d = 1, ...l_k \\
&\quad\quad x \geq 0
\end{aligned}$$

The outline of the adaptive multicut procedure in steps can now be given as:

Step 0 Set k=0 and initialize a certain set aggregation $\Omega(0)$.

Step 1 Solve (MP). Set k=k+1 and store the candidate solution of the Master Problem as $(x^k, \theta_1^k, ..., \theta_{l_k}^k)$

Step 2 Generate $\Omega(k)$ based on $\Omega(k-1)$ according to some aggregation policy (for example: scenarios without tight/active constraints are aggregated). Then

sum the probabilities of the old aggregations to get the probability of the new aggregation. Also delete the variables θ_d which belonged to aggregations which are now aggregated and introduce a new variables θ_d for the new aggregation. Construct optimality cuts for the new aggregations by taking a weighted sum over the old optimality cuts.

Step 3 Solve the subproblems $Q(x, \omega)$ to get new optimality cuts in the master problem. Stop if x satisfies the optimality cuts, x is optimal. Else add them to the (MP) and go to step 1.

When doing this for our problem we should probably start with $l_0 = |\Omega|$ to have a pure multicut procedure and let the algorithm aggregate scenario's afterwards. This is motivated by the point that scenario's cannot be disaggregated in this algorithm such that the aggregation level can never become lower and multicut would otherwise not have been possible. Note though that the authors recommend further research on possibilities of disaggregation. For our model we will use the redundancy threshold(δ) of the authors. It means that if more than a fraction of δ of the optimality cuts for an aggregation is a redundant cut, a cut which is already satisfied by the choice of x and θ_ω, the aggregation is nominated to be aggregated.

Finally something specific can be noted for our model and in general stochastic models with only right-hand uncertainty. Technology matrix B and cost vector d are the same for all scenarios in our assumptions and therefore the feasible region is the same for all duals of the subproblems $Q(x, \omega)$. This should lead to computational advantages. For example it is said that in a paper of Wets from 1983 the concept of bunching is introduced. The point is that solving one problem $Q(x, \omega)$ by the dual simplex method leads to an optimal basis, say $\mathcal{B}$. This optimal basis is also dual feasible for every other right hand side (in the primal formulation) $b_\omega - A_2 x$. It is possible to check whether the basis is also primal feasible after calculating $\mathcal{B}^{-1}(b_\omega - A_2 x)$. This vector consists of the values we want to assign to dual variables π and obviously they are obliged to be larger or equal than 0.

In this way it is possible to already find a solution for some subproblems $Q(x, \omega)$ without doing any operations on the basis and then these subproblems can be removed from the set of subproblems which need to be solved. The next subproblem from this set can be solved by using the old optimal basis $\mathcal{B}$ and applying the dual simplex method. When the optimal solution is found again the other subproblems are checked to have the same optimal basis. This procedure continues until every subproblem is solved or bunched as it is called.

A related idea to use the bunching method is trickling down. There is a set of right-hand side values $b_\omega - A_2 x$, lets call it $\mathcal{R}$. Then we start with a first element $b_\omega - A_2 x \in \mathcal{R}$. We solve it's subproblem to optimality. We store it's optimal basis in the root of a tree. Then we pick the next element of $\mathcal{R}$ and first we check if $\mathcal{B}^{-1}(b_\omega - A_2 x)$ is primal feasible. If this is primal feasible register this subproblem to the root note. If it is not primal feasible, perform one step of the simplex method. Store the number of the row which is removed from the basis and the row which is inserted in the basis in a branch of the root node. If this basis is not primal feasible continue to make a step of the dual simplex method and create a branch for the created node. Continue until primal feasibility is reached.

Then we can continue by picking the next element of $\mathcal{R}$. If the basis of the root node is already primal feasible we add the subproblem to the root node. Otherwise if that basis has a row which is not primal feasible for the subproblem that corresponds to a row for which we made a step of the dual simplex method earlier and thus created a branch, we move along that branch and it is already known which row is inserted. And again we check for primal feasibility. If there were no branches already in the tree for all primal infeasible rows, we have to perform a step of the dual simplex

method and create a branch and node ourselves. We continue until primal feasibility is reached. We continue until we have gone through all subproblems. The advantage to the normal bunching is that once we found an optimal basis for one subproblem and another optimal basis for another subproblem, obtained from n dual simplex method steps, we can obtain the optimal basis for a subproblem which needs only a part of those n steps at no price.

In [17] and [15] optimal settings for such a trickling down method are discussed. Important questions are what needs to be stored in the nodes (the entering row, an eta vector or something else) and what pivoting rules should be applied by choosing on which row a step of the dual simplex method has to be made. Those choices turn out to have an important effect on the calculation time. In this thesis we will not deal with the implementation a trickling down method and we will only use the normal bunching principle. For the sake of completeness we note that another technique called sifting is sometimes mentioned for solving multiple LP's with only right-hand side differences and it is said to be introduced in article of Gartska and Rutenberg from 1973.

Furthermore we propose some additional options for this problem when there is only right-hand side uncertainty. An observation is when $Q(\bar{x}, 1)$ is solved this leads to a cut $\theta_1 \leq \pi(\bar{x})_1(b_1 - A_2x)$ in the multicut case. However the obtained dual variables can also lead to valid cuts for all other scenarios: $\theta_\omega \leq \pi(\bar{x})_1(b_\omega - A_2x)$. These cuts could also be added to the master problem.

It might be that they are the optimal cuts for such a scenario for this $\bar{x}$ (A). But it is also possible that for such a scenario and such a $\bar{x}$ it is just a valid suboptimal cut. Then there are two options, (B1) it is for that scenario an optimal cut for another value of x, (B2) there are no values of x for which it is an optimal cut for that scenario. Anyhow it means that if 1 subproblem is solved, in principle $|\Omega|$ cuts can be obtained of different quality. These cuts of different quality all give some information to the (MP), but it has the downside of a (large) increased number of constraints. The number of variables on the other hand remains the same. Now the effects of this trade-off on the computation time cannot be stated theoretically, in some problems the fraction of cuts of type (A) and (B1) may be large and in other problems the fraction could be zero such that all extra cuts are rather worthless. The advantage over just generating some random extreme points of the polyhedron describing the feasible region of the dual variables is larger when there is more similarity between forecast scenarios. Keep in mind that as it is not possible to give a general rule defining which cuts never will be active, there is no possibility of identifying cuts of type (B2), there is no procedure to delete cuts. We can try some methods based on this observation.

If we would apply this to the single cut decomposition the story is different. When we have for each $\pi(\bar{x})_\omega$ in $\theta \leq \sum_{\omega\in\Omega}(p_\omega * \pi(\bar{x})_\omega(b_\omega - A_2x))$ a number of $|\Omega|$ possible values we could in principle create $|\Omega|^{|\Omega|}$ cuts. However, this would lead to a very undesirable situation: with a much larger number of cuts than in the similar multicut case we represent the same information. In this situation choosing the multicut approach is thus a free lunch. Furthermore we did find another sensible set of rules using the $|\Omega|$ possibilities for the $|\Omega|$ scenarios to create more than $|\Omega|$ cuts.

For the same price of solving $|\Omega|$ subproblems more information for other realizations of x is gathered. The opposite idea is behind the approach of solving 1 subproblem without solving the other $|\Omega| - 1$ problems. Then an amount of computation time is saved and still a reasonable amount of information is gathered. With less information in general, there exist counterexamples, more iterations to solve the (MP) are required. Also here a difference between the multicut procedure and the single-cut procedure can be pointed out: in the multicut procedure it is certain that one of the added cuts is an optimal one, while for the single-cut procedure it is unlikely that the added cut is an optimal one.

We now offer three choices in this model with only right-hand side uncertainty for the generation of cuts in the multicut procedure. In all these choices it seems desirable to apply the bunching method and to store previous optimal bases for the subproblems to facilitate warm starts.

1 Traditional method This is the traditional method which solves all subproblems to optimality in each iterations and does not use the $\pi(\bar{x})_\omega$ values for generating cuts on other scenarios. The only way it can make use of the special structure of out problem is by the bunching method.

2 $|\Omega| * |\Omega|$ cuts per iteration This is the method in which all subproblems are solved to optimality in each iteration and every value $\pi(\bar{x})_\omega$ is used in $|\Omega|$ cuts such that in total $|\Omega| * |\Omega|$ cuts are generated. The bunching method can also be applied such that some subproblems do not have to be solved by a simplex method, but can be given their optimal $\pi(\bar{x})$ immediately.

3 Solve one subproblem to optimality This method is formally described by the following steps:

Step 0 Initialize $a = 1, start = 1$ and initialize (MP). Go to step 1.

Step 1 Solve (MP) to get a candidate solution with $\bar{x}$ being the candidate solution for the first stage variables in iteration k. Initialize $i = 0$. Go to step 2.

Step 2 Solve $Q(\bar{x}, a)$ by applying the dual simplex method and find $\pi(\bar{x})_a$.
IF $\bar{\theta}_a \leq \pi(\bar{x})_a(b_a - A_2x)$ already holds
set $a = a + 1, i = i + 1$ and go to Step 2
IF a=start
STOP
ELSE Construct $|\Omega| - i$ cuts (not for scenarios earlier solved to optimality for this candidate solution) with this solution $\pi(\bar{x})_a$.
Set start=a
Go to Step 1.

Now it seems not to be literally true that in each iteration only one subproblem is solved as when the found optimality cut of a subproblem is already satisfied another subproblem is solved. However without loss of generality it can be said that this is the same as two iterations in which one subproblem is solved as solving the (MP) after a set of already included cuts is introduced is just recalling the last candidate solution and costs no time. Informally the convergence of this algorithm in a finite number of steps can be derived from the point that there are only a finite number of extreme points in the feasible region of each subproblem for which in each step a cut is found. Because in each iteration only 1 subproblem is solved, the bunching method cannot be exploited here.

Parallel to the issue related to the choice between a single cut option and the multicut option when the adaptive multicut algorithm was interesting it might be that in this case a golden mean between option 2 and 3 can be found, a part of the subproblems is solved to optimality. Possibilities one might think of are for example:

- solving more subproblems to optimality when the objective value of the (MP) is not decreased enough during the last iteration
- solve in each aggregation in the adaptive multicut algorithm only one subproblem to optimality
- solve in each aggregation in the adaptive multicut algorithm or all subproblems or no subproblems to optimality

Another option could be only to add cuts for a part of the scenarios in case solving the (MP) is slowed down considerably by the number of constraints as most constraints we add now are redundant. However, it is not really possible in advance to say which constraints are redundant and even in the normal multicut case there are no proposals to add only cuts for certain scenarios in a major iteration, while it is equally possible that it speeds up the algorithm. Therefore we will not consider such an option.

To complete the story we have to mention the difference for the three choices if single-cut or adaptive multicut is applied. As said, when aggregation occurs choice 2 is not really possible anymore. Choice 1, the traditional option, is of course applicable in the single-cut or adaptive multicut procedure. It is clear that before the algorithm ends at least once a total optimal cut is introduced.

Another critical view on stochastic Benders decomposition comes from Ruszczynski. His point is that in a cutting plane method like stochastic Benders decomposition the first optimality cuts may be inefficient and later on in the algorithm there is no reliable possibility of deleting earlier cuts. Therefore he studies in [29], [30], [32] and [31] his so called regularized Benders decomposition. His idea is to change the objective of (MP) into a non-linear function

$$\max_x \{-\frac{1}{2 * \rho^k} * \|x - z\|^2 + cx + \sum_{\omega \in \Omega} (\theta_\omega)\}$$

We will now call this changed master problem the regularized master problem, (rMP). The advantage of this new object function is that the method is stabilized, i.e. a candidate solution in an iteration will not be very different from the previous candidate solution. Here the variable z, which is a feasible solution, is called the reference point. In each iteration the reference point can be replaced by the candidate solution, an exact serious step or an approximate serious step, or remain the same, a null step. The candidate solution is found by applying bundle methods and in the end the method still converges.

This means that it is now possible to discard the generated cuts which were only meant to describe a θ_ω accurately for a very different first-stage solution from the model: after all this first-stage solution will not be a candidate in the next iterations by the quadratic penalty. This will make the (rMP) smaller than a normal (MP) such that it can offset the increased solution time due to the non-linearity. We will not implement this method because it takes more effort to implement the solution method. Therefore we also refer for a more detailed description to the papers of Ruszczynski. At least it may be an interesting option.

To conclude we have many ways to solve our model I: solving the deterministic equivalent or solving the decomposition method when it is possible to choose between single-cut multicut or adaptive multicut and for each of these choises there are still some options. In the next chapter we will again highlight what was exactly implemented.

4.4.2 Model II

As was shown the model which optimizes the worst-case goal function can also be written as a deterministic LP formulation. Therefore it can also be solved by a LP solver and again this is the way the implementation is done in the Quintiq software. Like for the previous model we will propose decomposition methods besides.

We use the same assumptions regarding d, B and the relative complete recourse characteristic. As already emphasized in section 4.3 after solving the big LP model, small LP models have to be solved to obtain the optimal y_ω. We do not know which scenario(s) are identified as worst by the algorithm as also scenarios other than the

worst can have their y_ω solution be set such that the equality $\delta = dy_\omega$ holds. And therefore we have to solve $|\Omega|$ of these problems.

Here it is possible to make extra gains with a decomposition method as then these problems are already solved. The possibilities are similar to those in model I, while there are some differences for this less standard stochastic Benders decomposition example. In the literature the combination of this kind of worst case problems and stochastic Benders decomposition is seldom seen. In [4] another type of decomposition is discussed for a worst case problem. But the ideas are obviously a bit similar to the ideas in the previous model. First note that the model can be rewritten to:

$$\begin{aligned} &\max cx + \theta \\ &\quad \text{st } \theta \leq Q(x,\omega) && \forall \omega \in \Omega \\ &\qquad A_1 x \leq b \\ &\qquad \theta, x \geq 0 \end{aligned}$$

with the subproblems being exactly as in model I:

$$Q(x,\omega) = \max \{dy_\omega | A_2 x + B y_\omega \leq b_\omega, y_\omega \geq 0\}$$

And note again that the subproblem $Q(x,\omega)$ can also be formulated in it's dual form:

$$Q(x,\omega) = \min \{\pi_\omega (b_\omega - A_2 x) | \pi_\omega B \leq d, \pi_\omega \geq 0\}$$

As for model I the feasible region is the same for every ω and every possible first-stage value x. The only difference with the multicut version for model I is thus that there is only one variable ω.

Here the constraints $\theta \leq Q(x,\omega)$ for every scenario dropped to get the (MP). This is solved multiple times to get candidate solutions $(\bar{x}, \bar{\theta})$. Each iteration $|\Omega|$ cuts

$$\theta \leq (\pi(\bar{x})_\omega (b_\omega - A_2 x))$$

can be introduced. As in the former model, a candidate solution is an optimal solution if all generated optimality cuts for that candidate solution are already satisfied. At least one of the satisfied generated cuts will be satisfied as equality. That occurs at, given by the optimality cuts, the worst-case scenario for first-stage solution $\bar{x}$. Then it is easy, at least if the optimal bases and solutions of the latest subproblems are still in memory, for the optimal candidate solution (x, θ) to get the optimal solutions for variables y_ω as they are the dual variables of the solved dual versions of $Q(x,\omega)$.

Unlike for model I, there is now no possibility to use a single cut method or a adaptive multicut method as nowhere occurs aggregation. However similarly to model I the feasible regions of each subproblem are the same. Therefore we do have also the three possibilities to use the fact that one generated solution for the dual variables is feasible for all first-stage solutions and all scenarios. The traditional method, the $|\Omega| * |\Omega|$ cuts per iteration method and the solve one subproblem to optimality method for this model work in the same way as explained at model I. Now the solve one subproblem to optimality method can turn out very strong if there is one scenario which is regardless of the choice for first stage variables worst and that scenario is chosen to be optimized. After the scenario is optimized to optimality the other scenarios have to be solved only once to derive the final optimality cut for those scenarios. The candidate solution will already satisfy this cut and the procedure is terminated.

Due to the maximin structure of the problem some further specific considerations come into mind. When we have a candidate solution $(\bar{x}, \bar{\theta})$ after a number iterations (say k), we may think that after the coming optimality cuts the candidates for x may remain in the neighborhood of $\bar{x}$ and furthermore (as a consequence) we may think

that the worst-case scenario(s) found at the candidate solution $\bar{x}$ will also remain worst-case scenario(s) in the neighborhood of $\bar{x}$. As other scenarios are not expected to become a worst-case scenario we expect cuts for those scenarios to be irrelevant. Therefore we can think of a depth first search method. Only when a generated optimality cut for our worst-case scenario on which we focus is already satisfied we have to generate the other optimality cuts on order to prove optimality of the solution. It can be described in the following steps:

Step 1 until k Apply as usual the stochastic Benders method with the traditional method or the $|\Omega| * |\Omega|$ cuts per iteration method. After k steps, go to depth step 1.

Depth step 1 Generate a new candidate solution $(\bar{x}, \bar{\theta})$. Identify for this candidate solution the constraint(s) and corresponding scenario(s) for which $\theta = \pi(\bar{x})_\omega(b_\omega - A_2\bar{x})$ holds. Store these scenario(s) in the set S.

Depth step i+1 Generate optimality cuts by solving $Q(\bar{x}, \omega)$ for all $\omega \in S$. IF the optimal optimality cuts already hold as equality, solve all other subproblems $Q(\bar{x}, \omega)$ to test whether these are also satisfied.
IF all these constraints are satisfied, $(\bar{x}, \bar{\theta})$ is the optimal solution.
ELSE go back to Depth step 1 to initialize a new set S.
ELSE generate a new candidate solution. $(\bar{x}, \bar{\theta})$ and generate cuts from the subproblems in S. Identify for this candidate solution the $\omega \in S$ for which there is no constraint which holds as equality anymore and remove these from S.
IF S is empty, go to Depth Step 1.
ELSE go to Depth Step i+1.

This can be added to the three options as the depth search after k steps method. The best value to be chosen for k will obviously be very problem instance dependent. To explain the difference with the solve one subproblem to optimality method it is important to note that in the worst case goal function situation in the end the first-stage candidate solution is indicated by the most tight cuts of the scenario(s) identified as worst and an optimal cut for the other scenarios only serve as confirmation of the solution. In the expected goal function the cuts of all scenarios are determining the first-stage solution. In the solve one subproblem to optimality method just a scenario is picked to be solved to optimality, while here the signals about the scenario(s) which are expected to become worst are used to determine which scenarios are solved. It is useful to start with k normal steps in order to get a reasonable signal about the scenario(s) to be identified as worst.

To conclude there are a number of methods to be used here: solving the deterministic equivalent or solving with Benders decomposition, when there are still possibilities to make a choice between the traditional option, the generate $|\Omega| * |\Omega|$ method and the solve one subproblem to optimality method. Besides we can switch to a depth first search method after k normal steps. In the next chapter we will again highlight what was exactly implemented.

4.4.3 Model III

Our third model may be harder to solve for large instances because it is an integer programming model. In principle it could always be enumerated into $\frac{|\Omega|!}{N! * (|\Omega| - N)!}$ models of the form of model II but it could be possible to find a solution with less effort.

We tried several solution methods.

But before we study the way of solving our model, first we have to address the way to determine our important parameter big M. A natural suggestion would be to solve deterministic LP models for each of the scenarios separately to get an upper bound for the goal function value and to use this value as bigM. These are the old deterministic LP models with as only difference an extra possibility of overproduction and the possibility of fulfilling inventory demand more than the quantity, when the real fulfilled demand is required to be lower or equal than the quantity.

Remark now that if for exactly one of the scenarios such a LP model would be unbounded, this does not immediately imply that the problem is ill-posed. After all, the goal function for that scenario has no influence on the objective function if x is chosen such that the subproblem has an unbounded solution: it can not be the Nth worst scenario. However, for this special case we can denote that if for one scenario the LP model is unbounded, for all scenarios it has to be unbounded. This can be explained by the fact that all problems have the same constraint matrix except those constraints for which the fulfilled sales demand plus the scrap should be lower or equal than the quantity for a sales demand. Both fulfilled sales demand and scrap are individually not bounded from above and if producing 1 product more on the supply routings for which a maximum capacity is already reached, thus fulfilling 1 more and scrapping 1 more is attractive, it will be so for any quantity of the sales demand. Therefore either all scenarios are unbounded and the problem is ill-posed or no scenario is unbounded. Futher note that these new LP models also only have right-hand differences such that it would be efficient to apply the dual simplex method to compute bigM.

Note that once we solve all deterministic LP models the computational effort for the N-reliability model is exactly the same as for a N-reliability model in absolute regret. That model looks like

$$
\begin{aligned}
&\min \delta \\
&\quad \text{st } \delta \geq O(\omega) - cx + dy_\omega - (1 - z_\omega)M && \forall \omega \in \Omega \\
&\qquad Ax + By_\omega \leq b_\omega && \forall \omega \in \Omega \\
&\qquad \sum_{\omega \in \Omega} (z_\omega) = N \\
&\qquad x, y_\omega \geq 0 \\
&\qquad z_\omega \in \{0, 1\}
\end{aligned}
$$

with $O(\omega)$ being the optimal value of the deterministic subproblem for scenario ω. The difference between the worst-case robustness measure and the minmax absolute regret measure is also this constant $O(\omega)$, and there solving of deterministic models would be extra effort compared to the worst-case robustness measure. In N-reliability it is done nevertheless for both. This means that if, in a project where this suggested way to determine big M is used, for some reason a Macro Planner user prefers to use an absolute regret objective this can easily be adapted.

However the described approach fails if it is also possible to have negative goal function values, nearly always. When for a certain candidate solution x the positive optimal value for a scenario which is included in the reliability set is more than big M higher than the negative optimal value for a scenario which is not included, the N-reliability method is messed up.

As we want to apply the Macro Planner to a general case in which negative goal functions occur and in which a situation in which the goal function value can not be bounded from below could easily occur, we want to make another proposal which is however not really tailored to the specific problem instance. In the literature in [8] possibilities to determine the upper bound on regrets are given, which can be

considered to find a big M in this situation as well. Their alternative 'MinimaxII' is the possibility which requires least direct computation time. In the rest of the thesis we choose this to be $1 * 10^{10}$. In theory it might prove to be low, or it might prove unnecessary high such that it slows done branch and bound methods, but that holds for any choice.

Returning to the possible solution methods for our N-reliability models, again the implementation in the Robust Macro Planner is just feeding the deterministic equivalent MIP to CPLEX although as previously explained the model is there slightly different. It will most probably use a kind of branch and bound procedure on the binary variables indicating whether a scenario is selected. As announced in 4.3 after solving the big MIP model, we still have no optimal values for y_ω. Therefore we have to solve the $|\Omega|$ small subproblems to obtain their solutions. Here several decomposition methods are listed. As N-Reliability is only mentioned in a few articles, these decomposition methods are not discussed in literature, but again we use the similarity with the previous models.

A requirement for decomposition methods is that there is a useful structure in the problem and $\sum_{\omega\in\Omega}(z_\omega) = N$ spoils this somewhat, as there are now not only variables (x) which come back in multiple constraints ($\forall\omega \in \Omega$), but also a constraint which uses variables of multiple scenarios. Some decomposition methods which would normally be possible in robust optimization will not work out in this case due to the constraint. An example is the scenario relaxation algorithm as in [4]. Although it is designed for binary and continuous variables in the first-stage of a minimax problem, it is not possible to use it in this situation. The subproblems of the Lagrange dual in a scenario decomposition method are not independent anymore and therefore their solution method cannot be applied.

But the previous seen stochastic Benders decomposition is still to be considered. It is known that the presence of integer (binary) variables can be a reason to apply Benders decomposition and that the presence of scenarios can be a reason to apply stochastic Benders decomposition, but it is unclear what to do when a mix of these phenomena occur. We present two possibilities to partition the problem into subproblems.

Normal stochastic Benders decomposition

This is totally similar to the methods for the previous models. Start with

$$\begin{aligned}
&\max\ cx + \theta \\
&\quad \text{st } \theta \le Q(x,\omega) + (1 - z_\omega)M && \forall\omega \in \Omega \\
&\qquad A_1 x \le b \\
&\qquad \sum_{\omega\in\Omega}(z_\omega) = N \\
&\qquad x \ge 0, z_\omega \in \{0,1\}
\end{aligned}$$

with

$$Q(x,\omega) = \max\{dy_\omega | A_2 x + By_\omega \le b_\omega, y_\omega \ge 0\}$$

As for all models in this chapter the feasible region of the dual version of $Q(x,\omega)$ is the same for every ω and every possible first-stage value x.

A number of $|\Omega|$ constraints, all $\theta \le Q(x,\omega) + (1 - z_\omega)M$ are dropped to get the (MP). This (MP) is solved multiple times to get candidate solutions $(\bar{\theta}, \bar{x}, \bar{z}_\omega)$ and each iteration $|\Omega|$ cuts

$$\theta \le (\pi(\bar{x})_\omega(b_\omega - A_2 x)) + (1 - z_\omega)M$$

can be introduced. A candidate solution is than an optimal solution if all generated cuts for that solution are already satisfied. At least one of these satisfied cuts will be satisfied by equality, the scenario which is identified as worst but $|\Omega| - N$ for first-stage solution $(\bar{x}, \bar{z}_\omega)$.

Like in model II there is no aggregation and therefore there are no single-cut or adaptive multicut possibilities. As there is again only right-hand uncertainty and as therefore the feasible regions of the subproblems are the same, we can vary in our method by using the traditional method, the $|\Omega| * |\Omega|$ cuts per iteration method and the solve one subproblem to optimality method for this problem. Also the depth search after k steps method can be tried, when in the outline sketched in the previous section $(\bar{x}, \bar{\theta})$ should be replaced by $(\bar{x}, \bar{\theta}, \bar{z}_\omega)$ and the constraint $\theta = (\pi(\bar{x})_\omega(b_\omega - A_2x))$ should of course be replaced by $\theta = (\pi(\bar{x})_\omega(b_\omega - A_2x)) + (1 - z_\omega)M$. The by the model identified worst-case scenario(s) is now just the by the model identified worst but $|\Omega| - N$ scenario(s) for which a constraint is satisfied by equality.

The difference with the previous method is that the master problem is harder to solve because it contains binary variables. This makes the computation time without doubt larger than the time for solving model II. Other effects may be that the always important optimal trade-off between solution time of the (MP) and the number of times (MP) has to be solved, now turns out to be differently and therefore another cut method is more efficient.

Nested Benders decomposition

Another option is to use nested Benders decomposition. This decomposition is nested in the sense that our problem is decomposed in the problem of choosing the binary variables, choosing the other first-stage variables and choosing the second-stage variables. Write the problem as

$$
\begin{aligned}
&\max \theta \\
&\quad \text{st } \theta \leq Q(z) \\
&\qquad \sum_{\omega \in \Omega} (z_\omega) = N \\
&\qquad z_\omega \in \{0, 1\} \qquad\qquad \forall \omega \in \Omega
\end{aligned}
$$

with Q(z), the problem consisting of choosing the first-stage variables given the choice for the scenarios in the reliability set, being:

$$
\begin{aligned}
&\max cx + \delta \\
&\quad \text{st } \delta \leq Q(x, \omega) + (1 - z_\omega)M \qquad\qquad \forall \omega \in \Omega \\
&\qquad A_1x \leq b \\
&\qquad x \geq 0
\end{aligned}
$$

and here, as usual, a subproblem per scenario is:

$$Q(x, \omega) = \max \{dy_\omega | A_2x + By_\omega \leq b_\omega, y_\omega \geq 0\}$$

which is in dual formulation

$$Q(x, \omega) = \min \{\pi_\omega(b_\omega - A_2x) | \pi_\omega B \leq d, \pi_\omega \geq 0\}$$

For every x and every ω (and z, which does not come back in this subproblem) this subproblem has the same feasible region. When problem Q(z) is solved by stochastic Benders decomposition the constraints $\delta \leq Q(x, \omega) + (1 - z_\omega)M$ or dropped and cuts

are added. If we write the dual variable generated for scenario ω in iteration k as $\pi_{\omega,k}$ we have thus after K iterations of the form:

$$\begin{aligned}
&\max cx + \delta \\
&\quad \text{st } \delta + \pi_{\omega,k} A_2 x \leq \pi_{\omega,k} b_\omega + (1 - z_\omega)M \qquad \forall \omega \in \Omega, k \in 1, ..., K \\
&\qquad A_1 x \leq b \\
&\qquad x \geq 0
\end{aligned}$$

This could also be written into an equivalent dual form. We denote the vector with dual variables of the constraint $A_1 x \leq b$ as μ and the dual variable of the cut from iteration k for scenario ω as $\mu_{k,\omega}$. Then the dual formulation is:

$$\begin{aligned}
&\min \sum_{k=1,...,K,\omega\in\Omega} \mu_{i,\omega}(\pi_{\omega,k} b_\omega + (1 - z_\omega M)) + \mu b \\
&\text{st} \sum_{k=1,...,K,\omega\in\Omega} \mu_{i,\omega} \pi_{\omega,k} A_2 \leq c \\
&\qquad \sum_{k=1,...,K,\omega\in\Omega} \mu_{i,\omega} \leq 1
\end{aligned}$$

The general outline of the stochastic Benders method is now as follows:

Major Step 0 Constraint $\theta \leq Q(z)$ is dropped to get (MP). For all $Q(z)$ with $z \in \{0,1\}^{|\Omega|}$ drop the $|\Omega|$ constraints $\delta \leq Q(x,\omega) + (1 - z_\omega)M$.

Major Step 1 For (MP) a candidate solution $(\bar{z}, \bar{\theta})$ is tried for which optimality cuts have to be generated from $Q(\bar{z})$. In the first try θ is denoted as a random variable preferably plus infinity.

Major Step 2 To generate the optimality cuts, $Q(\bar{z})$ has to be solved and that is itself done by Benders decomposition. Set $k = 1$ and go to minor step 1.

Minor Step 1 To solve $Q(\bar{z})$, a candidate solution $(\bar{\delta}, \bar{x})$ is generated.

Minor Step 2 Subproblem(s) $Q(\bar{x}, \omega)$ has(have) to be solved to obtain optimality cuts $\delta \leq \pi_{\omega,k}(\bar{x})(b_\omega - A_2 x) + (1 - z_\omega)M$.
IF these cuts are already satisfied by $(\bar{\delta}, \bar{x})$ then that is apparently the optimal solution of $Q(\bar{z})$. Go to Major Step 3.
ELSE add the constraints to $Q(\bar{z})$, set $k = k + 1$

Major Step 3 Add constraint $\sum_{k=1,...,K,\omega\in\Omega} \mu_{i,\omega}(\pi_{\omega,k} b_\omega + (1 - z_\omega M)) + \mu b - \theta \geq 0$ to (MP).
IF the generated cuts are already satisfied by $\bar{z}$, the solution is optimal.
ELSE go to Major Step 1.

Now for the task of solving the problems $Q(\bar{z})$ obviously the same possibilities as for model II apply as these models only differ by a constant. So the traditional method, the generate all cuts method or the solve one subproblem to optimality method could be tried next to the depth search after k steps option.

4.4.4 A general solving thought

Another interesting idea would be created when a planner would be interested to see the solutions of all three models, such that she can a posteriori choose which goal she should aim for. This seems plausible for a planner who starts making more robust planning without having any idea of the consequences. Now a hypothesis would be that once a model is solved the other models can be solved quicker. The idea can be explained as follows. In solving model I with multicut stochastic Benders

decomposition, model II with Benders decomposition or model III with either of the two possibilities for Benders decomposition cuts of the form $\theta \leq \pi(\bar{x})_\omega(b_\omega - A_2 x)$ have to be generated. These cuts are also valid cuts in the other models. In other words, it is possible to add these constraints to the (MP) of the second model which is going to be solved. This gives a similar trade-off as was caused by the earlier mentioned choices for generating cuts: it gives more information, but it may cause an increase in calculation time for the (MP). We expected the result of such a trade-off to be very instance-specific. If there is one scenario which can be considered far 'worse' than the other scenarios, a first stage solution x for a maximin problem will be far different from the first stage solution for a problem with expected goal function such that the cuts give less information. This hypothesis is not studied in the literature as probably because it is usually assumed that the decision maker knows it robustness measure beforehand. There are counterexamples, for example in [12] for the case of α-reliability the input parameter α is varied from the value for which one scenario is in the reliability set until 1 such that all scenarios are in the reliability set. No considerations about efficiency in solving these problems sequentially are made.

4.5 Customer Benefits

After the literature search and assumptions made we proposed three robustness models and solution methods for these models. But what are the expectations on the benefits for customers? A brief summary is presented in this section.

The Macro Planner is used at maximum a few times per year to make a high-level strategic planning. Because this frequency of planning is so low, there is always a certain uncertainty about the demands that will be faced in the coming planning periods. However decisions made on contracts about quantities to be ordered from subcontractors are difficult to adjust later on and it directly influences the performance results of the planning later on.

It is common for companies to analyze the planning performance of the chosen planning if the demand forecast later turns out to be different after the planning is made. Here models are suggested in which possible different demand forecasts are already taken into account, such that the planning will be good or not so bad for any realization of demand instead of making a planning optimal for one realization of demand and possibly bad for the other scenarios. It is an advanced way of planning for companies acting in an uncertain environment with the purpose of better results in reality.

One of the three models can be used depending on the companies perception of 'good' for any scenario. In a scenario is for every sales forecast a quantity determined. If the expected goal function model is chosen we want to take our high level strategic decisions such that our expectation of the final performance of the planning over the possible scenarios is optimal.

If the worst case goal function is chosen we want to maximize the planning performance we obtain if the worst scenario for our planning occurs. The company has a guarantee that the performance is at least as high as this optimized value when the demand realization will turn out to be one of the specified scenarios. Of course no guarantee is available for unspecified possible scenarios. It is attractive for companies which are very risk-averse.

If the N-reliability model is chosen the planning maximizes the performance when the N-th best scenario would occur. This is attractive if the company wants to be sure the performance is of a certain level unless one of the very unlikely (N+1)-th best, (N+2)-th best, ... scenarios or an unspecified scenario occurs.

From these models it can be seen that for each risk-seeking type of company there is a useful model and a next step in planning. Risk neutral companies can take advantage of recognizing uncertainty and robust planning by maximizing their expectation

over possible demand realizations such that better results are to be expected. Risk averse companies can construct a planning in which they secure themselves against the risk of other demand realizations. In the exceptional case of a risk seeking company N-reliability with $N = 1$ can be used to maximize on the best possible demand realization.

Chapter 5

Robust Planning Results

In this chapter the results of robust planning will be explored by means of a comparison between robust planning and normal planning. This comparison is made in the Macro Planner environment. Therefore, after presenting a numerical example to give insight in the robust planning models presented in the previous chapter, we will look at considerations which came up at the implementation of the new models in the Macro Planner.

In the Macro Planner changes had to be made to make it possible to feed multiple forecast scenarios, probabilities of these scenarios and scrap prices to the model. The algorithm of the Macro Planner can be adapted such that there are three versions in which the deterministic equivalents of the expected goal function, worst-case goal function and N-reliability can be solved. This means that the algorithm had to be changed such that the right variables, constraints and goal functions were represented. Also a requirement was that the supply chain planning still could be represented in a GUI in a meaningful way.

For the purpose of comparing the robust planning results for the different robustness measures with the normal planning results we used a real dataset. This real dataset is further described in section 5.4. Thereafter the differences between the supply chain planning for the robustness measures and the normal approach for this dataset are presented using the chosen KPI's.

5.1 Numerical Example

We created a very small set of test data. The idea is that we can calculate the optimal solutions ourselves and therefore obtain insight in the results of our models. Also we can immediately spot incorrect outcomes of the Macro Planner and therefore we will use this example also in the process of verification.

First here is a description of the test data. We have one stocking point 'EndSP' in which there is demand for product 'End', the only product in our supply chain. The demand is from one customer, which represents the only sales segment in the model. There is one unit, the end product supplier. This is an external source which delivers End to our stocking point. There is only one period and the optimizer settings are such that no pre-production is possible. There are two scenario's, one scenario in which the customer is immediately persuaded to buy our product, BUY, and a scenario in which the customer hesitates and only buys some test units, TEST. The price of product END is 50 euro and the sales forecast for scenario BUY is 100 while the forecast for scenario TEST is only 25. The probability of BUY is 0.3 and therefore the probability of TEST is 0.7. There are no orders. The cost of ordering one unit at our external source is 35 euro and the scrapping price of END is -5 as we

have to pay 5 euro to let it be scrapped.

There are no fulfillment goals, sales targets, inventory targets, stocking point maxima, external source minima or maxima or any other possibility not mentioned in this text. The optimizer setting is such that both the revenue and the direct cost is weighted with factor 1 while the capacity violations are weighted with 999999, although that is unimportant because there are no maximum capacities.

Now we can see that optimizing according to the expected goal function robustness measure optimizes

$$0.3*(50*(\min\{q,100\})-5*(\max\{q-100,0\}))+0.7*(50*(\min(q,25))-5*(\max(q-25,0)))-35q$$

in which q should obviously higher or equal than 0, but in fact it is wise to chose it higher or equal than 25 as that quantity is sold for sure and on these 25 items a profit of 15 euro per item is made. Also it should be lower or equal than 100 as producing more would only yield costs for scrapping. This reduces the goal to $0.3*(50q)+0.7*(1375-5q)-35q$ on interval $[25,100]$. The solution should be $q=25$. The worst case goal function optimizes

$$50*(\min\{q,25\})-5*(\max\{q-25,0\})-35q$$

as we know by the structure of our problem that TEST will be identified as worst case scenario. This reduces to $5500-5q-35q$ for interval $[25,\infty]$. The solution will be $q=25$. Producing more would only give scrapping costs and when producing less the missed income outweighs the less scrapping costs. The same holds for N-reliability with N=2. For N=1

$$50*(\min\{q,100\})-5*(\max\{q-100,0\})-35q$$

is optimized and now it basically should have a value on interval $[0,100]$ for q. Then the expression is equal to $50q-35q$ such that $q=100$ is the optimal solution. If the probabilities for the scenarios TEST and BUY are adjusted to 0.25 and 0.75 respectively, we expect the solution for the expected goal function measure to be $q=100$ as $50p-5*(1-p)=35$ holds for $p=8/11$ and $0.75>8/11$.

We could also enrich this example using the old probabilities with a fulfillment goal. In this numerical example we will use the normal fulfillment goal measured in unfulfilled percentage. For our customer in the only period we add the fulfillment goal that at least 80% of his demand has to be fulfilled, with a cost of 80 euro per unfulfilled percent. We add the optimizer setting such that the costs of under fulfillment are weighted by a factor 1. With such a large penalty for under fulfilling the fulfillment goal, we get different objective terms and different solutions. The objective for the expected goal function robustness measure becomes

$$z=\begin{cases} 0.3*(5500-5q)+0.7*(1375-5q)-35q & \text{if } q\geq 100 \\ 0.3*(50q)+0.7*(1375-5q)-35q & \text{if } 80\leq q<100 \\ 0.3*(-6400+130q)+0.7*(1375-5q)-35q & \text{if } 25\leq q<80 \\ 0.3*(-6400+130q)+0.7*(50q)-35q & \text{if } 20\leq q<25 \\ 0.3*(-6400+130q)+0.7*(-1600+370q)-35q & \text{if } 0\leq q<20 \end{cases}$$

It is easy to check that the optimal solution is $q=80$. This is the first time we encounter a solution which is not optimal for any of the scenarios. For the worst-case robustness measure and the N-reliability measure with N=2 the objective reduces to

$$z=\begin{cases} \min\{5500-5q,1375-5q\}-35q & \text{if } q\geq 100 \\ \min\{50q,1375-5q\}-35q & \text{if } 80\leq q<100 \\ \min\{-6400+130q,1375-5q\}-35q & \text{if } 25\leq q<80 \\ \min\{-6400+130q,50q\}-35q & \text{if } 20\leq q<25 \\ \min\{-6400+130q,-1600+370q\}-35q & \text{if } 0\leq q<20 \end{cases}$$

In this case the optimal solution is $q=\frac{1555}{27}\approx 57,59$. Remark that for this candidate solution both scenario BUY and scenario TEST are identified as equally good/bad

and thus worst. For the N-reliability measure with $N = 1$ on the other hand the objective reduces to

$$z = \begin{cases} \max\{5500 - 5q, 1375 - 5q\} - 35q & \text{if } q \geq 100 \\ \max\{50q, 1375 - 5q\} - 35q & \text{if } 80 \leq q < 100 \\ \max\{-6400 + 130q, 1375 - 5q\} - 35q & \text{if } 25 \leq q < 80 \\ \max\{-6400 + 130q, 50q\} - 35q & \text{if } 20 \leq q < 25 \\ \max\{-6400 + 130q, -1600 + 370q\} - 35q & \text{if } 0 \leq q < 20 \end{cases}$$

Now the optimal solution is $q = 100$. For this candidate solution scenario TEST is identified as worst scenario and scenario BUY as worst but one, in this case best, scenario. When the probability of BUY gets higher (and the probability of TEST lower) the higher the difference in expected goal function between the solution of the expected goal function model and the worst case goal function solution and the lower the difference with the N-reliability solution with $N = 1$.

Recall that we introduced a normal fulfillment goal. If we would had a fulfillment goal in the interpretation of the models solved to compare solution times, the objective for the domain $0 < 20 < q$ will be slightly different. Not fulfilling one unit is now seen as not fulfilling 4% and therefore $320q$ was taken as penalty term in the part of the goal function determined by scenario TEST (plus $50q$ from the possible revenue gives $370q$), while when the penalty would be 80 per unit, this would be replaced by $80q$. In this example exactly the same outcomes will appear, but in general this is not true.

In the next section we will describe datasets for which the effects of a single parameter can not be identified so easily.

5.2 Macro Planner Implementation

From the point of view of Capgemini the implementation of the new robust planning ideas in the Macro Planner is in the end very important to make the robust planning ideas concrete. The considerations for this implementation consist mainly of practical issues.

The suggestions for robust planning first have been implemented in the Macro Planner. As Quintiq uses an object-oriented structure (the language is based on C++) this structure had to be changed somehow. A detailed overview of the changes is available in a document at Capgemini. It is quite possible that this design could be chosen more efficiently. A Quintiq or IT specialist may find inefficiencies and possibilities to improve the design. One example is that the structure of the Knowledge Base determines how many effort is needed to feed data to the Macro Planner. Yet no attempt has been made to find the most convenient option.

It should be mentioned that for each forecast scenario a probability is specified. Sometimes, this is unknown. We assume that if the probabilities in the knowledge base not sum up to 1 the probabilities are in fact unknown. The probabilities are then set automatically at $1/|\Omega|$ for all scenarios such that a weighted mean of values over the forecast scenarios is equal to the arithmetic mean.

In the implementation we had to correct the strange revenue calculation mentioned in chapter 2 to pass the verification checks and not specifically by the average as proposed in 4.4. While only the deterministic equivalent is solved in the Quintiq application there is no need to reduce B_ω to B or d_ω to d. In the original situation the revenue for the same sales demand can be higher in a scenario with a lower sales forecast quantity and this irrational result can become painfully clear in the KPI scoreboard when robust planning is applied.

First recall that the revenue per unit of a sales demand is determined in this demand netting process by first summing the sales forecast quantities times the revenues per quantity of these sales forecast if the total sales forecast quantity is

higher or by the order quantities times the revenue per unit of these orders otherwise. The revenue per unit follows by dividing this revenue by the total quantity of the sales demand.

In this situation the first strange effect is that a lower sales forecast quantity can lead to use the revenue per unit of the orders which is higher than those of the sales forecasts. Furthermore a second strange effect is explained with an example. Suppose you have two sales forecasts with different revenues from which a sales demand is constructed having quantities of 2000 and 3000, of which the sales forecast of 3000 has the highest revenue. If the quantity of 3000 is lowered to 2500, the average revenue per unit is lowered and therefore fulfilling a quantity of 1500 will yield a lower revenue than before. In general it should also be noted that fulfilling the first amounts of a sales demand composed of multiple sales forecasts and orders with different revenues will in practise always yield a higher revenue than the last amounts. By using only one value revenue per unit a piecewise linear function is approximated by a linear function.

Practically it was hardly possible to discover these 'mistakes' in the normal Macro Planner, but with multiple forecast scenarios it can visibly lead to an illogical result. Therefore we had to adapt the method to determine revenue. To avoid approximating the piecewise linear function the whole principle of demand netting would have to be adapted. Then after verifying the sales forecasts with the order quantities, they cannot be aggregated into one sales demand but should be treated separately. We try to adapt the method such that it keeps the error in the revenue estimate to a minimum while sticking to the ideas of the original Macro Planner.

The proposed procedure is now per sales demand:

- First determine the quantity of a sales demand by the old demand netting procedure: take the maximum of the total order quantity and the total sales forecast quantity.
- 'Include' an order or sales forecast totally in the sales demand if
  - The summed quantity of orders and sales forecast with a higher or equal revenue per unit is lower or equal than the quantity of the sales demand considered (to which the order or sales forecast should be netted).
- 'Include' an order or sales forecast partially in the sales demand if
  - The summed quantity of orders and sales forecast with a higher revenue per unit is lower than the quantity of the sales demand AND The summed quantity of orders and sales forecast with a higher or equal revenue per unit is higher than the quantity of the sales demand
- The included quantity of the order or sales forecast is then the remaining quantity divided by the number of those orders and sales forecasts with an equal revenue. Remark that this may be larger than the quantity of the order of sales forecast itself, but it will not do any harm, because it is only used for the purpose of calculating revenue.
- The order or sales forecast is not included
  - otherwise
- The average revenue per unit by summing the included quantities times the revenue per unit for the concerning order or sales forecast.

As said this will still result in an approximation of the revenue and the possibility of failing for check 5 in the following subsection.

Figure 5.1: KPI scoreboard

An important point of interest is how to design the GUI. The difficulty of visualizing a robust planning even discouraged Quintiq to work on robust planning possibilities. One issue is which KPI's to include as the KPI's now used in the Macro Planner are only meaningful for one scenario. Another aspect is the Gantt chart and other GUI components which use scenario-dependent data. All scenarios cannot be visualized on a Gantt chart at once. Now we will explain how these issues are handled.

The visualization of only one scenario is solved by letting the object structure intact and defining one scenario as main scenario. That scenario is shown in the Gantt chart and it is possible to select in the GUI another scenario as main scenario.

As there is no information from comparable problems about meaningful KPI's on robustness, we chose this ourselves. The proposal for the KPI's is to make a scoreboard which indicates per original KPI what the minimum value, the maximum value and the (weighted) expected value for the KPI is over all scenarios. This has the advantage that a detailed comparison of the outcomes for different robustness measures can be made, but we acknowledge it has also some disadvantages. One disadvantage is that it is possible for the worst-case values of the KPI's not to occur simultaneously for the same scenario, while it does under another goal. This cannot be seen from the KPI scoreboard. Another point is that the N-reliable goal function optimizes on something which does not come back in the KPI's. Remark also that even when the N is stored and used to show the N^{th} worst KPI-score over the scenarios it is questionable if it means anything as this not necessarily occurs in the N^{th} worst scenario.

From the current KPI's of profit, total revenue and demand fulfillment are clearly scenario-dependent and can be worked out in a minimum, mean and a maximum. Also the KPI on TargetInventoryLevel will now become scenario-dependent as it fully depends on the quantity and the fulfilled quantity of inventory demands. Of these two, the quantity of an inventory demand at a stocking point where sales demands takes place, is scenario-dependent when the target inventory level is a number of days times the sales demand quantity per day. Then the real inventory demand fulfilled is automatically also scenario-dependent as it cannot be higher than the quantity. Therefore this KPI also will be presented by a minimum, mean and maximum. The KPI on Inventory Cost depends on the start and end inventory level of every pispip in the supply chain planning. These depends on the total input quantity and the total output quantity at the pispip which is equal under every scenario. While the distribution of the outflow to the sales demands is different the total quantity flowing out is equal. This means that this KPI still can be represented by one value. For the total supply costs and total fixed cost the same holds. On the screen the result will look as in figure 5.1.

The implementation in the Macro Planner is cross checked with the following checks:

Check 1 When robustness measure N-reliability is chosen, a popup menu gives the possibility to choose a number N and this N should only be chosen as an integer between 1 and $|\Omega|$.

Check 2 When the probabilities in the knowledge base have strange values, which do not count up to 1, the optimizer should still give correct values and the

KPI scores should be no different than when each scenario gets assigned a probability of $1/|\Omega|$ in the knowledge base.

Check 3 It should be possible to create a sales forecast in the Macro Planner, it should be possible to edit a sales forecast and to copy it. After all this functionality is present in the normal Macro Planning. This is the only part of the functionality which is affected by the new robust planning ideas. It was for example not possible to set inventory targets, or switch from quantity to days for inventory targets.

Check 1 and 2 are checking whether the new ideas related to robust planning are working out as they are intended. Apart from importing information from the knowledge base, there used to be a possibility for adapting the knowledge in the Macro Planner itself. We still want this to be possible and therefore we have check number 3. Check 1 and check 2 are positive. Check 3 is also positive in the sense that it is possible to create a sales forecast and it is possible to set the quantity of the sales forecast for forecast scenario for the main scenario. Without changing the main scenario it is not possible to set the quantity for these other scenarios.

5.3 Algorithm Verification

After implementing these models, it is very important to check whether all algorithms do what they are required to do. These are the checks on the algorithm which must hold under trivial requirements:

Check 1 When N equals the number of scenarios $|\Omega|$ the min-max goal function planning and the N-reliability planning should yield the same result.

Check 2 When all scenarios have the same quantity for their sales forecasts, and are thus identical, the supply chain planning should yield the same result for all three robustness measures and for all possible N in the case of N-reliability.

Check 3 When we only have one forecast scenario, planning in the Robust Macro Planner according to either of the three robustness measures leads to the same result as planning with the original Macro Planner.

Check 4 If we apply the worst-case robustness measure on a dataset in which there are no fulfillment goals, product fulfillment goals, sales targets and product sales targets, the scrapping price is 0 and the quantities of the sales forecasts are of the structure that the quantity of each sales forecast in LOW is lower or equal than for the same sales forecast in MEDIUM as well as lower or equal than the same sales forecast in HIGH the result should be the same as applying the normal supply chain planning method on scenario LOW.

Check 5 For any result of the Macro Planner in which the relations between the quantities of the scenarios are satisfying the description in check 4, the revenue of scenario HIGH should be higher or equal than the revenue of scenario MEDIUM and on it's part it should be higher or equal than the revenue of scenario LOW.

Check 6 Do the algorithms perform as is supposed for our very simple test set?

In the end all checks held for our robust Macro Planner application for the tried data instances. In appendix C these chosen data instances are described.

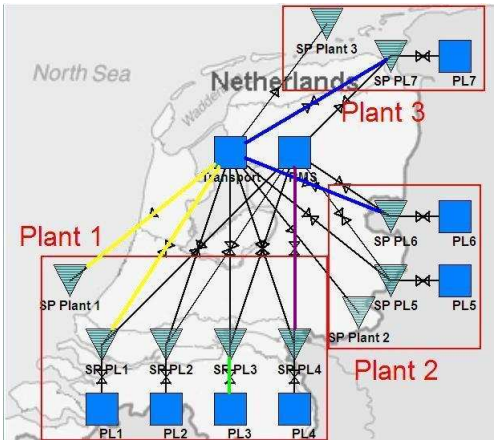

Figure 5.2: Supply Chain of Shoe box company

5.4 Shoe Box Dataset

Please note that this real dataset is anonymized in this description. The process in reality is similar to the process that is described here but we use different names. We regard the company as a shoe box producer.

To produce shoe boxes a number of steps have to be taken:

- First the raw material has to be obtained from the raw material supplier. There is one type of raw material.
- Then it has to be processed to a carton possessing the characteristics a shoe box needs. This can be obtained by processing the raw material by a product routing operation on a machine of type A to 1 carton product.
- Thereafter a print is applied on the carton. Once the carton is printed it is already clear which sizes it has and the end product is already defined. There are 14 possible printed products and consequently 14 possible finished products. The carton product can be processed by a product routing operation on a machine of type B to one of the 14 possible printed products.
- To process the printed product to its finished product, the printed product has to be cut and folded on a machine of type C and type D respectively. The folding has to occur immediately after cutting and therefore it both has to occur on a machine in the same product line. That means one product routing can be defined in which cutting and folding are subsequent product routing steps.

There are 3 plants in the supply chain. In these plants there are a number of so called product lines (4, 2 and 1 respectively) and in these product lines there can be machines of type A, B, C and D. As the distance between the product lines in a plant is substantial there is a transport cost between the plants as well as a transport cost between the product lines. However for some products there is no cost defined for the transportation between plants and it is set as default to 0.

For each product line and each plant we created a stocking point. Furthermore we create a virtual transport unit to model the situation that transport can take place between stocking points at a certain cost. Also there is a raw material supplier at which the can be ordered. A graphical representation of the network is shown in figure 5.2. Here stocking points are represented by triangles and units by squares. The abbreviation 'PL1' means product line 1 and 'SP PL1' consequently the stocking point for production line 1. All stocking points of the production lines are connected with the transport dummy and with the raw material supplier. The three plants

have all their own stocking point which is only connected to the transport dummy as output stocking point. At these stocking points sales demand occurs.

These sales demands consists of 5156 sales forecasts for a period of 18 months ahead concerning 14 products and 3 plants. Note that the number of sales forecasts in principle does not influence the computation time as sales demands are created for every product in stocking point in period and every sales segment. However, the higher the number of sales forecasts the more differences we expect to see.

In the network now four types of product routings are created:

- From the raw material supplier to a stocking point at a product line having no input and as output the raw material at the stocking point of that product line. An example in figure 5.2 is marked with the purple line. This product routing has no costs.

- From a stocking point at a product line via the transport unit to a stocking point at another product line having the same input and output products to model transportation between the product lines. In figure 5.2 an example is marked with the dark blue line. This product routing exists for all $(14 + 14 + 1 + 1) = 30$ (finished products, printed products, carton and raw material) products. The marked example is an inter-plant transportation as the product lines PL6 and PL7 are in different plants. A similar product routing between PL1 and PL2 would be an intra-plant transportation. These routings have different transportation costs.

- From a stocking point at a product line via the transport unit to the stocking point of its corresponding plant. This product routing has no costs and no duration, its only purpose is to make it possible to specify demand on a stocking point of a plant. Note that it is impossible to transport from the stocking point of the plant to a stocking point of a product line. In figure 5.2 an example is marked with the yellow line. This product routing exists for all 14 finished products.

- From a stocking point at a product line via the product line to the stocking point at that product line. This product routing represents a process and the product routing operations can be specified on the machines available in the product line. An example is marked with the green line in figure 5.2. This product line exits for $(1 + 14 + 14) = 29$ (raw material to carton, carton to 1 of the 14 printed products, 1 of the 14 printed products to 1 of the 14 finished products) different input-output relations.

For the sales forecasts the forecast quantities of the normal forecast scenario fall in the interval $[4, 512000]$ units. Actually this is a reduced number as every unit in reality accounts for 1000 units, but all other parameters are in reality specified per 1000 units. This will be called forecast scenario MEDIUM and the other forecast scenarios are derived from it. We decided to give the scenarios LOW, MEDIUM and HIGH probabilities of 0.3, 0.4 and 0.3 respectively but this was an arbitrary choice. There are no fulfillment goals, product fulfillment goals, sales targets and sales target products for this company. Once a shoe box has to be scrapped it can be recycled to be used as raw material again but this is not completely without cost. It is decided to give all shoe boxes a scrapping price of -0.01, such that scrapping 1000 shoe boxes costs the same as selling 1 shoe box yields, to discourage scrapping.

The total planning horizon consists of 18 months from 1-jan-2010 to 1-jul-2011 and there are two periods before the start of the planning horizon. The total number of products in stocking points in periods is now $(14 + 14 + 1 + 1 + 4 + 1) * 7 * 20 + (14 + 1 + 1) * 3 * 20 = 5860$. This calculation can be explained by the fact that there are three product levels, a level all products, a level with finished products, printed products, semi-finished products and raw materials, and the third level with concrete

products belonging to one of the four categories at the higher product level and also for these categories pispip's are created. So at the stocking points of product lines the total number of products equals 35 while at the stocking points of plants where only finished products can be stored the total number of products equals 16.

The choice of the optimizer setting is such that there is a weight of 1 for inventory cost, revenue and direct costs, a weight of 999999999 for exceeding the capacity and 0 for every other goal. Early supply is allowed.

5.5 Shoe Box Results

As announced we will use the dataset of a company we describe as a shoe box producer to generate multiple times a supply chain planning for a robustness measure and to generate a normal supply chain planning. This is done for forecast scenarios LOW, MEDIUM and HIGH in which MEDIUM is the original forecast quantity and LOW and HIGH are 10% lower and higher, 20% lower and higher and 30% lower and higher for setting i, ii and iii respectively.

Although this dataset does not comply completely with the description at check 4 of section 5.3 as the scrapping price is not equal to zero, the planning made with N=2 for N-reliability is equal to the normal supply chain planning. The chosen -0.01 as scrapping price is too negligible to the revenues and costs. The N-reliability planning with N=3 is obviously equal to the worst case goal function planning and therefore in tables 5.1, 5.2 and 5.3 summarizing measures on the supply chain planning are given for only four robustness measures. We compare the outcomes in terms of total supplied shoe boxes, total scrapping quantity for scenario MEDIUM, expected revenue, minimum, expected and maximum profit, inventory quantity and quantity transported. The inventory quantity is split into the raw material and the carton product on the one hand and printed products and finished products on the other hand as the first are measured in meters and the second in units. The transported quantity is split into transportation inside plants (intra-plant transport) and transportation between plants (inter-plant transport).

Although we said that the N-reliability planning with N=2 is equal to the normal supply chain planning of the Macro Planner we can observe that the number of meters in inventory and the number of units transported between plants is not equal over the three settings. This can be explained by the fact that the raw material and the carton product do not have inventory costs, and by the fact that some product routings have no specified costs such that the optimizer is indifferent between the solutions. As a consequence this problem can be seen as ill-defined, but on the other hand high level decisions as for example on the capacities of machines can be perfectly made.

After mentioning the consequences of N=3 and N=2 it is fair also to consider N=1. Then N-reliability is obviously equal to planning on forecast scenario HIGH.

Graphical representations on the total number of shoe boxes planned to be produced by the planning and the minimum, expected and maximum profits over the forecast scenarios are given in diagrams 5.3 and 5.4. It can be observed that when the expected goal function robustness measure is applied the planned number of shoe boxes to produce increases when the demand can deviate more. In other words the possible extra revenue is apparently attractive enough to risk scrapping the products. The deviation in total production between an expected goal function robust planning and the normal planning for setting iii is 1.1% which may not seem much, but if we refer to diagram 5.4 we see that the increase in expected profit by this robust planning is already larger. This means that if other high-level decisions are taken based on the other planning it will have an important consequence on the expected profit. From diagram 5.4 we also see that for this producer it is possible to increase the worst case profit by $\frac{36273772-32580042}{32580042} * 100 > 10\%$ when the worst case goal function robustness

Setting i	Expected	Worst case	N-reliability (N=1)	N-reliability (N=2)
Total Supplied shoe boxes	13380064	1296358	13712807	13338523
Total Scrapping Qty MEDIUM	51895	0	1195892	0
Expected Revenue	55745930	54701257	54142231	55385821
Expected Profit	36772644	36273772	34640215	36438047
Minimum Profit	34706602	36273772	29409851	32580042
Maximum Profit	37805738	36273772	39870579	38091477
Total Inventory (units)	1052193	999273.13	2952207.78	1761126.28
Total Inventory (m)	2920510	15564558	7800022	6391940
Total Inter-plant transport	1639259	1769091	1989822	1599249
Total Intra-plant transport	622128	580996	620366	613812

Table 5.1: Results for setting i

Setting ii	Expected	Worst case	N-reliability (N=1)	N-reliability (N=2)
Total Supplied shoe boxes	13440618	11898470	14086199	13338523
Total Scrapping Qty MEDIUM	103556	0	2202931	0
Expected Revenue	54479865	49936790	51896524	53731763
Expected Profit	35444926	33895976	31827982	34780150
Minimum Profit	30891471	33895976	22039485	27053719
Maximum Profit	37691017	33895976	41661478	38091477
Total Inventory (units)	1482506	1475703	2253629	1761126
Total Inventory (m)	3110714	3801320	10591555	3637126
Total Inter-plant transport	1722226	2621526	1343818	2059835
Total Intra-plant transport	630202	542358	628463	613812

Table 5.2: Results for setting ii

Setting iii	Expected	Worst case	N-reliability (N=1)	N-reliability (N=2)
Total Supplied shoe boxes	13487507	10411161	14451148	13338523
Total Scrapping Qty MEDIUM	151680	0	3210921	0
Expected Revenue	53173494	43694691	49571319	52064871
Expected Profit	34058155	30853705	28855327	33109382
Minimum Profit	24558424	30853705	14503530	21484493
Maximum Profit	38560865	30853705	43200557	38091477
Total Inventory (units)	1245864	1695067	3803018	1761126
Total Inventory (m)	3429822	10496493	7257938	2165837
Total Inter-plant transport	1700199	3853402	2103386	1599249
Total Intra-plant transport	635643	539994	635272	613812

Table 5.3: Results for setting iii

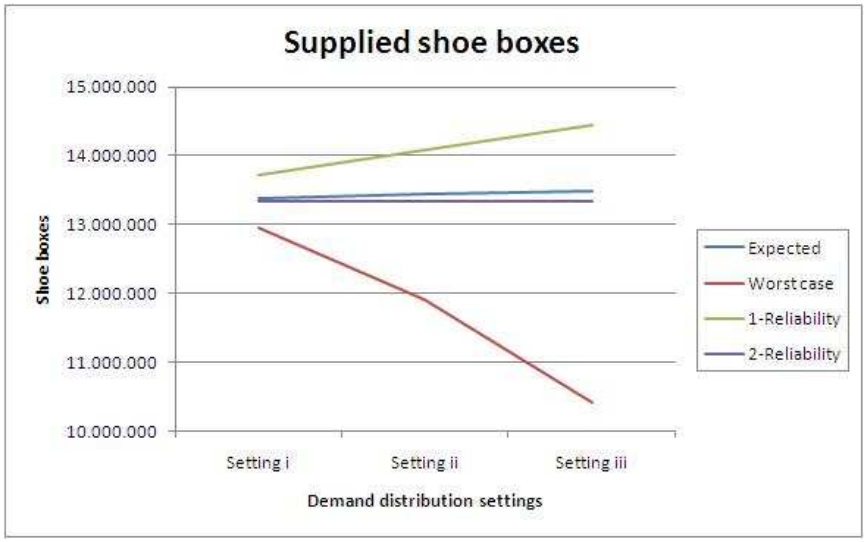

Figure 5.3: Supplied shoe boxes

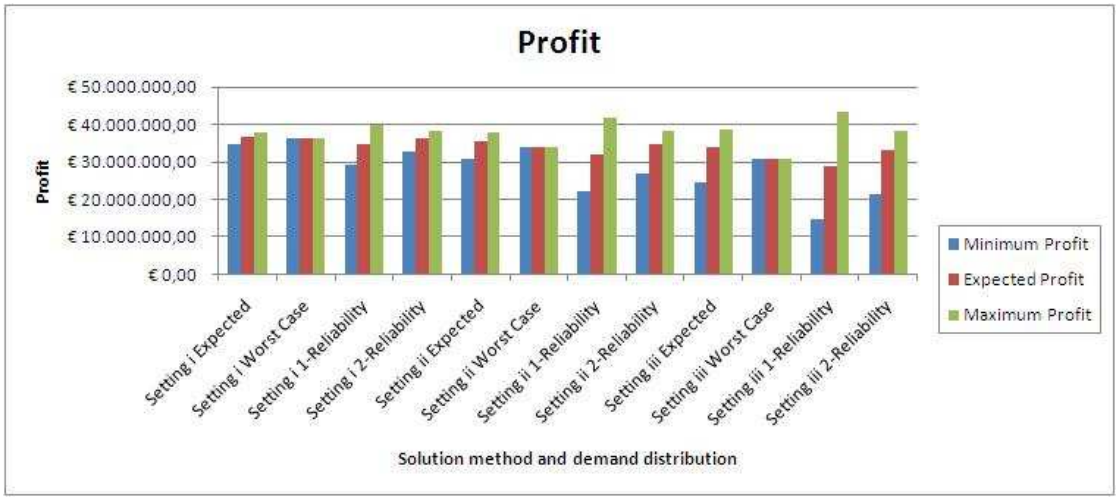

Figure 5.4: Profit

measure is applied instead of the normal supply chain planning in setting i, while the expected profit drops $\frac{36273772-36438047}{36438047} * 100 < -0.5\%$. Depending on the goals of the shoe box producer this may be very interesting.

As a result, for the shoe box producer we conclude that using the robust Macro Planner instead of the normal Macro Planner offers a lot of new possibilities with different consequences compared to the normal planning.

Chapter 6

Solution Method Results

In this chapter the results of applying the new posed decomposition methods for the model in which B_ω and d_ω are relaxed to B and d as mentioned in section 4.4 are explored. First of all we consider topics encountered at the implementation. The implementation is not done in Quintiq as the integration with CPLEX is currently not at a level such that decomposition methods can be supported. For that reason all proposed solution methods for which we will compare solution times are implemented in a JAVA application with ILOG Concert Technology. Both Capgemini and the Erasmus University have licenses for CPLEX that make it possible to run the application.

In contrary to the previous chapter this will not deal with considerations on graphical representation but solely on computional and numerical problems. Thereafter we will describe the datasets for which we tested the solution methods. Multiple datasets are chosen as the decomposition methods could have a strong performance for datasets with certain characteristics. All these datasets are taken from the Macro Planner environment as we will focus on the performance of the solution methods for Macro Planner problems. In general it may be interesting to test the solution methods for different robust planning problems, but this falls outside the scope of this thesis. In the end the generated results are presented and interpreted.

6.1 ILOG Concert Technology Implementation

The implementation of the JAVA application with ILOG Concert Technology can be seen as a sequence of two steps: constructing the problem, master problem and subproblems, and applying the algorithm to the problem.

First remarks can be made about the construction of the problem. It would be a tremendous job to rebuild all Macro Planner logic in JAVA and luckily in the Quintiq application there is a possibility to write the problem to a .lp file which can be read by CPLEX or our JAVA application. Afterwards this problem should be decomposed in the master problem and all subproblems. Unfortunately this turned out to be a time consuming process, thus somewhat limiting the possible datasets to use. On the other hand the possible datasets are also already limited by memory considerations, either the CPLEX memory or the JAVA heap space. This is a pity as there is only a chance to beat the approach of feeding the deterministic model to CPLEX for really large problems. For problems of relatively small size we will expect the loss of efficiency, caused by the fact that the methods are implemented by an amateur programmer instead of the specialists at CPLEX, to be too significant.

In the end the implemented algorithms are summarized in table 6.1. We will refer to the letters of the methods in the results section. Unfortunately it was not possible to

Reference	Model	Aggregation	Option	Special
A	Expected	Single	Traditional	
B	Expected	Single	1toOptimality	
C(δ)	Expected	Adaptive	Traditional	δ as threshold
D(δ)	Expected	Adaptive	1toOptimality	δ as threshold
E	Expected	Multi	Traditional	
F	Expected	Multi	All	
G	Expected	Multi	1toOptimality	
H	Worst	Multi	Traditional	
I	Worst	Multi	All	
J	Worst	Multi	1toOptimality	
K(k)	Worst	Multi	Traditional	Depth search after k steps
L	N-reliability	Multi	Traditional	
M	N-reliability	Multi	All	
N	N-reliability	Multi	1toOptimality	
O	N-reliability	Multi	Traditional	Nested

Table 6.1: Implemented decomposition methods

apply the idea of bunching. This certainly increases the solution times in the results and may also influence the relative strength of the decomposition methods.

The results of the algorithm are verified by considering their optimal solution values. Those values should obviously be the same for the same dataset and the same model for every solution method. However, this turned out not to hold exactly for the N-Reliability models, and most likely not due to implementation errors but to numerical errors. Because we introduced variables with really large coefficients compared to other coefficients in the problem, there is a large risk for the problem to be too ill-conditioned. When a problem is ill-conditioned a small change in the parameters would result in a large change of the solution.

In CPLEX a small tolerance for the feasibility of variables is used, the degree to which a variable may violate its (implicit) bounds represented by parameter RpEPS. In our numerical example given in section 5.1 with default setting 10^{-6} for RpEPS and bigM chosen as 10^{10}, it can be explained that we will experience problems even when solving the deterministic equivalent as $10^{10} * 10^{-6} = 10^4$ is already a higher value than the difference between the goal functions of two scenarios. Actually we wonder how the Macro Planner could deal with this problem correctly (it does) while exporting the model to a .lp file, importing the .lp file in CPLEX and solving it leads to the wrong solution.

But as said even more troublesome is that the fact that the problem may be ill-conditioned. In our results we printed a condition number for the nested Benders decomposition as it is possible to request the condition number of a linear programming problem in CPLEX. The nested Benders decomposition is the only solution method in which models are used without binary variables as decision variables, but with the large parameter values connected to bigM, namely Q(z).

A condition number is defined as $\kappa(A) = \|A^{-1}\|\|A\|$. It is unclear which matrix norm CPLEX exactly uses, but as explained in [27] if a condition number is high for one matrix norm it will be high for all matrix norms. It is not so clear-cut for which values of $\kappa(A)$ a model should be called ill-conditioned and what the rounding error consequences are. In [39] it is stated that round-off errors in the size of $2^{\kappa(A)-t}$ can occur in which $\kappa(A)$ is the condition number and t is the number of digits representing a number. Doubles are represented by 14 or 15 significant digits in JAVA. Given the condition numbers we will report in the results it is clear significant numerical errors

can occur.

With these two important numerical aspects it is not surprising that we do not reach the exact optimal value with our methods due to numerical problems. When in subproblems cuts with multiple slightly incorrect coefficients are constructed, the master problem can easily converge to a point in which for candidate solution $(\bar{x}, \bar{z}_\omega)$ a wrong value for θ is determined and therefore an incorrect solution is obtained. Then the final candidate solution may also not be the correct optimal solution. We should not exaggerate this, but in this way the incorrect optimal values for decomposition methods in the N-Reliability context can be explained.

The nested decomposition method seems more resistant for these errors as in problem Q(z) the value for the binary variables is already pinned to exactly 1 or 0. The biggest risk is that the big cuts added in the master problem that are subject to large rounding errors. Then it is possible to cause the problem to converge to the wrong optimal solutions for the binary decision variables. To get the optimal value of our decomposition method it may be more precise to request the optimal value of the last solved Q(z) than to request the optimal value of the master problem. In the normal decomposition methods the binary variables have to be chosen simultaneously with a lot of other variables and this may lead to even larger numerical errors.

It proved impossible to gather results for all these methods as memory problems occurred and/or solution times were extraordinary. For method F, I and M this can be explained by the large amount of redundant constraints added to the problem which is apparently not appreciated. Anyway, it was already questionable to what extend those methods would be really improving on just taking some random dual variables for the subproblems. Also for the D(δ) only for dataset I results could be obtained. For the other datasets JAVA heap space problems occurred. The problem here seems to be the history of an aggregation, previous cuts and previous redundancies, that needs to be stored and to be used in case of aggregation.

6.2 Data Description

The datasets we use are all based on a demo dataset, 'sales demo metal data', currently available in the Macro Planner. Although it is not strictly necessary to interpret the results here a brief description will be given to show the Macro Planner relevance of the tested datasets.

In the dataset there is raw material, slab (a semi-finished product), hot rolling coil (a semi-finished product) and two finished products, thick coil and thin coil. The planning horizon goes from 1-1-2010 to 1-1-2011 and is divided into time buckets of a month. Two months are added in front to allow pre-producing. Demand in these periods occurs only for end products thick coil and thin coil.

The network consists of 3 plants, casting, hot rolling and cold rolling, one supplier of raw material, one supplier of slab and the possibility of subcontracting the task of making hot rolled coil from slab. See figure 6.1. In between there are stocking points. All stocking points can only store one type of product, except the final stocking point CR coil SP, which can hold thick coil as well as thin coil. In total there are 9 product routings in this network, for all units but Cold Rolling in which thin coil or thick coil can be produced from hot rolling coil one.

There are maximum capacities on the stocking points of 5000 in every period and for all machines in the plants shift patterns are defined such that there is a maximum time. In all periods a maximum capacity for each subcontractor is defined. There are 16 orders and 36 sales forecasts to determine the sales demands for the whole planning horizon. The orders and sales forecasts are approximately equal divided between the two sales segments, Europe and Asia. There is one fulfillment goal

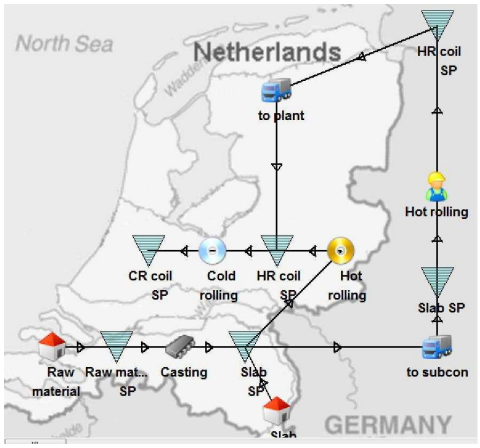

Figure 6.1: Supply Chain of Sales Demo Metal Data

which says that between 1-Jan-2010 and 1-Jan-2011 the goal is to fulfill 90% of the demand from Asia. There are no product fulfillment goals, sales targets and product sales targets.

The starting point for robust planning is this demo set with three forecast scenarios as can be seen in table 6.2. The quantities under Q1, Q2 and Q3 now belong to forecast scenario LOW, MEDIUM and HIGH respectively. The original forecast is not used, but its total forecast quantity was 93000, somewhere between the total quantity of MEDIUM and HIGH used here. We chose the scenario quantities such that the total forecast quantity for scenario HIGH was higher and the total quantity of LOW was lower than 93000. Furthermore LOW still has a higher quantity than MEDIUM for some forecasts and MEDIUM still has a higher quantity than HIGH for some forecasts. However, as mentioned before this is not necessary to prevent the worst-case robustness measure to be equal to planning for scenario LOW.

The total number of variables in the master problem for the starting dataset is 2167 and the total number of constraints is 1186, while in each subproblem the total number of variables is 7296 and the total number of constraints is 2798. After having described the starting dataset for comparing the solution times, we will mention how the dataset is adjusted to increase the problem size. It can be divided into increasing the number of forecast scenarios and increasing the number of periods. Also the spread of the quantities over the forecast scenarios is varied as some methods will performe relatively better or worse for a higher of lower spread than others.

Dataset I The demo set with three forecast scenarios as explained above.

Dataset II The original sales demo metal data with 10 forecast scenarios. The 7 new forecast scenarios are generated from a normal distribution with a mean of 2500 and a standard deviation of 750.

Dataset III The original sales demo metal data with 10 forecast scenarios. Of the new forecast scenarios 5 are generated from a normal distribution with a mean of 2500 and a standard deviation of 750. Forecast scenario number 9 is a forecast scenario in which every demand has quantity 1 and in forecast scenario 10 every demand has quantity 10000. In this way a kind of extreme spread is simulated which will influence the identification of the worst scenario strongly.

Dataset IV The original sales demo metal data with twice as many periods (24 instead of 12). In the added year the capacities and all other data are equal to the original year. This does not make the solution for the new year immediately identical as the possibility of carrying inventory to the next year should be considered.

SalesID	Product	Start	End	SalSegment	Revenue	Q1	Q2	Q3
1	Thin Coil	1-jun-2010	1-jul-2010	Asia	400	1000	3000	2900
2	Thin Coil	1-dec-2010	1-jan-2011	Europe	400	1500	1400	5000
3	Thick Coil	1-jan-2010	1-feb-2010	Asia	300	1000	3000	2900
4	Thin Coil	1-jul-2010	1-aug-2010	Asia	400	1500	1400	5000
5	Thin Coil	1-mar-2010	1-apr-2010	Asia	400	1000	3000	2900
6	Thin Coil	1-feb-2010	1-mar-2010	Asia	400	1500	1400	5000
7	Thick Coil	1-jun-2010	1-jul-2010	Asia	300	1000	3000	2900
8	Thin Coil	1-dec-2010	1-jan-2011	Asia	400	1500	1400	5000
9	Thick Coil	1-oct-2010	1-nov-2010	Asia	300	1000	3000	2900
10	Thin Coil	1-jan-2010	1-feb-2010	Europe	450	1500	1400	5000
11	Thin Coil	1-nov-2010	1-dec 2010	Europe	450	1000	3000	2900
12	Thick Coil	1-mar-2010	1-apr-2010	Asia	300	1500	1400	5000
13	Thin Coil	1-jan-2010	1-feb-2010	Asia	400	1000	3000	2900
14	Thick Coil	1-feb-2010	1-mar-2010	Asia	300	1500	1400	5000
15	Thin Coil	1-aug-2010	1-sep-2010	Asia	400	1000	3000	2900
16	Thin Coil	1-jun-2010	1-jul-2010	Europe	450	1500	1400	5000
17	Thin Coil	1-apr-2010	1-may-2010	Asia	400	1000	3000	2900
18	Thin Coil	1-oct-2010	1-nov-2010	Asia	400	1500	1400	5000
19	Thin Coil	1-oct-2010	1-nov-2010	Europe	450	1000	3000	2900
20	Thin Coil	1-may-2010	1-jul-2010	Europe	450	1500	1400	5000
21	Thin Coil	1-mar-2010	1-apr-2010	Europe	450	1000	3000	2900
22	Thin Coil	1-apr-2010	1-may-2010	Europe	450	1500	1400	5000
23	Thin Coil	1-aug-2010	1-sep-2010	Europe	450	1000	3000	2900
24	Thick Coil	1-dec-2010	1-jan-2011	Asia	300	1500	1400	5000
25	Thin Coil	1-nov-2010	1-dec-2010	Asia	400	1000	3000	2900
26	Thick Coil	1-jul-2010	1-aug-2010	Asia	300	1500	1400	5000
27	Thick Coil	1-nov-2010	1-dec-2010	Asia	300	1000	3000	2900
28	Thick Coil	1-aug-2010	1-sep-2010	Asia	300	1500	1400	5000
29	Thick Coil	1-apr-2010	1-may-2010	Asia	300	1000	3000	2900
30	Thin Coil	1-sep-2010	1-oct-2010	Europe	450	1500	1400	5000
31	Thin Coil	1-may-2010	1-jul-2010	Asia	400	1000	3000	2900
32	Thick Coil	1-sep-2010	1-oct-2010	Asia	300	1500	1400	5000
33	Thin Coil	1-feb-2010	1-mar-2010	Europe	450	1000	3000	2900
34	Thin Coil	1-sep-2010	1-oct-2010	Asia	400	1500	1400	5000
35	Thin Coil	1-jul-2010	1-aug-2010	Europe	450	1000	3000	2900
36	Thick Coil	1-may-2010	1-jul-2010	Asia	300	1500	1400	0

Table 6.2: sales metal demo data forecasts

Dataset V The original sales demo metal data with 50 forecast scenarios. The 47 new forecast scenarios are generated from a normal distribution with a mean of 2500 and a standard deviation of 750.

Dataset VI The original sales demo metal data with 50 forecast scenarios of which the first 45 new are generated as in dataset V and then there is a forecast scenario in which all demands are 10000 and a forecast scenario in which all demands are 1.

6.3 Results

In the following tables 'sub solved' stands for the amount of subproblems solved, 'agg' stands for the number of times aggregation occurred and 'final agg' stands for the final partitioning size when the algorithm terminates. The results are obtained from runs on a computer with Pentium(R) D CPU 3.00 GHz processor and 1.99 GB RAM at the university, but the tests are only performed on a single core.

As the number of results is small it is hard to come to any general conclusions, but we will try to explain the results we got here in the best way. The first observation is that the solution time of the decomposition methods is far from beating the solution time of solving the deterministic model. But this was more or less expected as we did not test for very large numbers of scenarios. There is not enough evidence to state that the solution time of the deterministic method is growing larger in the scenarios than the solution time of decomposition methods, so no further comparisons with the deterministic version are made and we will stick to comparing the decomposition methods. Remark that decomposition methods can also prove more useful when instead of CPLEX an open source LP solver has to be used.

What can be observed from the results for the expect model is that the multicut procedure is always dominant to the single-cut procedure. This is hardly surprising as the rule of thumb mentioned earlier says that the multicut method is preferable when the number of scenarios is not much larger than the size of the first stage decision dimension space. For our models the number of scenarios is clearly lower than the number of first stage decision variables.

For all datasets the performance of method C falls between the performance of A and E in every aspect. This can be explained by the fact that, although usually the optimal partitioning size is between $|\Omega|$ and 1, it is here just $|\Omega|$.

Our method B shows a bad performance overall, it needs in every dataset the most iterations and the most computation time, except for the single result for D(0.05) in dataset I. It indicates that the loss of tightness when only a part of the new cut is obtained from solving the corresponding subproblem to optimality is very significantly.

On the other hand, the idea of solving one subproblem to optimality should be regarded as an interesting for these datasets when it is applied in the multicut procedure, when always at least one of the introduced cuts is really optimal. For all the datasets method G yields the lowest number of times a subproblem has to be solved. If we compare the results of dataset II and dataset III, and dataset V and dataset VI it seems that a lower spread between the demand forecasts is advantageous for method G which correspond to the intuition that the optimal dual variables for a subproblem are a good approximation of the optimal solution for another subproblem if the difference in the right-hand side is low.

The difference in the number of iterations between dataset I and dataset IV is striking. It becomes totally clear that a large number of decision variables and a small number

Expected model	iterations	sub solved	solution value	solution time	agg	final agg	remarks
Dataset I							
Determin			$3.737172508923842 * 10^8$	125 (ms)			
method A	403	1209	$3.7371725089238423 * 10^8$	73321 (ms)			
method B	586	1105	$3.737172508923844 * 10^8$	76186 (ms)			
method C(0.05)	126	378	$3.737172508923844 * 10^8$	25890 (ms)	0	3	
method D(0.05)	585	1117	$3.73717250892384 * 10^8$	83407 (ms)	2	1	
method E	126	378	$3.737172508923843 * 10^8$	23389 (ms)			
method F							No Results
method G	298	306	$3.7371725089238405 * 10^8$	27173 (ms)			
Dataset II							
Determin			$3.678328993211823 * 10^8$	343 (ms)			
method A	463	4630	$3.6783289932118255 * 10^8$	259755 (ms)			
method B	888	4180	$3.678328993211825 * 10^8$	311254 (ms)			
method C(0.05)	82	820	$3.678328993211825 * 10^8$	54948 (ms)	2	4	
method D(0.05)							No Results
method E	68	680	$3.678328993211824 * 10^8$	40389 (ms)			
method F							No Results
method G	191	232	$3.678328993211826 * 10^8$	29140 (ms)			
Dataset III							
Determin			$-3.15607188274296 * 10^8$	297 (ms)			
method A	630	6300	$-3.1560718827428687 * 10^8$	358505 (ms)			
method B	983	5499	$-3.1560718827428686 * 10^8$	405486 (ms)			
method C(0.05)	137	1370	$-3.15607188274287 * 10^8$	94731 (ms)	5	2	
method D(0.05)							No Results
method E	81	810	$-3.15607188274287 * 10^8$	49373 (ms)			
method F							No Results
method G	309	342	$-3.15607188274286 * 10^8$	64495 (ms)			
Dataset IV							
Determin			$6.827763060268685 * 10^8$	234 (ms)			
method A	1830	5490	$6.827763060268693 * 10^8$	2519443 (ms)			
method B	2363	5027	$6.82776306028683 * 10^8$	888479 (ms)			
method C(0.05)	517	1551	$6.827763060268683 * 10^8$	320503 (ms)	0	3	
method D(0.05)							No Results
method E	517	1551	$6.827763060268683 * 10^8$	245778 (ms)			
method F							No Results
method G	1410	1422	$6.827763060268685 * 10^8$	558068 (ms)			
Dataset V							
Determin			$3.7247593263602334 * 10^8$	1827 (ms)			
method A	651	32550	$3.724759326360237 * 10^8$	1786402 (ms)			
method B	2430	60031	$3.724759326360237 * 10^8$	7100241 (ms)			
method C(0.05)	95	4750	$3.7247593263602364 * 10^8$	408049 (ms)	5	3	
method D(0.05)							No Results
method E	53	2650	$3.7247593263602364 * 10^8$	161661 (ms)			
method F							No Results
method G	260	463	$3.724759326360237 * 10^8$	288139 (ms)			
Dataset VI							
Determin			$2.287361791233809 * 10^8$	2047 (ms)			
method A	634	31700	$2.2873617912338105 * 10^8$	1765306 (ms)			
method B	1593	27923	$2.287361791233809 * 10^8$	3236744 (ms)			
method C(0.05)	74	3700	$2.2873617912338084 * 10^8$	307994 (ms)	5	5	
method D(0.05)							No Results
method E	48	2400	$2.2873617912338072 * 10^8$	146802 (ms)			
method F							No Results
method G	308	520	$2.2873617912338087 * 10^8$	361589 (ms)			

Table 6.3: Results for model expected

Worst Case	iterations	sub solved	solution value	solution time	remarks
Dataset I					
Determin			$2.6545671300675368 * 10^8$	140 (ms)	
method H	110	330	$2.6545671300675374 * 10^8$	20863 (ms)	
method I					No Results
method J	283	294	$2.6545671300675377 * 10^8$	27392 (ms)	'M' was first scenario
method J	215	226	$2.6545671300675377 * 10^8$	19739 (ms)	'L' was first scenario
method K(50)	110	330	$2.6545671300675374 * 10^8$	26580 (ms)	
method K(5)	110	330	$2.6545671300675374 * 10^8$	26579 (ms)	
Dataset II					
Determin			$2.6245348969313005 * 10^8$	390 (ms)	
method H	73	730	$2.6245348969313014 * 10^8$	43351 (ms)	
method I					No Results
method J	245	286	$2.62453489489307 * 10^8$	32358 (ms)	'A' was first scenario
method J	307	344	$2.6245348948930693 * 10^8$	45272 (ms)	'I' was first scenario
method K(50)	78	628	$2.6245348969313005 * 10^8$	43476 (ms)	
method K(5)	125	461	$2.6245348969313017 * 10^8$	35012 (ms)	
Dataset III					
Determin			$-5.986022508370819 * 10^9$	344 (ms)	
method H	73	730	$-5.9860225083708 * 10^9$	43132 (ms)	
method I					No Results
method J	64	82	$-5.98602250837081 * 10^9$	7793 (ms)	'I' was first scenario
method J	73	82	$-5.986022508370817 * 10^9$	8230 (ms)	'J' was first scenario
method K(50)	73	532	$-5.986022508370803 * 10^9$	34075 (ms)	
method K(5)	75	129	$-5.986022508370813 * 10^9$	7965 (ms)	
Dataset IV					
Determin			$4.418622295482726 * 10^8$	328 (ms)	
method H	347	1041	$4.418622295482727 * 10^8$	140188 (ms)	
method I					No Results
method J	1279	1291	$4.418622295482726 * 10^8$	433808 (ms)	'M' was first scenario
method J	1343	1354	$4.4186222954827225 * 10^8$	391784 (ms)	'L' was first scenario
method K(50)	347	1041	$4.418622295482727 * 10^8$	253532 (ms)	
method K(5)	347	1041	$4.418622295482727 * 10^8$	244553 (ms)	
Dataset V					
Determin			$2.6207209023334247 * 10^8$	2408 (ms)	
method H	57	2850	$2.6207209023334256 * 10^8$	167261 (ms)	
method I					No Results
method J	193	243	$2.6207209023334232 * 10^8$	146506 (ms)	'A1' was first scenario
method J	269	471	$2.6207209023334253 * 10^8$	294295 (ms)	'B1' was first scenario
method K(50)	57	2598	$2.6207209023334256 * 10^8$	196100 (ms)	
method K(5)	213	1263	$2.6207209023334247 * 10^8$	92976 (ms)	
Dataset VI					
Determin			$-5.986022508370816 * 10^9$	1718 (ms)	
method H	75	3750	$-5.986022508370821 * 10^9$	218551 (ms)	
method I					No Results
method J	64	161	$-5.98602250837082 * 10^9$	28196 (ms)	'A1' was first scenario
method J	75	124	$-5.986022508370815 * 10^9$	26916 (ms)	'W2' was first scenario
method K(50)	75	2574	$-5.986022508370821 * 10^9$	193844 (ms)	
method K(5)	70	364	$-5.986022508370814 * 10^9$	21448 (ms)	

Table 6.4: Results for model worst case

of scenarios is unattractive for decomposition methods. However this occurs often in Macro Planner puzzles.

It seems that for the worst case goal function models the performance of decomposition methods relative to solving the deterministic equivalent is definitely better than for the expected goal function model although the gap is still huge. Remark again that the difference between dataset I and dataset IV is large for the decomposition methods. The increased number of decision variables has a big effect.

For the worst case models we can see encouraging results for the new proposed methods compared to the classical approach, method H. For method J the starting scenario is mentioned. It is unimportant to know what the scenarios are to which the letters refer, but the idea is that method J is executed with the a posteriori worst scenario once and with an other scenario once. The differences can be quite low which indicates that sometimes optimal cuts for the other scenarios have to be found in order to prevent them to become the worst case scenario.

Especially in the case of a larger spread between the sales forecast quantities, the depth search method performs also well compared to method H. This is to be

N-Rel (N=1)	large iterations	iterations	sub solved	solution value	solution time	remarks
Dataset I						
Determin				$4.758464840950938 * 10^8$	515 (ms)	
method L		289	867	$4.7584577152504396 * 10^8$	626063 (ms)	
method M						No Results
method N		425	428	$4.75845771525044 * 10^8$	2646403 (ms)	
method O	4	1279	3837	$4.758457715250437 * 10^8$	284045 (ms)	Cond: $2.18 * 10^{14}$
Dataset II						
Determin				$4.7255062583994514 * 10^8$	2745 (ms)	
method L		316	3160	$4.725506258399451 * 10^8$	16265014 (ms)	
method M						No Results
method N		> 1000			> 24(u)	
method O	12	3779	37790	$4.72550625839945 * 10^8$	2462249 (ms)	Cond: $1.76 * 10^{14}$
Dataset III						
Determin				$4.7255062583994514 * 10^8$	998 (ms)	
method L		367	3670	$4.7255062583994544 * 10^8$	5646560 (ms)	
method M						No Results
method N		> 900			> 24(u)	
method O	11	3014	30140	$4.725506258399449 * 10^8$	2226197 (ms)	Cond: $1.94 * 10^{14}$
Dataset IV						
Determin				$8.64067918696305 * 10^8$	2059 (ms)	
method L						No Results
method M						No Results
method N						No Results
method O	4	6014	18042	$8.64067918696305 * 10^8$	3324986 (ms)	Cond: $1.54 * 10^7$

Table 6.5: Results for model N-reliability

expected as then there is one worst scenario, while otherwise all scenarios may still be worst. For our datasets apparently making traditional steps in the first k steps to identify the worst scenario does not help much as for K(5) solution times are reported that are always lower than the times for K(50).

In table 6.5 the results for N-reliability, when they were obtained, are presented. In the nested Benders decomposition the condition number of the last Q(z) solved is shown, but also for dataset IV condition numbers of the size of 10^{14} were recorded in the process. By ad hoc methods, trying different values for bigM and for the epsilon used to compare two doubles (for example in the stop criterion) and parameter RpEHS in CPLEX, we obtained results which are mostly still close enough to the optimal solution or which are optimal. Still we feel we should refrain from using decomposition methods for these ill-conditioned problems.

But if we do want to use decomposition methods, the conclusion is clear. We should use the nested decomposition method. The number of times a problem with binary variables has to be solved increases rapidly for the other decomposition methods and we even did not try $N = 25$ for the case of 50 forecast scenarios when the branch and bound procedure will become much tougher when solving problems with those binary variables. A lot of results could not be obtained due to a very large solution time. Just looking at the number of iterations reveals only the effort if we keep in mind that binary variables are involved.

Chapter 7

Other Planning Ideas

In this chapter we will consider other possible extensions to the Macro Planner. First we focus on opportunities to take CO_2 emission in account when planning with the Macro Planner, a theme which was in the beginning suggested as main subject by Capgemini. To understand that interest in CO_2 emission we will sketch its economic relevance first. There we can also link the goals of robust supply chain planning and planning on CO_2 emission. Thereafter we will consider more concretely the possibilities in the Macro Planner. The other section discusses capacity extension which can be modeled in multiple ways in the Macro Planner and link it to the issues of CO_2 emission and robust planning. The explanation why robust planning became the main subject of this thesis instead of these two interesting possible extensions is that it was interesting from an operations research point of view and in comparison with these possible extensions it was easier to obtain or simulate data.

7.1 Planning on CO_2 Emission

Greenhouse gases are the gases in the atmosphere which absorb the infrared radiation, which is emitted from earth to atmosphere as a result of solar radiation absorbed by the earth. These greenhouse gases include for example carbon dioxide (CO_2), methane (CH_4), nitrous oxides (NO_x), and chlorofluorocarbons. The growth of greenhouse gases in the atmosphere results in more infrared radiation absorption and therefore a warmer planet. This is called the greenhouse effect. To slow down the greenhouse effect, the emission of these gases should be reduced. In the past there was no incentive for companies to contribute to the reduction, but their attitude has changed. We will only mention CO_2 . After weighting the emission of the greenhouse gases with their global warming potential (GWP), CO_2 is responsible for more than 99% of the total weighted greenhouse gas emission.

7.1.1 Economic background

Companies are nowadays for various reasons taking responsibility or for their share of the greenhouse gas emission. Naturally regulation is a key driver: the Kyoto protocol formulates the goal of bringing the amount of greenhouse gas emission down to the level of 1990 and the government wants to enforce that. Their idea for reaching this goal is to shift the external costs from society to the supply chain partners. Instead of using taxes in most countries an emission trading system is used, in Europe the European Union Greenhouse Gas Emission Trading System (EU ETS). In an emission trading system on national level emission rights for a year are assigned to participating companies such that the total equals a certain cap. Within the trading system the companies can buy or sell emission rights and when at the

end of the year the emission exceeds the rights they have acquired, they have to pay a penalty. Currently most companies voluntary participate in the trading system, only for paper, utilities and cement industries participation is compulsory.

The trading has as consequence that companies with the lowest cost for reducing their CO_2 emissions are most inclined to sell their rights and in the end have to lower their emissions. Therefore it is possible to say that the advantage of a trading system over a tax system is that it ensures that the CO_2 emission is reduced to a level below the cap at the lowest cost for society. A disadvantage of the trading system is that it may prove hard for the government to choose the right cap. An example could be seen on the 15th of May 2006 when the amount of emission in the European union was presented. The cap chosen was too low such that there was a surplus of emission rights in the market and the price per ton emission dropped from 30 euro to 12 euro. When a cap is chosen too high the prices of emission rights may rise explosively such that it is more expensive to buy additional emission rights than to accept the penalty of emitting more than the rights the company has. With respect to legislation it is important to note that countries without CO_2 constraints have a competitive advantage in domestic and export which can lead to profit and possibly CO_2 leakage when customers are importing products with a larger emission from those countries. That does also explain now the Kyoto protocol runs out in 2012 the importance of the in 2009 held Copenhagen conference. If all countries would have agreed with new regulations the basis of CO_2 trading systems becomes much higher.

Other reasons for the pursue of lower CO_2 emission come from other economic players. For shareholders the environmental performance can be a non-financial indicator for the stock price. Other companies, the trading partners, can influence the sustainability policy when they prefer to trade with partners emitting less CO_2 if they are judged on the CO_2 emission of their partners.

This leads to 2 desirable consequences for the company using the Macro Planner. It wants to have a planning in which a minimum of CO_2 emission is produced for all reasons above. Moreover it wants to have an accurate forecast of the resulting CO_2 emission in advance such that it can base its trading behavior on this forecast. A CO_2 emission forecast has to be made a few times per year, just like the high level strategic planning for which we designed robust planning. Therefore we think robust planning is even more encouraging if CO_2 emission is taken into account. A user gets for each forecast scenario a clear picture of the extra rights she will have to buy or sell on the trading market if the sales forecast occurs.

7.1.2 Measuring setting

A usual way to present carbon dioxide emission is the carbon footprint of a product. It is defined as the carbon dioxide emitted across the supply chain for a single unit of that product. That includes the production, the usage and the disposal of a product. It seems reasonable to use this in the Macro Planner as well and assign such a number to each supply. However, companies normally control a part of the life process of the product and therefore one have to think over the scope of the problem in terms of the organizational boundary when using such a number in the supply chain planning. A more general scope question asks whether emissions caused by other actions than operational controlled or maybe financial controlled should be included. To give an example, the production of electricity which is used in a plant does not fall into the operational controlled actions. It is not easy to answer that question. The answer depends among others on whether the company is the largest in the supply chain, when the audience often regard the total emission as the company's responsibility, or what at the time the regulations are. In the Macro Planner we will prefer to leave all options open.

It is interesting to look at the conventions on the reporting of the emissions are. For a company it is in their interest to publish the emission results in their social

report, even if this is not obliged, if it has goals in emissions. Keeping track of emissions can also lead to benefits like managing GHG risks and identifying reduction opportunities. A definitive way of presenting these results is not established yet. See for example [1], an annual report of the Dutch traffic and transport organization, where a benchmark for the CO_2 emission is introduced, but a lot of important data is not known.

However the World Business Council for Sustainable Development and the World Resources Institute work on the 'Greenhouse Gas Protocol' which is a leading guideline nowadays. In 2001 they published the first "The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard" and subsequently they developed additional calculation tools. ISO, the International Organization for Standardization, based their guideline ISO 14064-I: "Specification with Guidance at the Organization Level for Quantification and Reporting of Greenhouse Gas Emissions and Removals" on the GHG protocol. If a company becomes member of ICROA, the International Carbon Reduction and Offset Alliance, it is obliged to follow the GHG protocol and the ISO 14064-I.

A normal approach for measuring the emission is not to measure a concentration in the air but to calculate the emission is to use a proxy measure of activity of an emission source and an emission factor with which it should be multiplied. The complete approach according to the GHG protocol is:

1. Identify GHG emissions sources
2. Select a GHG emissions calculation approach
3. Collect activity data and choose emission factors
4. Apply calculation tools
5. Roll-up GHG emissions data to corporate level

In the Macro Planner itself it would be very illogical to store values for those proxy measures and apply emission factors to them. If you compare it with the way costs are taken into account this would be extraordinary detailed. Therefore we will in the following section use parameters which should have been calculated elsewhere before applying it to the Macro Planner. As shown in [24] calculators for carbon dioxide emission of households can differ heavily in their outcomes. This will be caused by different emission factors and by different default values on measurement proxies. The calculations of these calculators lack any transparency. Therefore choosing the right way of calculation for the parameters we want to include in our Macro Planner model is a typical business intelligence problem and we prefer to refrain from get our fingers burned on that issue. In Capgemini also a calculation tool called Clint is developed, so there is some expertise in this field.

7.1.3 Macro Planner interpretation

In comparison with section 3.2 we have an extra business goal, CO_2 emission. This addition will result in a new KPI matrix like in table 7.1. We think that the effect of the supply chain planning decisions on the emission business goal can be captured in 2 KPI's:

9 Total fixed CO_2 emission: Sum of fixed emission per period of units and the opening/closing emission of units.

10 Total CO_2 emission of supply: Sum of emissions from the supply routings and inventories.

	profit	inventory	satisfaction	productivity	emission
Supply Chain Planning					
Quantity of supply to produce	(1),(2),(3),(4)	(4),(5)	(6)	(8)	(10)
Sources used for supply	(1),(3)	-	-	-	(10)
Operation allocation to machines	(1),(3)	-	-	-	(10)
Demand served by supply	(1),(2),(4)	(4),(5)	(6)	-	-
Inputs for Supply Chain Planning					
Demand netting	-	-	-	-	-
Fulfillment goal, sales target	-	-	(6)	(8)	-
Inventory target decision	-	(4)	-	-	-
Supply Chain Design	(7)	-	-	-	(9)

Table 7.1: Planning Decisions versus Business Goals

This resembles for a large extend to the structure of cost, where there is fixed cost and supply cost. Therefore the proposal for the parameters to use is also very similar of the parameters for cost. To read the rest of this section, the description of the Macro Planner given in chapter 2 does not suffices. It is required to read appendix A. We suggest to include parameter fields in the knowledge base as stated in 7.2. Some of these parameters are used to give values for fixed emission and contribute thus to determining KPI 9 while others really affect the supply chain planning and are used by the optimizer. The variables mentioned in 7.3 represent are the result of the supply chain planning.

These parameters and variables should be sufficient to make a planning when reducing CO_2 emission is an important business goal. However as we have seen in the previous subsections it is not always possible for a company to have all information to calculate these parameters at its disposal. What to do when a value for a parameter is missing?

As mentioned in [19] for example APS systems should in general not be built needing data which is not available. If a value for $\text{CO2EmissionPerQuantity}_{\text{sr}}$ is missing and is set to 0, subcontracting operations to the subcontractor of supply routing *sr* is regarded as more favorable due to the CO_2 emission of 0 and the optimizer will assign large quantities to the supply routing. This will never have been the intention of planning on CO_2 emission. Similar statements can be made for all parameters used in supply chain planning. From this example it can be no surprise that choosing a value on basis of expert opinion is therefore always favorable to any other solution. Missing parameters which are used for supply chain design are frustrating the trouble of forecasting the emission.

If there is no calculation and no expert opinion for some parameter and there is a thorough wish to consider CO_2 emission in the supply chain planning it seems best to choose the values for the missing parameters based on the other values. Our proposal would be for $\text{CO2EmissionPerQuantity}_{\text{op}}$ to take an average over the known values for $\text{CO2EmissionPerQuantity}_{\text{op}}$ in which the machine in $op = (m, so)$ is equal and the supply operation is from a supply routing that is a copy of the same product routing. If these emission quantities are all unavailable an average has to be taken over all the known values of $\text{CO2EmissionPerQuantity}_{\text{op}}$ for which the supply routing is a copy of the same product routing. Else it should be set to the average of all $\text{CO2EmissionPerQuantity}_{\text{op}}$ or if that is not possible to 0. The lesser preferred the option, the lesser plausible it is. The same can be done for $\text{CO2EmissionPerQuantity}_{\text{sr}}$ starting with the supply routings from the same product routings and then all supply routings. For $\text{EndOfLifeFootprint}_{\text{p}}$ the average over all finished products can be taken or else 0 should be chosen. For $\text{CO2EmissionPerQuantity}_{\text{pispip}}$ first the average over all *pispip*'s from the same *pisp* should be chosen , then the average over all *p* and then 0. If the option is less preferred the assumption is less plausible. As a consequence

Parameters for Emission			
Parameter	Domain	Unit of Measure	Definition
Parameters for supply chain design			
$\text{FixedCO2Emission}_{ut}$	$\forall ut \in UT$	(ton CO_2)	An emission that always occurs if the unit is not closed. Possible to construct from fixed emission per day as is given in the knowledge base
$\text{OpeningEmission}_{u}$	$\forall u \in U$	(ton CO_2)	An emission that occurs when a unit is opened or started. As the opening and closing decisions are performed manually this parameter does not come back in the optimizer.
$\text{ClosingEmission}_{u}$	$\forall u \in U$	(ton CO_2)	An emission that occurs when a unit is closed or shut down. As the opening and closing decisions are performed manually this parameter does not come back in the optimizer.
$\text{FixedCO2Emission}_{spip}$	$\forall spip \in SPIP$	($tonCO_2$)	An emission that always occur for the stocking point. To construct from fixed emission per day in the knowledge base.
TotalFixedEmission		(ton CO_2)	Total fixed CO_2 emission of all unit periods and all stocking points in periods.
Parameters used in supply chain planning			
$\text{CO2EmissionPerQuantity}_{ut}$	$\forall ut \in M \times T$	$(\frac{tonCO_2}{ton})$	Production dependent CO_2 emission for a machine, here dependent on quantity.
$\text{CO2EmissionPerHour}_{ut}$	$\forall ut \in M \times T$	$(\frac{tonCO_2}{hour})$	Production dependent CO_2 emission for a machine, here dependent on production time.
$\text{CO2EmissionPerQuantity}_{op}$	$\forall op \in OP_u, u \in M$	$(\frac{tonCO_2}{ton})$	This parameter is deduced from the former two parameters and is actually used in the planning just like $\text{CostPerQuantity}_{op}$.
$\text{CO2EmissionPerQuantity}_{sr}$	$\forall sr \in SR_{sc}, sc \in SC$	$(\frac{tonCO_2}{ton})$	The amount of emission a subcontractor generates when producing one ton. This can be seen as her carbon footprint.
$\text{CO2EmissionPerQuantity}_{pispip}$	$\forall pispip \in PISPIP$	$(\frac{tonCO_2}{ton})$	Inventory dependent CO_2 emission for a stocking point. An example would be a stocking point in which the inventory needs to be constantly mixed.
$\text{EndOfLifeFootprint}_{p}$	$\forall p \in FinalProducts$	$(\frac{tonCO_2}{ton})$	The average amount of CO_2 emitted in the rest of the life of product p, which is important if the company wants to measure the CO_2 on the broadest scope.

Table 7.2: New CO_2 parameters

Variables for Emission			
Variable	Domain	Unit of Measure	Definition
$\text{AverageFootprint}_p$	$\forall p \in P$	$\frac{\text{ton } CO_2}{\text{ton}}$	Instead of calculating a value for CO_2 emission per sales demand, a company wants to link a product with a carbon footprint value.
TotalCO2EmissionOfSupply		ton CO_2	This is in the end the value in which the company is interested.

Table 7.3: New CO_2 variables

our advice is to lower the weight for CO_2 emission in the optimizer sharply when less plausible options are chosen and take a significantly larger error margin into account. Also it might be worthwhile to reconsider the idea of incorporating CO_2 emission in the planning process if not everything can be calculated and not enough expert knowledge is available.

Apart from the advantages of planning when CO_2 emission is taken account compared to normal planning we can now return to the 2 consequences of coupling supply chain planning with CO_2 emission and the effect of robust planning we observe that CO_2 emission is one of the goals to be minimized in the supply chain planning. As the CO_2 emission results directly from the first-stage decisions, it can not be said that robust planning gives lower or for most scenarios a lower emission. However if we would introduce a parameter for CO_2 emission at scrapping of the products, robust planning tries to minimize the scrapping per scenario and then we would expect a lower or equal CO_2 emission in all scenarios in comparison with normal planning. It is possible to make a forecast on basis of the planning result in the Macro Planner which is positive anyhow. Robust planning does lead to other actions and another forecast, and if scrapping emission is taken into account multiple outcomes for the CO_2 emission depending on the scenario can be given which could help to determine a more robust emission trading strategy.

7.2 Capacity Extension

For the possibility of capacity extension we will propose the opportunities to expand or reduce the capacity in a supply chain we to take into account. We describe which costs for these opportunities are defined and by that describe the scope for the Macro Planner industry solution. This could be a starting point for the actual implementation of capacity extension possibilities. We recognize the differences in scope between users of the Macro Planner and describe them. We do not expect a problem with data availability as these capacity extension problems will have been encountered in the past such that there is some idea about the costs involved.

First observe the relation between this capacity extension idea and our earlier ideas for the Macro Planner. When making a robust planning in which capacity extension is taken into account it is more likely that terribly low occupancy rates for plants or stocking points are avoided and the chance that the capacity limits the production wishes of the company is also lowered. In some cases also an interesting phenomenon could occur if we combine capacity extension and CO_2 emission. It can indicate that a machine should be replaced by a newer machine that produces fewer CO_2 which is a decision that maybe would not have occurred immediately to the planner.

Capacities in owned resources occur in the quantity of stocking points and the number of time available on the machines. If capacity is a bottleneck in the supply chain

planning those are resources at which something could be changed. The capacity of a stocking point can change in many ways. We assume that in every period it is possible to buy or sell capacity of a stocking point. The change in capacity will come back in all subsequent periods by the nature of buying and selling. The other possibility which will be encountered in practice frequently is to hire capacity in a period. This has only effect on that period. On the other hand we assumed it is not possible to rent capacity of stocking points. Experience has shown that it is often undesirable to allow other parties in parts of the stocking point.

In our proposal buying or selling capacity results in a certain fixed cost for the transaction, a variable cost for the amount bought and a variable cost for the amount sold. The separation between selling and buying is debatable. The purpose of this separation is the possibility to penalize buying an amount and selling the same amount in the next period compared to keeping the capacity constant. However the presence of fixed cost has already the same purpose. Also for hiring we define a fixed hiring cost and a variable cost proportional to the capacity hired. We assume the product capacity in a stocking point to be proportional to the total capacity. Therefore no separate decisions on the product capacity have to be made. Obviously a requirement is that the 'new' capacity per period is always higher or equal than zero.

The other possibility to increase or decrease capacity in the supply chain is in the plants. It could be possible to buy or sell a machine or to work more hours on a machine than originally specified by the shift pattern. We always assume that the same operation times are used for machines of one type. We also assume that there are always enough employees to use the machines or that the cost of hiring and firing employees is incorporated in the costs of buying and selling machines or changing the number of hours worked. The effects of these actions can be classified as permanently and temporarily. A changed number of machines in a period is used as input for all subsequent periods. Operating a number of hours different than originally specified by the shift pattern is applied only in the period itself and not taken forward to the subsequent periods. Obviously the resulting 'new' number of machines and time per machine should be larger or equal than 0 in each period.

When buying or selling machines there is a variable cost or revenue for the unit itself, but we assume there are no costs associated with finding a buyer or seller as the possible partners are already known from previous transaction. Therefore we propose not to include fixed cost. Also this part of the proposal is debatable when the placement or removal from a machine is a large operation. A fixed cost for changing the number of hours a machine is operated, cost for rescheduling employees, and a variable cost depending on the number of hours changed can be defined. If this amount of time is negative, the variable cost may be negative as well.

This would be our proposal and in Macro Planner terms the model is written out in appendix B. Here a problem can be observed. When both the number of machines and the number of hours worked on a machine is variable, the capacity in time is a product of two variables and a non-linear term appears in a constraint. For this undesirable effect a workaround can be thought of. At the moment it seems better only to make one of the two decisions, number of operating hours and number of machines, variable. An experienced planner might list a limited amount of attractive options for the number of machines to be bought. It seems very reasonable that the only sensible options are to buy or sell 1 machine. Then the decision of buying and selling machines should remain the decision to be made manually in the supply chain design.

For a specific company, apart from the choices already given as debatable, more parts of this description could not hold. If there are no fixed cost for buying or hiring stocking point capacity, which could be for example when there is a regular renter from which capacity is hired such that there are no 'renter orientation costs', it is not

most optimal to set these costs to 0. Immediately deleting all the binary variables will speed up calculation time. The same holds of course for fixed costs incurred at changing the operating time of a machine.

Besides it is to be expected for companies to have only a discrete set of possibilities for hiring, buying and selling stocking point capacity. Hiring inventory space can be hiring a building, hiring a box or anything and hiring 1.2 m^2 does not seems to be an option. Just as when transportation should occur in batches in the normal Macro Planner instead of handling this discreteness we propose to search first for a continuous solution and than solve multiple models in which the capacity extensions or reductions is fixed to values around the values if the first solution.

After this proposal a final note on capacity extension in the context of robustness can be added. In literature the favorite problems to study in robust planning are P-median models, Uncapacitated Facility Location Problem models and Capacitated Facility Location Problem models. In all the cases when a recourse model is applied to these problems the first-stage decisions are supply chain design decisions, the decision where to place the plants. The second stage decisions which are made after information on the demand is revealed are the assignment of customers to plants from which they are served, the supply process.

Translating this to our case, if a supply chain design decision as capacity extension is very important for the Macro Planner user, it may be worth to develop another planning application on basis of which these supply chain design decisions are made. In that application the capacity decisions are made before information is revealed and the supply decisions are only made after the information is revealed. This is sensible when management has more possibilities to adjust the decisions made on the planning than the possibilities to adjust capacities which is mostly the case in practise. The link with the original Macro Planner for this application would be very limited but it is an interesting direction to explore. After the decisions on the capacity are made, the Macro Planner without capacity extension can be used to make the supply chain planning.

Chapter 8

Conclusions and Recommendations

8.1 Conclusion

To construct a 'good' supply chain planning for multiple forecast scenarios instead of only one scenario as is current practice, we defined three two-stage recourse models. After literature research we find these three models as representing different types of risk-seeking behavior and having models for risk as well as uncertainty. For the Macro Planner a concrete separation into first-stage and second-stage decisions is made and this is implemented in Quintiq with a slightly adjusted underlying structure and GUI design to get a new Robust Macro Planner.

When we tested this on a real dataset the supply chain planning for the expected goal function robustness measure is different than the normal supply chain planning. In this example we saw for example that the total number of produced shoe boxes increases if it is possible to have higher deviations (higher or lower) than the expected demand. This would make it possible to increase the expected profit with 10% while planning according to the worst case goal function robustness measure would increase the worst possible profit significantly without affecting the expected profit too much. These effects will be different for every problem instance of course, but at least it shows that there can be differences between a robust supply chain planning and a normal supply chain planning.

We tried also alternative decomposition methods. This was however for a slight adjustment in our model which allowed only right-hand side uncertainty. It should be kept in mind that it is not possible to use a percentage-wise fulfillment goal anymore. While these stochastic Benders decomposition methods are common in robust optimization, we faced less-studied models as the worst case and N-reliability models and we faced only right-hand uncertainty. To exploit these characteristics we came up with new options in the decomposition methods.

For the Macro Planner type of datasets we considered there was no possibility to beat CPLEX as it was not possible to test really large examples among others due to the time-consuming process of exporting the model from the Quintiq application and importing it with ILOG Concert Technology. For small examples the it is less likely to beat CPLEX and especially because CPLEX specialists program more efficient. However, our decomposition methods may be more useful when CPLEX is not available.

If we compare the decomposition methods against each other we see according to our expectation that for the expected goal function model the multicut procedure dominates the single-cut procedure. For these datasets the idea of solving one sub-

problem to optimality could be an idea in the multicut case. For the worst case model however the traditional method is performing less than our solve one subproblem to optimality method and the depth search after k steps method. Depending on the data characteristics, for example spread in forecasts, this may save a huge amount of computation time. In implementing the decomposition methods for the N-Reliability case we encountered numerical problems as this model can be ill-conditioned. If we would apply decomposition models for N-Reliability we would advise the nested Benders decomposition, but it may be wiser to refrain from it at all.

Other possibilities of increasing the attractiveness of the Macro Planner mainly came back in chapter 7 and we will refer to it in the next section.

8.2 Recommendations

I do not claim to provide an application modeling all supply chain planning aspects for an arbitrary or even any situation, but merely provide a framework for choosing a right direction when situations in which topics as robustness are interesting are encountered in practise. Therefore we will give here a few recommendations for further research that suggested themselves during writing this thesis.

At first to answer the part of the main question which other possibilities could be suggested, a reference to chapter 7 can be made. Two promising ideas for increasing the attractiveness, incorporating CO_2 emission or incorporating capacity extension, are considered and although this is not made concrete, a lot of suggestions are made.

About the models proposed for robust planning in the Macro Planner it could be said that as mentioned multi-stage models may be interesting when one takes into account that the Macro Planner is not run once but maybe a few times per year. Also in the Macro Planner models it might still be interesting to introduce recourse variables for extra carried forward inventory or a combination of these variables with the current variables.

On the decomposition methods it could be said that they have to be tested in non-Macro Planner problems with an enormous amount of scenarios to establish their worth in robust planning in general. Also the possibility of bunching should be incorporated as this could decrease the solution time without doubt, but could not be implemented in our application. Another point on which we did not elaborate in this thesis is how to get a rule of thumb for choosing a right value for k in the depth search after k steps method. This will be interesting when the method is used for more problems.

Appendix A

Mathematical Model

In this section the mathematical model currently used for solving the planning problem in the Macro Planner is presented. In total there exist 3 algorithms in the logic behind the Macro Planner. Two of them are supporting manual actions in the Macro Planner. One calculates the maximum quantity it is possible to plan on a supply routing without violating capacity constraints. This is displayed as support for the manual decision of creating a new supply routing connected to a new supply. The other algorithm processes a manually created supply by allocating the operations in an optimal way to the machines in a unit on which the operation can be carried out. This allocation cannot be influenced by the user, but probably it is also very seldom that a user is wanting to choose machines themselves in a strategic planning. The focus is on the third and most important algorithm called CapacityPlanningAlgorithm, which makes an automatic planning for the whole supply chain based on the orders and forecasts. In this chapter that algorithm will be described, and all parts of the Macro Planner which are not used by and not relevant for the CapacityPlanningAlgorithm are omitted in this description.

A.1 Assumptions

In this mathematical model there are several assumptions made.

1. In the goal function only a cost term on the carried forward inventory at the end of the period is used to model inventory costs, while in reality these have to be paid on both end-inventory and inventory growth during a period.

2. Demand is continuously taking place over the time in a period. And if an operation or supply routing is processed in a fraction of a period, it is regarded as being spread and continuously taking place over the total period such that it can also fulfill demand taking place at the beginning of the period. This is equivalent with the assumption that demand is taken place at the end of the period. This assumption is standard for most problems in the supply chain optimization area like lot-sizing problems.

3. There is only continuous transport in the supply chain and no bulk transport is taking place. Therefore no integer variables are needed to represent transport.

4. The cost structure is such that there are costs per quantity and/or costs per period and/or opening and closing costs based on opening and closing events specified in advance in the supply chain design for the units in the model. So there is no possibility of expressing costs for starting and shutting down machines in the supply chain design.

5. Costs and revenues are not discounted: cash flows in the future are valued the same as cash flows at the start of the planning period. This is common for small models. Only if the planning horizon becomes very large it will become interesting to drop this assumption.

In practise there will occur situations in which these assumptions won't hold exactly. When there is bulk transport in the supply chain, for example a batch of 500 should be transported, there have to be set up some kind of branch or enumeration approach. If y is the outcome of the variable for the transport in the solution of a linear programming model, the possibilities should then be branched on $\lfloor y/500 \rfloor \times 500$ and $\lceil y/500 \rceil \times 500$ as value for the concerned transport. So the algorithm should then be executed multiple times, but the big amount of time concerned with the import of knowledge bases to get the parameters when creating scenario's only has to be done once. The results from the LP runs usually come very quick. However, it should be kept in mind that this approach with rounding on the variables which are required to be integer does not guarantee an optimal solution. An example:

$$\begin{aligned}
\max \quad & 3x_1 + 1.6x_2 && && && \text{(A.1)}\\
\text{st} \quad & 6x_1 + 4_2 && \leq && 24 && \text{(A.2)}\\
& -3x_1 + 2x_2 && \leq && 0 && \text{(A.3)}\\
& x_1 && \leq && 3.5 && \text{(A.4)}\\
& x_1, x_2 \in \mathbb{N} && && && \text{(A.5)}
\end{aligned}$$

The solution of the LP-relaxation is (3.5,0.75). When this is branched on the integer values which will be obtained when rounding the solution, it is not possible to get the optimal solution of (2,3) any more. An even more extreme example is:

$$\begin{aligned}
\max \quad & x_2 && && && \text{(A.6)}\\
\text{st} \quad & -0.2x_1 + x_2 && \geq && 0.7 && \text{(A.7)}\\
& -0.2x_1 + x_2 && \leq && 0.9 && \text{(A.8)}\\
& x_1 && \leq && 2.5 && \text{(A.9)}\\
& x_1 && \geq && -0.5 && \text{(A.10)}\\
& x_1, x_2 \in \mathbb{Z} && && && \text{(A.11)}
\end{aligned}$$

The solution of the LP-relaxation is (2.5,1.4). All rounded solutions (2,1),(2,2),(3,1) and (3,2) are infeasible as there is in fact only one feasible and therefore optimal solution (1,1).

A.2 Sets

In table A.1 all the sets used in the mathematical model are listed. The sets are usually objects in the Macro Planner logic, and they can be subdivided in different categories. Some sets deal with the supply chain design, the fundamental of the whole system. Some sets represent the different demands, which is sometimes defined on a part of the supply chain. Also some sets deal with supply, the variables on these sets are the outcome for the planning. These supplies can also be defined on sets from the supply chain design.

In the definition of the sets is should be pointed out that in reality there are product routings. Each supply routing is a copy of one product routing, which actually has all the parameters now assigned to sets of supply routings or supply steps or supply operations. So these parameters of supply routings is derived from the parameters of the underlying product routings and only the variables for the quantity of a supply routing, supply steps and supply operations of the supply routing, are determined per supply routing. However, for the sake of simplicity here only sets are created for supply routings. This does create more parameters, but has no influence on the number of variables such that the model size is actually the same and this formulation

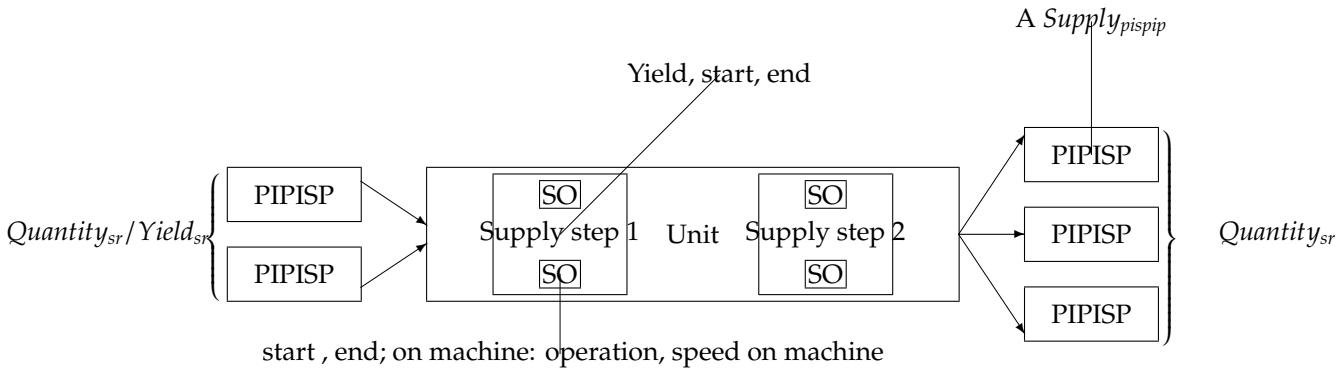

Figure A.1: Detailed Routing Diagram

is equivalent. Furthermore the term PeriodTask is omitted. This is just equivalent to part of an operation in a period which can be written down as (an operation, a period). For clarity purposes the supply routing diagram is given here in more details, A.1. The clear difference with a product routing diagram is that now periods and quantities are mentioned.

Sets in Planning Model			
Set	Construction	Index	Description
Supply Chain Design Sets			
PL		pl	Set of own plants and controlled transport nodes.
M		m	Set of machines/vehicles (or other units which have a non-quantity type of capacity)
SC		sc	Set of subcontractors (or other units which have a quantity type of capacity)
U	$PL \cup M \cup SC$	u	Set of all units
SP		sp	Set of stocking points
T		t	Set of periods
P		p	Set of products
UT	$U \times T$	ut	Set of unit periods
SalS		sals	Set of sales segments
FG		fg	Set of fulfillment goals
FG_{sals}	$\{fg \in FG \mid FGinSalS_{fg,sals} = 1\}$	fg	Set of fulfillment goals which are specified for sales segment sals
FGP		fgp	Set of product fulfillment goals
FGP_{sals}	$\{fgp \in FGP \mid FGPinSalS_{fgp,sals} = 1\}$	fgp	Set of product fulfillment goals which are specified for sales segment sals
ST		st	Set of sales targets
ST_{sals}	$\{st \in ST \mid STinSalS_{st,sals} = 1\}$	st	Set of sales targets which are specified for sales segment sals
STP		stp	Set of products sales targets
STP_{sals}	$\{stp \in STP \mid FGPinSalS_{stp,sals} = 1\}$	stp	Set of product sales targets which are specified for sales segment sals
SPIP	$SP \times T$	spip	Set of stocking-point-in-periods
PISP	$P \times SP$	pisp	Set of products-in-stocking-point

P_{sp}	$\{p \in P \mid ProdSPAss_{p,sp} = 1\}$	p	Set of products allowed to being stored in stocking point sp
PISPIP	$P_{sp} \times SP \times T$	pispip	Set of product-in-stocking-point-in-periods
UTP	$\{(ut, p) \in SC \times P \mid CapPinUT_{ut,p} = 1\}$	(ut, p)	Set of product in unit periods for which product capacities are defined
	Demand Sets		
D		d	Set of demands
Dd	$Dd \subseteq D$	dd	Set of dependent demands
Sd	$Sd \subseteq D$	sd	Set of sales demands
Id	$Id \subseteq D$	id	Set of inventory demands
Id_{pispip}	$\{id \in Id \mid DinPISPIP_{id,pispip} = 1\}$	id	The inventory demand for pispip
Sd_{pispip}	$\{sd \in Sd \mid DinPISPIP_{sd,pispip} = 1\}$	sd	Set of sales demand for pispip
Sd_{sals}	$\{sd \in SD \mid SdinSalS_{sd,sals} = 1\}$	sd	Set of sales demand for a sales segment
Dd_{pispip}	$\{dd \in Dd \mid DinPISPIP_{dd,pispip} = 1\}$	dd	Set of dependent demand for pispip
D_{pispip}	$Id_{pispip} \cup Dd_{pispip} \cup \{\cup Sd_{pispip,sals}\}$	d	Set of demand for pispip
	Supply Sets		
SR		sr	Set of supply routings
SR_u	$\{sr \in SR \mid SupRoutonU_{u,sr} = 1\}$	sr	Set of supply routings defined on unit u
SS		ss	Set of supply steps
SO		so	Set of supply operations
SS_{sr}	$\{ss \in SS \mid SSinSR_{sr,ss} = 1\}$	ss	Set of supply steps of a supply routing
SO_{ss}	$\{so \in SO \mid SOinSS_{ss,so} = 1\}$	so	Set of supply operations of a supply step
OP_u	$\{(so, u) \in SO \times SC \cup M\} \mid SOonU_{so,u} = 1\}$	op	Set of (supply) operations on unit u (=machine m or subcontractor sc)
NS		ns	Set of new supplies
IS		is	Set of inventory supplies, 'from outside'
NS_{pispip}	$\{ns \in NS \mid SinPispip_{ns,pispip} = 1\}$	ns	Set of new supplies for a pispip
IS_{pispip}	$\{is \in IS \mid SinPispip_{is,pispip} = 1\}$	is	Set of inventory supplies for a pispip
SR_{ns}	$\{sr \in SR \mid NSfromSR_{ns,sr} = 1\}$	sr	The supply routing which results in new supply ns
SRI_{pisp}	$\{sr \in SR \mid IsInput_{pisp,sr} = 1\}$	sr	Set of supply routings for which pisp is used as input

SRO_{pisp}	$\{sr \in SR \mid IsOutput_{pisp,sr} = 1\}$	sr	Set of supply routings for which pisp is used as output

Table A.1: Overview of Sets

A.3 Parameters

In table A.2 all the parameters used in the mathematical model are listed. As there are many parameters, the table consists of multiple pages. To get some order in the parameters they are divided in categories. First there are the 0-1 parameters describing whether or not there is a relation between two elements of sets (objects). This is mainly used for constructing conditioned sets in this model and called 'Relation Parameters'. Then there is a category of parameters describing the supply chain design. This concerns capacity data, but also targets of the producer for other business goals. Then there are 'Parameters for Demand' as demands have not only a quantity but also a revenue and more unique characteristics. Also there are 'Parameters related to Supply', which is mostly about the supply routings resulting in new supplies. At last there are parameters which are only used in the goal function and used to balance the different goals, 'Goal Function Parameters'. The values for these parameters for the optimizer are set by the user. It is possible to create different optimizer settings which can be selected to make a planning.

Not all of these parameters will be immediately clear and these parameters will be explained further in this section. When there are relations between parameters these will be presented. In fact, some parameters can also be substituted by the other parameters.

Parameters in Planning Model			
Parameter	Domain	Unit of Measure	Definition
Relation Parameters			
$SupRoutonU_{u,sr}$	$\forall u \in PL \cup SC, sr \in SR$		1 supply routing sr is defined on unit u; 0 otherwise
$NSfromSR_{ns,sr}$	$\forall ns \in NS, sr \in SR$		1 if supply routing sr results in new supply ns; 0 otherwise
$SSinSR_{sr,ss}$	$\forall sr \in SR, ss \in SS$		1 if supply step ss is included in supply routing sr; 0 otherwise
$SOinSS_{ss,so}$	$\forall ss \in SS, so \in SO$		1 if supply operation so is included in supply step ss; 0 otherwise
$SinPISPIP_{s,pispip}$	$\forall s \in \{NS \cup IS\}, pispip \in PISPIP$		1 if supply s delivers a quantity to pispip; 0 otherwise
$IsInput_{pisp,sr}$	$\forall pisp \in PISP, sr \in SR$		1 if pisp is an input for supply routing sr; 0 otherwise
$IsOutput_{pisp,sr}$	$\forall pisp \in PISP, sr \in SR$		1 if pisp is an output for supply routing sr; 0 otherwise
$DinPISPIP_{d,pispip}$	$\forall d \in D, pispip \in PISPIP$		1 if demand d is defined for pispip; 0 otherwise
$CapPinUT_{ut,p}$	$\forall ut \in SC \times T, p \in P$		1 if for product p in unit period ut capacities are defined; 0 otherwise
$ProdSPAss_{p,sp}$	$\forall p \in P, sp \in SP$		1 if stocking point is assigned to product p; 0 otherwise
$SdinSalS_{sd,sals}$	$\forall sd \in Sd, sals \in SalS$		1 if sales demand sd is a demand from sales segment sals; 0 otherwise
$SOonU_{so,u}$	$\forall so \in SO, u \in M \cup SC$		1 so can be processed on unit u (=machine m or subcontractor sc); 0 otherwise
$STPinSalS_{stp,sals}$	$stp \in STP, sals \in SalS$		1 if stp is defined on segment sals; 0 otherwise
$FGPinSalS_{fgp,sals}$	$fgp \in FGP, sals \in SalS$		1 if fgp is defined on segment sals; 0 otherwise
$STinSalS_{st,sals}$	$st \in ST, sals \in SalS$		1 if st is defined on segment sals; 0 otherwise
$FGinSalS_{fg,sals}$	$fg \in FG, sals \in SalS$		1 if fg is defined on segment sals; 0 otherwise
$FGPonP_{fgp,p}$	$\forall fgp \in FGP, p \in P$		1 if fgp is a fulfillment goal on product p; 0 otherwise
Parameters for supply chain design			

$YearlyInterestRate$		(%)	The estimation of yearly interest rate taken for the cost calculations.
$InventoryItemPrice_{pispip}$	$\forall pispip \in PISIP$	($)	The price for that product in a stocking point for that period.
$TotalAvailableCapacity_{ut}$	$\forall ut \in SC \times T$	(ton)	Maximum capacity available in quantity-based unit in a period
$MinCapacity_{ut}$	$\forall ut \in SC \times T$	(ton)	Minimum amount to be processed in quantity-based unit in a period.
$TotalAvailableTime_{ut}$	$\forall ut \in M \times T$	(hour)	Total time available on a time-based unit in a period
$MaximumProductCapacity_{ut,p}$	$\forall (ut, p) \in UTP$	(ton)	Maximum capacity specified in quantity-based unit in a period for a product.
$MinProductCapacity_{ut,p}$	$\forall (ut, p) \in UTP$	(ton)	Minimum level specified in quantity-based unit in a period for a product.
$MaxCapacity_{spip}$	$\forall (spip \in SPIP)$	(ton)	The maximum inventory level of a stocking point in a period
$MaxCapacity_{pispip}$	$\forall pispip \in PISPIP$	(ton)	The maximum inventory level of a product in a stocking point in a period
$FulfillmentPercent_{fg}$	$\forall fg \in FG$	(%)	The goal percentage of fulfillment goal fg
$FulfillmentPercent_{fgp}$	$\forall fgp \in FGP$	(%)	The goal percentage of product fulfillment goal fgp
$Start_{fgp}$	$\forall fgp \in FGP$	(date:hour)	Time at which product fulfillment goal fgp starts
End_{fgp}	$\forall fgp \in FGP$	(date:hour)	Time at which product fulfillment goal fgp ends
$MinimumQuantity_{st}$	$\forall st \in ST$	(ton)	The minimum quantity to be sold for sales target st.
$MaximumQuantity_{st}$	$\forall st \in ST$	(ton)	The maximum quantity to be sold for sales target st.
$MinimumQuantity_{stp}$	$\forall stp \in STP$	(ton)	The minimum quantity to be sold for product sales target stp.
$MaximumQuantity_{stp}$	$\forall stp \in STP$	(ton)	The maximum quantity to be sold for product sales target stp.
$Start_{stp}$	$\forall stp \in STP$	(date:hour)	Time at which product sales target stp starts
End_{stp}	$\forall stp \in STP$	(date:hour)	Time at which product sales target stp ends
$Start_t$	$\forall t \in T$	(date:hour)	Time at which discrete period t starts

End_t	$\forall t \in T$	(date:hour)	Time at which discrete period t ends
$TargetInventoryLevel_{pispip}$	$\forall pispip \in PISPIP$	(ton)	The target for the inventory level at the end of the period for pispip.
Parameters for demand			
$Quantity_{sd}$	$\forall sd \in Sd$	(ton)	The quantity demanded by sd
$Start_{sd}$	$\forall sd \in Sd$	(date:hour)	Start of sd
$Quantity_{id}$	$\forall id \in Id$	(ton)	The quantity demanded by id
$RevenuePerQuantity_{sd}$	$\forall sd \in Sd$	($/ton)	The revenue per fulfilled quantity for demand sd
$Priority_{sd}$	$\forall sd \in Sd$		A real number to express the priority of the sales demand in comparison with other sales demands
Parameters related to Supply			
$Quantity_{is}$	$\forall is \in IS$	(ton)	The quantity of inexplicable extra inventory from outside
$Yield_{sr}$	$\forall sr \in SR$		The total yield of the supply routing.
$CombinedYield_{ss}$	$\forall ss \in SS$		The yield of all next supply steps in the supply routing. This excludes the yield of step ss itself.
$Yield_{so}$	$\forall so \in SO$		The ratio of total output quantity and total input quantity for a supply operation.
$SequenceNr_{ss}$	$\forall sr \in SR, ss \in SS_{sr}$		The place in the sequence of supply steps of supply routing sr.
$InputFactor_{pisp,sr}$	$\forall (pisp, sr) \in PISP \times SRI_{pisp}$		The fraction of input quantity obtained from this pisp.
$OutputFactor_{pisp,sr}$	$\forall \{(pisp, sr) \in PISP \times SRO_{pisp}\}$		The fraction of output quantity going to this pisp.
$LeadTime_{sr}$	$\forall sr \in SR$	(day:hour)	Duration of processes in getting the product to the first supply step
$CycleTime_{ss}$	$\forall ss \in SS$	(day:hour)	Time it will take before the first product in the supply step is finished after the production is started.
$PreferenceBonus_{sr}$	$\forall sr \in SR$	($/ton)	Preference bonus for the optimizer per unit fulfilled via this supply routing.

$RequiredCapacity_{op}$	$\forall u \in M \cup SCop \in OP_u$	(ton/hour)	The speed at which a supply operation is processed on a machine.
$CostPerQuantity_{op}$	$\forall u \in M \cup SCop \in OP_u$	($/ton)	The cost of processing one ton by operation op.
$CostPerQuantity_{sr}$	$\forall sc \in SC, sr \in SR_{sc}$	($/ton)	The cost of processing one ton by supply routing sr.
$CostPerTon_{ut}$	$\forall ut \in M \times T$	($/ton)	The cost of processing one ton on unit u in period t.
$CostPerHour_{ut}$	$\forall ut \in M \times T$	($/hour)	The cost of processing one hour on unit u in period t.
$Start_{ss}$	$\forall ss \in SS$	(date:hour)	Start of supply step ss
End_{ss}	$\forall ss \in SS$	(date:hour)	Start of supply step ss
$Start_{so}$	$\forall so \in SO$	(date:hour)	Start of supply operation so
End_{so}	$\forall ss \in SO$	(date:hour)	Start of supply operation so
$Start_{op}$	$\forall op \in OP$	(date:hour)	Start of operation op
End_{op}	$\forall op \in OP$	(date:hour)	Start of operation op
$EndProd_{sr}$	$\forall sr \in SR$	(date:hour)	Real start of production in supply routing sr
$StartProd_{sr}$	$\forall sr \in SR$	(date:hour)	End of production in supply routing sr
$RelativeDurationInOperation_{op,ut}$	$\forall u \in SP \cup M, op \in OP_u, t \in T$		Fraction of supply operation processed in this unit period compared to total operation duration
$RelativeDuration_{sr,t}$	$\forall sr \in SR, t \in$		
Goal function parameters			
$CostPerPercentDeviation_{fg}$	$\forall fg \in FG$	($/%)	The penalty for not reaching the planned service level.
$CostPerPercentDeviation_{fgp}$	$\forall fgp \in FGP$	($/%)	The penalty for not reaching the planned service level.
finite			1 if the planning is made under finite capacity; 0 if the planning is made under infinite capacity
IsEarlySupplyAllowed			1 if it is possible to supply more than demanded in the current period to have extra buffer for future demand; 0 otherwise;

WeightRoutingPreference	The reward factor for preference bonuses when choosing the preferred supply routings in the planning.
WeightPriority	With this parameter a bonus can be set on each 'priority-point' of the fulfilled demand.
WeigthInventoryTarget	Penalty for undershooting the inventory target
WeightInventoryCost	Weight of the total inventory costs in the planning horizon.
WeightDirectCost	Weight of the direct cost of supply
WeightCustomerSatisfaction	Reward for sales demand quantity fulfilled in view of customer satisfaction
WeightFulfillmentGoal	Penalty for costs of not reaching a fulfillment goal.
WeightFulfillmentGoalProduct	Penalty for costs of not reaching a product fulfillment goal.
WeightSalesTarget	Penalty for not reaching or exceeding a sales target
WeightSalesTargetProduct	Penalty for not reaching or exceeding a sales target product
WeightMinimumCapacity	Penalty on the quantity by which the minimum level of a unit or a product in a unit is not reached
WeigthFiniteCapacity	Penalty for amount by which a unit is overloaded when planned under finite capacity
WeightInfiniteCapacity	Penalty for amount by which a unit is overloaded when planned under infinite capacity

Table A.2: Table of parameters

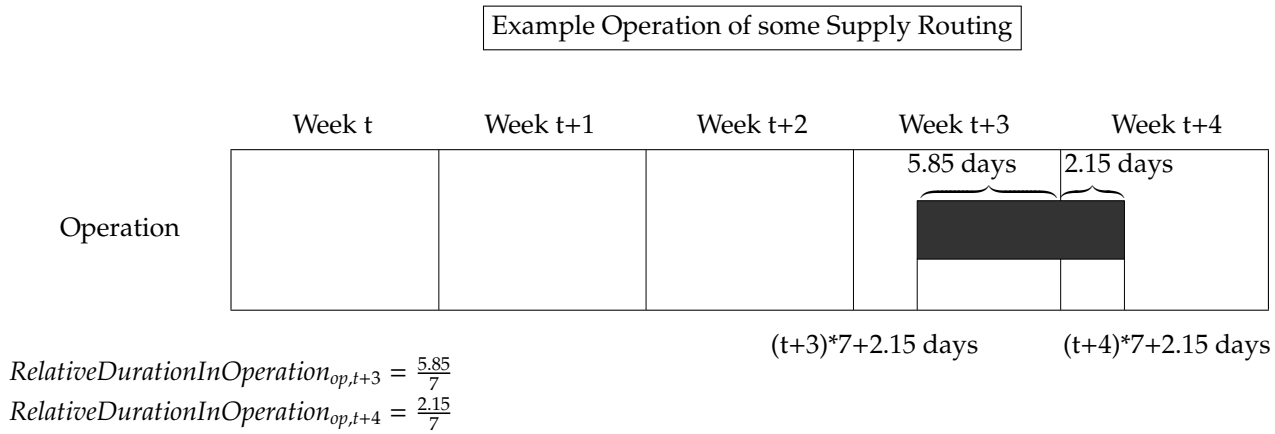

Figure A.2: RelativeDuration example

Two main concepts which can have some explanation are the parameters for the time window of supply routings and the parameters on the cost of operations. Furthermore some minor topics on parameters are mentioned.

A.3.1 Time window of Supply Routings

First two figures are presented in which the concepts can already be seen a bit: figure A.2 in which one operation on a machine is displayed and the concept relative duration is shown and figure A.3 which shows a supply routing with multiple steps.
Figure A.3 is for a supply routing defined on a controlled unit. Only then are the supply steps and their cycle times interesting as they have to be planned on machines owned by the producer which have a certain capacity. For supply routings defined on a subcontractor it is not interesting which steps in that routing are actually taken. Then it suffices to specify one default supply step and a leadtime to indicate how much time it takes for the inputs to be processed to the outputs of the supply routing. In this figure the cycle time is the amount of time needed to process inputs to outputs of a supply step. The process time is the time it takes to actually process all input quantity. It will be shown after the formal relations below that the process time is not dependent of the quantity to be processed and is a parameter instead of a variable. It is clear that for supply routings in own plants the leadtime is the sum of cycle times: $Leadtime_{sr} = \sum_{ss \in SS_{sr}}(CycleTime_{ss})$. After all, that is the time needed for the inputs to go through all supply steps.

The next relations are formally used to calculate the time window for supply routings:

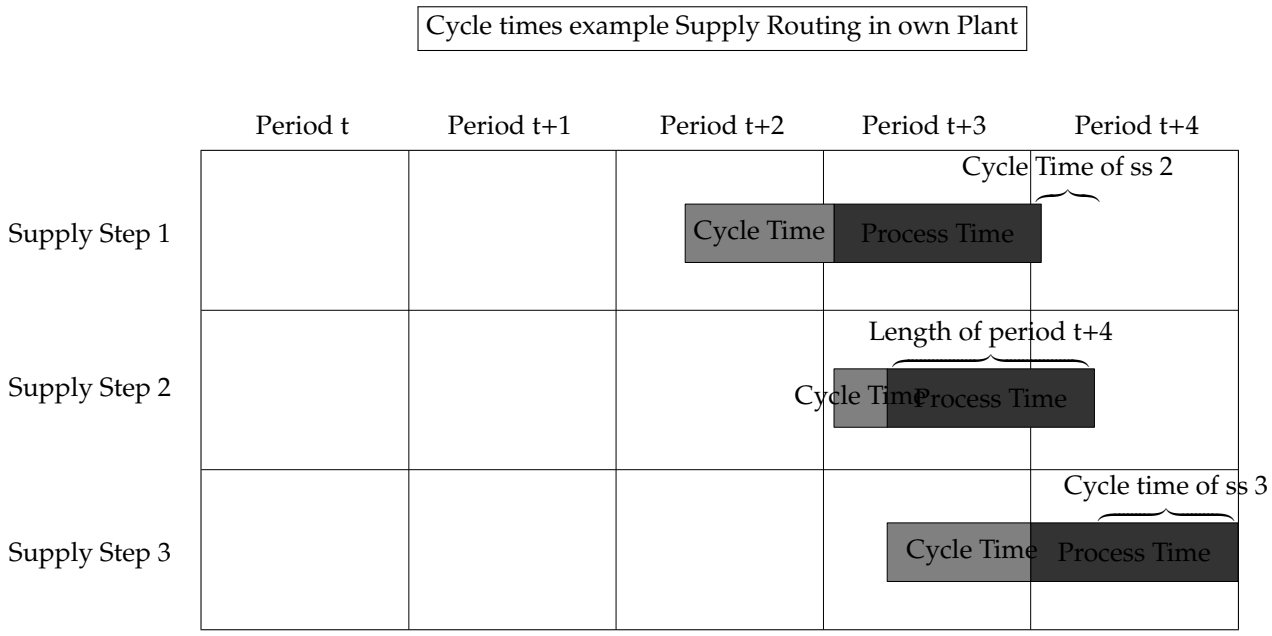

Figure A.3: Cycle times example

$$End_{sr} = End_t$$
$$\forall t \in T, pisp \in PISP, sr \in SR_{NS_{(pisp,t)}} \quad \text{(A.12)}$$

$$End_{ss} = End_{ss2} - CycleTime_{ss2} \quad \text{if } SequenceNr_{ss2} = SequenceNr_{ss} + 1 < \max_{ss2 \in SS_{sr}} (SequenceNr_{ss2}) - 1$$
$$= End_{sr} \quad \text{if } SequenceNr_{ss} = \max_{ss2 \in SS_{sr}} (SequenceNr_{ss2})$$
$$\forall ss, ss2 \in SS_{sr} \quad \text{(A.13)}$$

$$End_{so} = End_{ss}$$
$$\forall ss \in SS, so \in SO_{ss} \quad \text{(A.14)}$$

$$End_{op} = End_{so}$$
$$\forall u \in SC \cup M, op = (so, u) \in OP_u \quad \text{(A.15)}$$

$$Start_{op} = End_{op} - (End_t - Start_t + Leadtime_{sr})$$
$$\forall u \in SC \cup M, pisp \in PISP, t \in T, ns \in NS_{pisp,t}, sr \in SR_{ns}, ss \in SS_{sr}, so \in SO_{ss}, op = (so, u) \in OP_u \quad \text{(A.16)}$$

$$Start_{so} = \min_{u \in SC \cup M, op \in OP_u} (Start_{op})$$
$$\forall so \in SO, op = (so, u) \quad \text{(A.17)}$$

$$Start_{ss} = \min_{so \in SO_{ss}} (Start_{so})$$
$$\forall ss \in SS \quad \text{(A.18)}$$

$$Start_{sr} = Start_{ss}$$
$$\forall ss \in SS | SequenceNr_{ss} = 1 \quad \text{(A.19)}$$

$$RelativeDurationInOperation_{op,(u,t)} = \frac{(min\{End_{op}, Start_t\} - max\{Start_{op}, Start_t\})}{End_{op} - Start_{op}}$$
$$\forall u \in M \cup SC, op \in OP_u, t \in T \quad \text{(A.20)}$$

$$EndProd_{sr} = End_{sr} - CycleTime_{ss}$$
$$\forall sr \in SR, \{ss \in SS_{sr} | SequenceNr_{ss} = 1\} \quad \text{(A.21)}$$

$$StartProd_{sr} = Start_{sr} - CycleTime_{ss}$$
$$\forall sr \in SR, \{ss \in SS_{sr} | SequenceNr_{ss} = 1\} \quad \text{(A.22)}$$

$$RelativeDuration_{sr,t} = \frac{\min\{End_t, EndProd_{sr}\} - \max\{Start_t, StartProd_{sr}\}}{EndProd_{sr} - StartProd_{sr}}$$
$$\forall sr \in SR, t \in T \quad \text{(A.23)}$$

The above relations are defining the time span of a supply routing. So this are the relations by which is determined in which period the inputs are demanded for a supply in a certain period t. Initially the end of a period determines the end of a supply routing which supplies in that period, (A.12), and thereby the end of the last supply step. The earlier supply steps should all finish the cycle time of the next supply step before it's end, such that a product processed by the earlier supply step can also be processed by the next supply step before it's end. This is expressed by constraint (A.13). This can also be seen in figure A.3. The supply steps end a cycle time before the next supply step ends. Then each supply operation ends at the end of a supply step and each operation ends at the end of a supply operation.

By the duration of the operation a link is obtained with the start times in the supply routing. This can be seen in constraint (A.16). The duration of an operation is equal to the duration of the period for which the supply routing supplies when the unit on which the supply routing is defined is a machine. This is a kind of assumption. An operation can in reality of course be processed only during a part of a period, but for the macro planning it is assumed that the process is going on during the whole period. An operation should also never take longer than a period, because some of the output would already be finished in an earlier period, which means it would belong to a supply routing serving an earlier pispip. Intuitively it would be logical if the quantity of an operation would influence it's duration - and these times are variables - but in this interpretation the quantity influences the number of hours capacity needed during this duration of one period. If the supply routing is defined on a subcontractor also the leadtime of the supply routing is added to the duration. The reason is that for subcontractors the leadtime indicates how early

demand already should take place before the process can start. This is equivalent to the cycle times for a machine. For machines this is taken into account in (A.13) and for subcontractors with only one supply step it is arranged at this moment.

From the start of the operations, we get the start of a supply operation by taking the minimum start time of it's operations, (A.17). Then the start of a supply step is the minimum of the start of it's supply operations, (A.18), and the start of a supply routing is the start of the first supply step, (A.17).

From these start and endtimes it is possible to derive measures for relative duration. A relative duration of an operation in a time unit is called *RelativeDurationInOperation*. This variable is used to calculate the quantity which is processed in a unit period. It can be seen in figure A.2 The variable $RelativeDuration_{sr,t}$ is used to determine in which period dependent demand should take place. Dependent demand is caused in the first supply step and it starts already before the actual start of the supply step, as the inputs have to be available a cycle time of the first supply step in advance. However, it's definition in (A.23) does not use the end of the first supply step, but the end of the whole supply routing instead. This seems to be wrong and in a next version of the Macro Planner this is defined otherwise. One would expect that (A.21) used End_{ss} instead of End_{sr}.

A.3.2 Cost of Operations

Then there are the costs of an operation. To determine them in the goal function a parameter for the cost per quantity is given here.

$$CostPerQuantity_{op} = CostPerQuantity_{sr}$$
$$\forall u \in SC, op \in OP_u, so \in SO_{ss}, ss \in SS_{sr}, sr \in SR_{sc} \tag{A.24}$$

$$CostPerQuantity_{op} = \sum_{\{t \in T | Start_{op} < End_t \& Start_t < End_{op}\}} (CostPerHour_{ut} / RequiredCapacity_{op} / Yield_{ss}$$
$$+ CostPerTon_{ut}) / |t \in T | Start_{op} < End_t \& Start_t < End_{op}|$$
$$\forall u \in M, op \in OP_u, so \in SO_{ss} \tag{A.25}$$

The first equation is used for supply routings of a subcontractor and the second equation is for own controlled supply routings. For a subcontractor there is just a cost per ton which she should process, while the cost of an own controlled supply routing is depending on both quantity and required time. In the knowledge base each machine has a cost per hour and a cost per ton. This is propagated to the discrete unit-periods. The costs per hour should not be mistaken for fixed costs, as these costs are for the time the machine is really busy. The cost of an operation on a machine in a unit period per quantity is now composed of a cost per quantity and a cost per time needed to process the quantity. From these costs the cost of an operation on a machine can be obtained. Now this is done in constraint (A.25) by taking an average over the periods in which the operation is processed. However, in my opinion instead a weighted average by *RelativeDurationInOperation* would be the right choice to get the cost per quantity of an operation. The costs per quantity in a period when most of the operation is processed should be count more. In the goal function that are the parameters used to really calculate costs of a supply routing on a machine. There it will be multiplied by the incoming quantity of the operation, which means that using the term $Yield_{ss}$ also here is wrong.

A.3.3 Other parameters

The yield of the next supply steps in the supply routing, excluding step ss, $CombinedYield_{ss}$ is dependent of the combined yield and the yield itself of the subsequent supply step

Supply Step	Yield	Combined Yield
1	0.9	0.66
2	0.6	1.1
3	1.1	1.0

Table A.3: Example yield and combined yield

in the supply routing: it is the product of these two. When a supply step is the last step in the supply routing, such that its sequence number equals the total number of supply steps, there are no next supply steps and the combined yield is 1 by definition. This means that this parameter can be obtained by a recursive formula:

$$CombinedYield_{ss} = \begin{cases} CombinedYield_{ss2} * Yield_{ss2} & \text{if } SequenceNr_{ss} < |SS_{sr}| \\ 1 & \text{otherwise} \end{cases}$$
$$\forall sr \in SR, ss \in SS_{sr}, ss2 \in \{SS_{sr} | SequenceNr_{ss2} = SequenceNr_{ss} + 1\}$$

These parameters are used to calculate the quantities of operations based on the quantity of the supply routing, for example in constraint (A.31) and to calculate the input quantity of a supply operation based on the output quantity as for example in capacity constraint (A.39). To explain this important concept further, an example with numbers for a possible supply routing with three supply steps is given in table A.3. The yields are parameters from which the combined yield parameters are derived.

From these parameters it is also possible to deduct the total yield of the supply routing as

$$Yield_{sr} = Yield_{ss} * CombinedYield_{ss} \qquad \forall sr \in SR, \{ss \in SS_{sr} | SequenceNr_{ss} = 1\}$$

or

$$Yield_{sr} = \prod_{ss \in SS_{sr}} (Yield_{ss}) \qquad \forall sr \in SR$$

Another time-related parameter is $Start_{sd}$, which is used in determining which sales demands are taken into account by for example a product fulfillment goal. It holds that the start of a salesdemand is equal to the start of the period from the pispip to which it is related:

$$Start_{sd} = Start_t \qquad \forall sd \in SD_{p,sp,t}$$

Per pispip there is always one inventory demand. In other words, if $DinPISPIP_{d,pispip}$ for a pispip is summed over $d \in Id$ the answer is always 1. The amount demand is equal to the target of the pispip at the end of the period. This is intuitive when it is a target. But the most important is to realize that it is a target for the end of the period.

$$Quantity_{id} = TargetInventoryLevel_{pispip} \qquad \forall (id, pispip) \in PISPIP \times Id_{pispip}$$

At last some words about the interpretation and relevance of some parameters. The existence of a minimum (product) capacity for a subcontractor is to model the contracts with subcontractors as good as possible. A reason that it is possible to define different inventory item prices per pispip could for example be that in a product life cycle the expectation for the revenue per sold item is declining. An interpretation of the preference bonus for supply routings is maybe that for some reason a discount linear related to the quantity of products on a supply routing is expected in the future. As for the case that the price of a contract for a next period with an external supplier is dependent on the number of products which have used the supply routing in this planning horizon.

A.4 Variables

In table A.4 all the variables used in the mathematical model are listed. Important are the variables representing the quantity of a supply routing, Quantity_{sr}. From these variables for example Quantity_{ns} of new supplies are derived with equality constraint A.26, and the variables Quantity_{so} for supply operations are derived by the equalities A.33 and A.32. Also completely independent are Quantity_{op}, the variables for the operations planned on a machine. This is explained in the section on the constraints. At the demand side are the variables $\text{FulfilledQuantity}_{d}$ with $d \in Sd \cup Id$ independent such that the optimizer can choose to which demand to allocate the supplies of a pispip.

All other variables which are mostly used as measures for goals in the goal function can be derived from these demand and supply routing variables. It is possible to substitute them by the above mentioned variables. This gives the conclusion that the model size is in the order of the number of supply routings plus the number of demands. And the number of demands is equal to the number of sales demands and the number of inventory demands (=number of pispip's). So the model of the Macro Planner is in the order of nPeriods x nProduct Routings + nPeriods x nProductRoutings x nSupplyOperationsPerRouting x nMachines + nSalesDemands + nPeriods x nStockingPoints x nProducts, which is in the order of nPeriods x nProductRoutings x nSupplyOperationsPerRouting x nMachines + nSalesDemands + nPeriods x nStockingPoints x nProducts.

Also remarked in the section on constraints is that a variable for overloading the maximum product capacity of a unit period would be expected in the list of variables, but it is not.

Variables in the Planning Model			
Variable	Domain	Unit of Measure	Definition
Supply routings and Supplies			
Quantity_{sr}	$\forall sr \in SR$	(ton)	The output quantity of supply routing sr
Quantity_{so}	$\forall so \in SO$	(ton)	Quantity which results from the supply operation.
Quantity_{ns}	$\forall ns \in NS$	(ton)	The quantity of a new supply supply
Quantity_{op}	$op \in OP$	(ton)	The quantity of a supply routing processed on a machine
Demands			
$\text{FulfilledQuantity}_{d}$	$\forall d \in D$	(ton)	Quantity supplied to fulfill demand of d.
Derived variables			
$\text{MinimumQuantityNotMet}_{ut}$	$\forall ut \in SC \times T$	(ton)	Amount by which less quantity in a unit period is processed than the minimum capacity.
$\text{MaximumCapacityOverloaded}_{ut}$	$\forall ut \in SC \times T$	(ton)	Amount by which more quantity is planned on a unit than the available capacity.
$\text{MinimumProductQuantityNotMet}_{ut,p}$	$\forall (ut, p) \in UTP$	(ton)	Amount by which less quantity of a product is planned on a unit than the minimum product capacity.
$\text{MaximumTimeOverloaded}_{ut}$	$\forall ut \in M \times T$	(hour)	Number of hours by which more time is planned on a unit period than available.
$\text{UnfulfilledPercentage}_{fg}$	$\forall fg \in FG$	(%)	Percentage by which the fulfillment goal is not reached
$\text{UnfulfilledPercentage}_{fgp}$	$\forall fgp \in FGP$	(%)	Percentage by which the product fulfillment goal is not reached
$\text{OverFulfilledQuantity}_{st}$	$\forall st \in ST$	(ton)	Quantity by which the maximum of st is outnumbered.
$\text{UnfulfilledQuantity}_{st}$	$\forall st \in ST$	(ton)	Quantity by which the minimum of st is not reached.

$\text{OverFulfilledQuantity}_{stp}$	$\forall stp \in STP$	(ton)	Quantity by which the maximum of stp is outnumbered.
$\text{UnfulfilledQuantity}_{stp}$	$\forall stp \in STP$	(ton)	Quantity by which the minimum of stp is not reached.
$\text{CarriedForwardInventory}_{pispip}$	$\forall pispip \in PISPIP$	(ton)	Total inventory of a product which is taken to the next period in that stocking point.
$\text{SPCarriedForwardInventory}_{spip}$	$\forall spip \in SPIP$	(ton)	Total inventory which is taken to the next period in that stocking point.
$\text{UnallocatedSupply}_{pispip}$	$\forall pispip \in PISPIP$	(ton)	The inventory of a product which is taken forward to the next period of the stocking point besides the fulfilled inventory demand to reach the inventory target for that next period

Table A.4: Table of variables

A.5 Constraints

The constraints listed in the model serve several purposes and there are hard and soft constraints. In this section all constraints are listed and an explanation is given. First we start with the constraints which hold everything together, the constraints which control the in and outflow at the stocking points in each period:

$$Quantity_{ns} = (OutputFactor_{pisp,sr} * Quantity_{sr})$$
$$\forall pispip = (pisp, p) \in PISPIP, s \in NS_{pispip}, sr \in SR_{ns} \cap SRO_{pisp} \quad (A.26)$$

$$(1 - IsEarlySupplyAllowed) \sum_{ns \in NS_{pispip}} (Quantity_{ns}) \leq \sum_{d \in D_{pispip}} (Fulfilled Quantity_{d})$$
$$\forall pispip \in PISPIP \quad (A.27)$$

$$UnallocatedSupply_{pispip} = \sum_{ns \in NS_{pispip}} (Quantity_{ns}) + \sum_{is \in IS_{pispip}} (Quantity_{is})$$
$$+ CarriedForwardInventory_{(p,sp,t-1)} - \sum_{d \in D_{pispip}} (Fulfilled Quantity_{d})$$
$$\forall pispip = (p, sp, t) \in PISPIP \quad (A.28)$$

$$CarriedForwardInventory_{pispip} = UnallocatedSupply_{pispip} + (Fulfilled Quantity_{id \in Id_{pispip}})$$
$$\forall pispip \in PISPIP \quad (A.29)$$

$$SPCarriedForwardInventory_{spip} = \sum_{p \in P} (CarriedForwardInventory_{(p,spip)})$$
$$\forall spip \in SPIP \quad (A.30)$$

The quantity of a new supply, which is related to a pispip, is the total output quantity of it's supply routing times the factor of output going to the product in stocking point of the pispip to which ns is related. See constraint (A.26). When IsEarlySupplyAllowed is false, the sum of these new supplies to a pispip should not exceed the total demand which is fulfilled in that period by products from that stocking point, see (A.27). By the inventory targets and inventory demands it is of course still possible to raise the inventory from period to period, but it is not allowed to have already supplies for inventory, dependent or sales demand which only takes place in subsequent periods. This constraint can be switched on or off by specifying the IsEarlySupplyAllowed parameter of the optimizer to true. These constraints are concerning supply. The constraint (A.28) is to define something called unallocated supply as the quantity which is left after subtracting the fulfilled demands from the incoming quantity in the pispip. In the later constraint (A.46) the fulfilled demand will be restricted to be at most the demand which is faced in the pispip. The unallocated supply concerns inventory as the amount which is carried forward of a product at the end of a period is the sum of the fulfilled inventory demand, to reach the inventory target in the next period, and the unallocated supply, see (A.29). The total amount of products carried forward for a stocking point is the sum of the amount carried forward per product over the products.

$$\sum_{op \in OP_{so}} (Quantity_{op}) = \frac{1}{CombinedYield_{ss}} \times Quantity_{sr}$$
$$\forall sr \in SR, so \in SO_{ss}, ss \in SSsr \tag{A.31}$$

$$Quantity_{so} = Quantity_{ss}$$
$$\forall ss \in SS, so \in SO_{ss} \tag{A.32}$$

$$Quantity_{ss} = \frac{1}{CombinedYield_{ss}} \times Quantity_{sr}$$
$$\forall sr \in SR, ss \in SS_{sr} \tag{A.33}$$

These constraints determine from the total output quantity of a supply routing the output quantities for the separate operations, supply operations and supply steps. The relation with the supply steps is in constraint (A.33) arranged by the parameter $\text{CombinedYield}_{\text{ss}}$, a parameter which is explained in the parameter section. In (A.32) the quantity of a supply operation is put equal to the quantity of it's supply step as these operations are carried out simultaneously. The sum of the quantities of operations on machines has to be equal to the quantity of their supply operation by constraint (A.31). It should be noted that this gives freedom to choose on what machine what quantity is processed. Also important to hold everything together is the constraint for fulfilled quantity:

$$Fulfilled Quantity_{dd} = \sum_{sr \in SRI_{pisp}} (RelativeDuration_{sr,t}) \times \frac{Quantity_{sr}}{Yield_{sr}} \times InputFactor_{pisp,sr}$$
$$\forall dd \in Dd_{pisp,t}, (pisp, t) \in PISPIP \tag{A.34}$$

Supply routings to obtain supplies result in dependent demands at the input pisp's of the supply routing, unless there are no input pisp's in case it is a supply routing for raw material. Then no dependent demand is made. These dependent demands are divided over periods by the relative duration of the first supply step of a supply routing in these periods. After all, the total input quantity is already needed in the first supply step. In the parameter section some doubts were expressed about the current way of calculating this. When a planning is automatically generated, all dependent demand has to be fulfilled as otherwise next supply routings can not be fulfilled and are useless. Therefore the variable $\text{FulfilledQuantity}_{\text{dd}}$ immediately takes the quantity of dependent demand. This means by constraint (A.28) and also the requirement that the $\text{UnallocatedSupply}_{\text{pispip}}$ should be higher or equal to zero that for the pispip's that serve as input for supply routings supply has to be created which has to be done via supply routings. This gives dependent demand at other inputs for which supply has to be created and so on. That means that sales demand is only fulfilled when the supplies can be fulfilled from raw materials to end products. Now the important quantities are defined it is possible to check the constraints for the capacities in the supply chain:

$$MinimumQuantityNotMet_{ut} + \sum_{op \in OP_{sc}} (RelativeDurationInOperation_{op,sc,t} \times Quantity_{op}) \geq MinCapacity_{ut} \quad \forall ut = (sc,t) \in UT, sc \in SC \tag{A.35}$$

$$\sum_{op \in OP_{sc}} (RelativeDurationInOperation_{op,(sc,t)} \times Quantity_{op}) - MaximumCapacityOverloaded_{ut} \leq TotalAvailableCapacity_{ut} \quad \forall ut = (sc,t) \in UT, sc \in SC \tag{A.36}$$

These constraints are soft constraints for the capacity which is available in periods at units whose capacity is quantity-based (subcontractors). For each unit the number of products which are processed in a period is the sum of the relative duration of the operation in the unit period times the total number of products which should be delivered as output by the operation over all the operations executed. If this is less than the minimum capacity of a unit period the $\text{MinimumQuantityNotMet}_{\text{ut}}$ variable for which there is a penalty in the objective function becomes positive. If it is higher than the available capacity of a unit period $\text{MaximumCapacityOverloaded}_{\text{ut}}$ on which there is a (different) penalty in the goal function should be positive.

$$MinimumProductQuantityNotMet_{ut,p} + \sum_{op \in OP_{sc}} (RelativeDurationInOperation_{op,ut} \times Quantity_{so}) \geq MinProductCapacity_{ut,p} \quad \forall ((sc,t),p) \in UTP \tag{A.37}$$

$$\sum_{op \in OP_{sc}} (RelativeDurationInOperation_{op,ut} \times Quantity_{so}) \leq MaximumProductCapacity_{ut,p} \quad \forall ((sc,t),p) \in UTP \tag{A.38}$$

The first constraint works the same as (A.35), but now it is formulated for the specified minimum capacity per product in a unit period, and only over the supply operations which produce this product should be summed. The second constraint involves the maximum capacity per product in a unit period. However, this constraint is very different from (A.36) in the sense that it is a hard constraint. This choice seems strange as it prevents 'planning under infinite capacity' from really planning under infinite capacity. The alternative is of course to turn this constraint into a soft constraint and introducing a variable $\text{MaximumProductCapacityOverloaded}_{\text{ut,p}}$.

$$\sum_{op \in OP_{m}} (RelativeDurationInOperation_{op,t} \times Quantity_{op}/RequiredCapacity_{op}/Yield_{ss}) - MaximumTimeOverloaded_{ut} \leq TotalAvailableTime_{ut} \quad \forall (m,t) \in UT, m \in M \tag{A.39}$$

This is the unit capacity constraint for the machines (or time-based unit in general) for which there is a maximum capacity defined measured in time units. As the time a process takes is defined by the total input quantity divided by the $\text{RequiredCapacity}_{\text{op}}$, first the input quantity has to be determined by dividing the output quantity by the yield. Then it should be multiplied by the fraction of the operation taking place in the period and then the time can be calculated.

Of course there is also a maximum capacity at the stocking points. This is not arranged by soft constraints, but just simply as a hard constraint, see (A.59) and

(A.58). And then there are also the soft constraints related to the business goals of customer satisfaction and productivity.

$$UnfulfilledQuantity_{st} \geq MinimumQuantity_{st} - \sum_{sd \in SD_{sals}} (FulfilledQuantity_{sd})$$
$$\forall sals \in SalS, st \in ST_{sals} \tag{A.40}$$

$$OverFulfilledQuantity_{st} \geq \sum_{sd \in SD_{sals}} (FulfilledQuantity_{sd}) - MaximumQuantity_{st}$$
$$\forall sals \in SalS, st \in ST_{sals} \tag{A.41}$$

$$UnfulfilledQuantity_{stp} \geq MinimumQuantity_{stp}$$
$$- \sum_{sd \in SD_{sals} | Start_{sd} \leq Start_{stp} < End_{sd} \text{ and } \sum_{p \in P}(STPonP_{stp,p} \times DinPISPIP_{sd,p,sp,t}=1)} (FulfilledQuantity_{sd})$$
$$\forall sals \in SalS, stp \in STP_{sals} \tag{A.42}$$

$$OverFulfilledQuantity_{stp} + MaximumQuantity_{stp} \geq$$
$$\sum_{sd \in SD_{sals} | Start_{sd} \leq Start_{stp} < End_{sd} \text{ and } \sum_{p \in P}(FGPonP_{fgp,p} \times DinPISPIP_{sd,p,sp,t}=1)} (FulfilledQuantity_{sd})$$
$$\forall sals \in SalS, stp \in STP_{sals} \tag{A.43}$$

$$UnfulfilledPercentage_{fg} \geq FulfillmentPercent_{fg} - (\sum_{sd \in SD_{sals}} (FulfilledQuantity_{sd}/Quantity_{sd}) \times 100)$$
$$\forall sals \in SalS, fg \in FG_{sals} \tag{A.44}$$

$$UnfulfilledPercentage_{fgp} \geq FulfillmentPercent_{fgp} -$$
$$(\sum_{sd \in SD_{sals} | Start_{sd} \leq Start_{fgp} < End_{sd} \text{ and } \sum_{p \in P}(FGPonP_{fgp,p} \times DinPISPIP_{sd,p,sp,t}=1)} (FulfilledQuantity_{sd}/Quantity_{sd}) \times 100)$$
$$\forall sals \in SalS, fgp \in FGP_{sals} \tag{A.45}$$

When for a sales target the target minimum is not reached or the target maximum is exceeded, $UnfulfilledQuantity_{st}$ or $OverFulfilledQuantity_{st}$ are forced to become positive and this will give a penalty in the objective function. The same holds for the product sales targets. For a fulfillment goal (product) the soft constraint to determine the value for $UnfulfilledPercentage_{fg}$ or $UnfulfilledPercentage_{fgp}$. It will be positive when the percentage of fulfilled quantity of the concerned sales demands is lower than the fulfillment percentage that was specified in the fulfillment goal. It will get a penalty in the goal function. The relation between sales demand and sales target products/product fulfillment goals is set by taking the sales segment, the product and the start and end dates into account. All sales demands defined for a period which starts after the start of the FGP/STP and before the end of the FGP/STP are taken into account. However for sales targets and fulfillment goals there is currently an easily repairable bug as the start and end of the ST/FG is not taken into account.Then there are still many constraints which defines the domain for the variables:

$$0 \leq FulfilledQuantity_d \leq Quantity_d \quad \forall d \in Sd \cup Id \tag{A.46}$$
$$UnfulfilledPercentage_{fgp} \geq 0 \quad \forall fgp \in FGP \tag{A.47}$$
$$UnfulfilledPercentage_{fg} \geq 0 \quad \forall fg \in FG \tag{A.48}$$
$$UnfulfilledQuantity_{stp} \geq 0 \quad \forall stp \in STP \tag{A.49}$$
$$OverFulfilledQuantity_{stp} \geq 0 \quad \forall stp \in STP \tag{A.50}$$
$$UnfulfilledQuantity_{st} \geq 0 \quad \forall st \in ST \tag{A.51}$$
$$OverFulfilledQuantity_{st} \geq 0 \quad \forall st \in ST \tag{A.52}$$
$$MinimumProductQuantityNotMet_{ut,p} \geq 0 \quad \forall (ut,p) \in UTP \tag{A.53}$$
$$MaximumProductCapacityOverloaded_{ut,p} \geq 0 \quad \forall (ut,p) \in UTP \tag{A.54}$$
$$MinimumQuantityNotMet_{ut} \geq 0 \quad \forall ut \in UT \tag{A.55}$$
$$MaximumCapacityOverloaded_{ut} \geq 0 \quad \forall ut \in UT \tag{A.56}$$
$$MaximumTimeOverloaded_{ut} \geq 0 \quad \forall ut \in UT \tag{A.57}$$
$$0 \leq CarriedForwardInventory_{pispip} \leq MaxCapacity_{pispip} \quad \forall pispip \in PISPIP \tag{A.58}$$
$$0 \leq SPCarriedForwardInventory_{spip} \leq MaxCapacity_{spip} \quad \forall spip \in SPIP \tag{A.59}$$
$$UnallocatedSupply_{pispip} \geq 0 \quad \forall pispip \in PISPIP \tag{A.60}$$
$$Quantity_s \geq 0 \quad \forall s \in \{NS \cup IS\} \tag{A.61}$$
$$Quantity_{so} \geq 0 \quad \forall so \in SO \tag{A.62}$$
$$Quantity_{sr} \geq 0 \quad \forall sr \in SR \tag{A.63}$$
$$Quantity_{op} \geq 0 \quad \forall op \in OP_u, u \in M \cup SC \tag{A.64}$$

With all these constraints the solution space is defined. Now the objective function will indicate which of these candidate solutions will be the best for the optimizer.

A.6 Objective Function

The optimization used here is a multi-objective objective optimization as several performance measures for different business goals are taken into account and weighted in relation to each other. Therefore there are many parts in this objective function and they will be presented separately with explanation below. A distinction can be made in the terms of the goal function that represent real cost and have an influence on the business goals of profitability, and terms that measure other goals or set penalties on soft constraints. Note that the objective function is maximized. Furthermore remark that the goal function will give an optimal value z, but the most interesting information for the user about the solution is normally what it displayed by KPI's on the supply chain planning output.

$$z = -\sum_{(p,sp,t)\in PISPIP} (WeightInventoryCost * InventoryItemPrice_{p,sp,t} * YearlyInterestRate * \frac{End_t - Start_t}{YearDuration} * CarriedForwardInventory_{p,sp,t})$$

This term is to give a weight on the inventory cost over the year, by weight WeightInventoryCost. Notice that in accordance with the first assumption only *CarriedForwardInventory* is taken into account. After a planning is made a $\text{CostOfInventory}_{\text{p,sp,t}}$ is calculated declaratively by the application. This does take WIP inventory into account and will lead to a slightly different result. However, for the optimizer above formulation is also working and in production problems the assumption of paying inventory costs on the end inventory is common.

$$+\sum_{sd\in SD} (WeightRevenue \times FulfilledQuantity_{sd} \times RevenuePerQuantity_{sd})$$

This term focuses on the business goal of profitability: the KPI total revenue is measured. WeightRevenue is the weight for this term.

$$- \sum_{u \in M \cup SC, so \in SO_{ss}, op=(so,u)} (Quantity_{op} / Yield_{ss} * CostPerQuantity_{op} * WeightDirectCost)$$

This term in the goal function denotes the direct cost of supply which is KPI (3). These costs come from all the operations processed in the supply chain horizon. For each operation the quantity is adjusted to the incoming quantity as the cost per quantity is defined on that amount which is to be processed. Of course the yield of all operations by subcontractors is 1, such that there is actually just a cost on the output quantity. The structure of the parameter $CostPerQuantity_{op}$ of an operation was described in the section about parameters.

The other terms have not directly anything to do with costs, revenues or profits. First some terms in which penalties are assigned for violating soft constraints will be given.

$$- \sum_{ut \in UT} (MaximumCapacityOverloaded_{ut}) \times WeightFiniteCapacity * finite + WeightInfiniteCapacity * (1 - finite)$$

There is a penalty on the capacity overloaded in a unit period with quantity based capacity. This weight penalty should be different for optimizer settings which indicate planning with finite capacity and planning with infinite capacity. When finite is 1 the very high WeightFiniteCapacity is selected which would make it relatively unattractive to overload a subcontractor and when finite is 0 the low WeightInfiniteCapacity is selected. Note that the solution space of both versions is the same, only the values of the objective function is different for the solutions. It seems that defining one WeightCapacity per optimizer setting which is very high when that optimizer setting is for planning under finite capacity and low when it is for planning under infinite capacity would also have been enough.

$$- \sum_{ut \in SC \times T} (MinimumQuantityNotMet_{ut}) \times WeightMinimumCapacity$$
$$- \sum_{(ut,p) \in UTP} (MinimumProductQuantityNotMet_{ut,p}) \times WeightMinimumCapacity$$

Here a penalty on the under loading of a unit period is added, when less than expected is demanded from subcontractors. This weight is not affected by the choice for finite of infinite planning. Also a penalty on the under loading of a product for a unit period is added. As mentioned before the maximum product capacity is a hard constraint, so no penalty is set on the overloading of a product capacity in a unit period.

The other objective function terms indicate other non-cost-related goals or preferences.

$$- \sum_{st \in ST} (WeightSalesTarget \times (Unfulfilled Quantity_{st} + OverFulfilled Quantity_{st}))$$
$$- \sum_{spt \in STP} (WeightSalesTargetProduct \times (Unfulfilled Quantity_{stp} + OverFulfilled Quantity_{stp}))$$

Here a penalty is given when the minimum level of a sales target is not reached or the maxmimum level of a sales target is exceeded. Note that the penalty for these possibilities is equal. Also a penalty is given when the minimum level of a

product sales target is not reached or the maxmimum level of a product sales target is exceeded. The penalty for these possibilities is again equal.

$$- \sum_{fg \in FG} (WeightFulfillmentGoal \times CostPerPercentDeviation \times Unfulfilled Percentage_{fg})$$
$$- \sum_{fgp \in FGP} (WeightFulfillmentGoalProduct \times CostPerPercentDeviation \times Unfulfilled Percentage_{fgp})$$

Here penalties are given on the percentage by which the fulfillment goals are not reached and on the percentage by which the fulfillment product goals are not reached. Excessive realized fulfillments are obviously not penalized as $Unfulfilled Percentage$ cannot be negative.

$$+ \sum_{sd \in Sd} (WeightCustomerSatisfaction * FulfilledQuantity_{sd})$$

The word WeightCustomerSatisfaction is a bit misleading as here actually the KPI for the business goal of productivity, the total fulfilled demand, is measured and weighted.

$$+ \sum_{id \in Id} (WeightInventoryTarget * FulfilledQuantity_{id})$$

This term is specific for measuring how well the inventory target is reached. However it would maybe be more logical to weight the percentage of fulfillment of the inventory targets instead of the fulfilled quantity of inventory demand, as that suggestion is the KPI. Also intuitively this should be clear: it is less desirable to fulfill a certain quantity of inventory targets when the targets are much higher than when it is exactly equal to the targets.

$$+ \sum_{pispip \in PISPIP} (\sum_{sd \in Sd_{pispip}} (WeightPriority \times RevenuePerQuantity_{sd} \times FulfilledQuantity_{sd} \times (Priority_{sd} - WeightPriority)))$$

This is to promote fulfilling sales demands with higher priorities. However the exact interpretation of this term is unclear. It is strange that the weight of the priority is subtracted from the priority and actually this is a bug in the Macro Planner. An advantage of this is also that it could help to prevent multiple solutions: when revenue is equal and fulfillment goals and sales targets are reached already, deciding from which sales segment demand to fulfill is possible in many equally good ways. This is prevented when the priorities are unequal.

$$+ \sum_{sr} (WeightRoutingPreference \times Quantity_{sr} \times PreferenceBonus_{sr})$$

This constraint sums the total preference bonus of the supply routings in the planning weighted with WeightRoutingPreference.

Appendix B

Capacity Extension Model

The proposal of parameters and variables for capacity extension is straightforward as they have a similar structure of the costs. In tables B.1 and B.2 the parameters and variables are given respectively. Not only extra parameters are specified. When introducing the option of capacity extension it is indefensible to keep regarding the capacity constraints as soft constraints. These should be changed to hard constraints on the new capacity. Here it should be noted that if one wants to plan against infinite capacity it suffices to set the *WeightExtraMachine* and *WeightShiftCosts* to a very low number. Therefore the old $WeightFiniteCapacity$ and $WeightInfiniteCapacity$, the goal function terms in which they played a role, and the penalty variables used in these goal terms can be deleted.

In this model we will also use two variables that do exits in the current model but are not used in the optimizer, $TotalCapacityPerUnit_{ut}$ and $TotalAvailableCapacityPerUnit_{ut}$. They denote respectively the total number of hours in the period and the total number of hours the machine is operated according to the original shift pattern. For the newly introduced parameters the units of measure are not always natural. For example the parameter $\text{VariableCostBuyPlus}_{spip}$ is expressed in terms of ($/ton) for computational purposes. This means that in reality a price per squared meter has to be converted.

For the newly introduced variables also new constraints have to be defined. Most important is that capacities or number of operating hours will not turn out lower than 0. Furthermore the binary variables should be kept track off and the variables representing a balance depend on earlier values for buy and sell variables.

Parameters for Capacity extension

Parameter	Domain	Unit of Measure	Definition
$\text{CostPerUnitBought}_{u}$	$\forall m \in M$	($/unit)	The cost of buying an extra machine or the revenue of selling a machine.
$\text{FixedCostShiftChange}_{ut}$	$\forall ut \in M \times T$	($)	This represents fixed cost when a deviation from the intended shift pattern is made.
$\text{VariableCostShiftChange}_{ut}$	$\forall ut \in M \times T$	($/hour)	When a deviation is made this represents the cost per hour deviation.
$\text{FixedHireCost}_{spip}$	$\forall spip \in SPIP$	($)	When stocking point capacity is hired this represents fixed cost (for searching a suitable building for example).
$\text{FixedBuyCost}_{spip}$	$\forall spip \in SPIP$	($)	When stocking point capacity is bought or sold this represents fixed cost (for finding a seller or buyer for example).
$\text{VariableCostBuyPlus}_{spip}$	$\forall spip \in SPIP$	($/ton)	The price per amount of bought capacity.
$\text{VariableCostBuyMin}_{spip}$	$\forall spip \in SPIP$	($/ton)	The revenue per amount of sold capacity.
$\text{VariableCostHire}_{spip}$	$\forall spip \in SPIP$	($/ton)	The cost per amount of stocking point capacity hired.
WeightExtraHired			The factor with which the hire costs are weighted in the goal function of the optimizer.
WeightExtraBought			The factor with which the costs for buying and selling are weighted in the goal function of the optimizer.
WeightShiftCost			The factor with which the costs using another operational time for the machines than originally specified is weighted.
WeightExtraMachine			The factor with which the costs and revenue for buying and selling machines are weighted.

Table B.1: New capacity extension parameters

Variables for Capacity extension			
Variable	Domain	Unit of Measure	Definition
$\text{NewMaxCapacity}_{\text{spip}}$	∀spip ∈ SPIP	(ton)	After buying, selling and hiring capacity this is the new available amount.
$\text{NewMaxCapacity}_{\text{pispip}}$	∀pispip ∈ PISPIP	(ton)	This is the new available capacity per product in the spip.
$\text{CapacityBuyed}_{\text{spip}}$	∀spip ∈ SPIP	(ton)	Capacity bought for this stocking point in period.
$\text{CapacitySold}_{\text{spip}}$	∀spip ∈ SPIP	(ton)	Capacity sold for this stocking point in period.
$\text{CapacityHired}_{\text{spip}}$	∀spip ∈ SPIP	(ton)	Capacity hired for this stocking point in period.
$\text{SaldoUnits}_{\text{ut}}$	∀ut ∈ M × T	(units)	This represents how many machines u in time period t are available.
$\text{UnitsBought}_{\text{ut}}$	∀ut ∈ M × T	(units)	This gives how many machines are bought or sold in this period.
$\text{AvailableHours}_{\text{ut}}$	∀ut ∈ M × T	(hours)	The number of hours available in period t per machine u.
$\text{ExtraHoursPerUnit}_{\text{ut}}$	∀ut ∈ M × T	(hours)	The extra number of operating hours for machine u in period t.
$\text{IsCapacityBuyed}_{\text{spip}}$	∀spip ∈ SPIP		1 stocking point capacity is bought or sold in this period; 0 otherwise
$\text{IsCapacityHired}_{\text{spip}}$	∀spip ∈ SPIP		1 stocking point capacity is hired in this period; 0 otherwise
$\text{IsShiftChanged}_{\text{ut}}$	∀ut ∈ M × T		1 another operation time is used than specified by the shift pattern; 0 otherwise

Table B.2: New capacity extension variables

$$NewMaxCapacity_{(sp,t)} = MaxCapacity_{(sp,t)} + \sum_{s \leq t} (CapacityBuyed_{(sp,s)} - CapacitySold_{(sp,s)})$$
$$\forall (sp,t) \in SPIP$$

$$SaldoUnits_{(m,t)} = NumberOfUnits_{(m,t)} + \sum_{s \leq t} (UnitsBought_{m,s})$$
$$\forall ((m,t) \in M \times T)$$

$$AvailableHours_{ut} = TotalAvailableCapacityPerUnit_{ut} + ExtraHoursPerUnit_{ut}$$
$$\forall (ut) \in M \times T$$

$$ExtraHoursPerUnit_{ut} \leq IsShiftChanged_{ut} * TotalCapacityPerUnit_{ut}$$
$$\forall (ut) \in M \times T$$

$$UnitsBought_{ut} \leq IsNewNrOfUnits_{ut} * M$$
$$\forall (ut) \in M \times T$$

$$CapacityHired_{spip} \leq IsCapacityHired_{spip} * M$$
$$\forall spip \in SPIP$$
$$CapacityBought_{spip} \leq IsCapacityBought_{spip} * M$$
$$\forall spip \in SPIP$$

$$CapacitySold_{spip} \leq IsCapacityBought_{spip} * M$$
$$\forall spip \in SPIP$$

$$NewMaxCapacity_{pispip} = NewMaxCapacity_{spip} * \frac{MaxCapacity_{pispip}}{MaxCapacity_{spip}}$$
$$\forall pispip \in PISPIP$$

$$SaldoUnits_{ut} \in \mathbb{N}_{+} \qquad \forall ut \in M \times T$$

$$0 \leq AvailableHours_{ut} \leq TotalCapacityPerUnit_{ut} \qquad \forall ut \in M \times T$$

$$NewMaxCapacity_{spip} \geq 0 \qquad \forall spip \in SPIP$$

$$CapacitySold_{spip} \geq 0 \qquad \forall spip \in SPIP$$

$$CapacityBought_{spip} \geq 0 \qquad \forall spip \in SPIP$$

$$CapacityHired_{spip} \geq 0 \qquad \forall spip \in SPIP$$

$$IsShiftChanged_{ut} \in \{0,1\} \qquad \forall (ut) \in M \times T$$

$$IsCapacityHired_{spip} \in \{0,1\} \qquad \forall spip \in SPIP$$

$$IsCapacityBought_{spip} \in \{0,1\} \qquad \forall spip \in SPIP$$

In these constraints M is a very large number. Now the old unit capacity constraints of the type time and the stocking point constraints can be changed to:

$$\sum_{op \in OP_m} (RelativeDurationInOperation_{op,t} \times Quantity_{op} / RequiredCapacity_{op} / Yield_{ss}) \leq$$
$$SaldoUnits_{ut} * AvalableHours_{ut}$$
$$\forall (m,t) \in UT, m \in M$$

$$SPCarriedForwardInventory_{spip} \leq NewMaxCapacity_{spip}$$
$$\forall spip \in SPIP$$

$$CarriedForwardInventory_{pispip} \leq NewMaxCapacity_{pispip}$$
$$\forall pispip \in PISPIP$$

In the goal function the following terms can be added for capacity extension.

$$+ \sum_{spip \in SPIP} (CapacityBought_{spip} * VariableCostBuyPlus_{spip} + CapacitySold_{spip} * VariableCostBuyMin_{spip} +$$
$$IsCapacityBought_{spip} * FixedBuyCost_{spip}) * WeightExtraBought$$

$$+ \sum_{spip \in SPIP} (CapacityHired_{spip} * VariableCostHire_{spip} +$$
$$IsCapacityHired_{spip} * FixedHireCost_{spip}) * WeightExtraHired$$

$$+ \sum_{ut \in M \times T} (ExtraHoursPerUnit_{ut} * VariableCostShiftChange_{ut} +$$
$$IsShiftChanged_{ut} * FixedCostShiftChange_{ut}) * WeightShiftCost$$

$$+ \sum_{ut \in M \times T} (UnitsBought_{ut} * CostPerUnitBought_{u}) * WeightExtraMachine$$

This model can not be implemented easily in the Macro Planner because of the non-linearity in the constraints. Another point of interest is that it is to be expected for most value of the parameters that at the end of the planning horizon a lot of machines and stocking point capacity are sold. Obviously this leads to a bad starting

point for the periods after the planning horizon. One simple solution would be to ignore the results for the last periods of the planning horizon, when you are confident enough that the end-of-horizon effects do not influence the planning in earlier periods. Another solution would be to value the assets at the end of the planning horizon in the goal function. Anyhow this is an open issue.

Appendix C

Verification Details

Check 1 holds for the sales demo metal data with 3 forecasts scenarios which we introduced as starting point if we set the probability of LOW to 0.1, the probability of MEDIUM to 0.6 and the probability of HIGH to 0.3 if we set N to 3. We can use this data and these probabilities for the other checks as well. With holds is meant here that the KPI scoreboard displays the same values and the goal total when we print this in the server output is the same for using 1 optimizer setting, an option defined in the knowledge base, 'Setting 4: Finite with pre-production and fulfillment goal'. The weights in the goal function for this setting are: CustomerSatisfaction = 1, Direct Cost = 1, FiniteCapacity = 99999999, FulfillmentGoal = 100000, InfiniteCapacity = 0.01, InventoryCost = 1, InventoryTarget = 10, Revenue = 10 and the other weights are zero.

For check 2 we changed the knowledge of the sales demo metal data such that the quantities for LOW and HIGH are set equal to the quantities of MEDIUM. It turns out that all measures, expected goal function, worst case goal function and N reliability with N=1,2,3 yield the same result. This means that the KPI scoreboard displays the same values and the goal total when printed in the server output is equal. This is again only done for the before mentioned setting 4.

Check 3 is done on only forecast scenario MEDIUM of the sales demo metal data, which gets assigned a probability of 1, although this is not strictly necessary as 1 is the value the application would set if the probability is not equal to 1. The scrap price is still set on 0. Note that if the scrap price would be larger than 0 for this situation, the outcomes may be different. It means that due to the higher scrap price the trade-off for the optimizer between scrap price and bonuses by for example preferred product routings and the cost of producing may be made differently. In general even for a scrap price of 0 the Robust Macro Planner may theoretically indicate that it is advantageous to produce more than the demand and to scrap it. Then we feel that the outcome of the Robust Macro Planner fits better to the preferences of the customer. In other words this check could have failed without having an error in the robust algorithms, but if this check fails multiple times it is a strong indication there is. For our demo set indeed it turns out that planning in the Robust Macro Planner for all three options yield the same result as the normal Macro Planner gives.

For check 4 and 5 we used the same data set. We first have to mention that a trivial requirement for check 4 is of course that the cost of producing a finished product should be larger than 0, but this is normally true. This holds also for our changed version of the knowledge of our sales demo metal data. We deleted the fulfillment goal for Asia, the only fulfillment goal in the knowledge. We set the quantities of LOW alternating to the quantity of MEDIUM minus 100 and the quantity of MEDIUM. We set the quantities of HIGH alternating to the quantity of MEDIUM plus 100 and the quantity of MEDIUM. We have costs in every supply routing and therefore the trivial requirement is satisfied. We again applied optimizer

setting 4. Check 4 holds in the sense that the obtained minimum profit, minimum revenue, supply cost, inventory cost, total fixed cost, maximum demand fulfillment and minimum inventory fulfillment at the worst case robust outcome match with the profit, revenue, supply cost, inventory costs, total fixed costs, demand fulfillment and inventory fulfillment in the normal outcome for scenario low. Check 5 also holds.

For check 6 we used the numerical example from section 5.1. Indeed all the results turned out to be as expected.

Bibliography

[1] Maatschappelijk jaarverslag, 2007. Koninklijke Nederlandse Toeristenbond ANWB.

[2] Aimms language reference - stochastic programming, 2009.

[3] S. Ahmed. Convexity and decomposition of mean-risk stochastic programs. *Mathematical Programming*, 106(3):433–446, 2006.

[4] T. Assavapokee, M.J. Realff, J.C. Ammons, and I. Hong. Scenario relaxation algorithm for finite scenario-based min-max regret and min-max relative regret robust optimization. *Computers and Operations Research*, 35(6):2093–2102, 2008.

[5] P. Beraldi, R. Musmanno, and C. Triki. Solving stochastic linear programs with restricted recourse using interior point methods.

[6] P. Beraldi, R. Musmanno, C. Triki, and S. Zenios. Limited recourse in two-stage stochastic linear programs. *Journal of Information and Optimization*, 24(3):445–465, 2003.

[7] J.R. Birge and F.V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operations Research*, 34:384–392, 1988.

[8] G. Chen, M.S. Daskin, Z.J. Max Shen, and S. Uryasev. The α-reliable mean-excess regret model for stochastic facility location modeling. *Naval Research Logistics*, 53(7):617–626, 2006.

[9] Dupačová. Stochastic programming with incomplete information: A survey of results on postoptimzation and sensitivity analysis. *Optimization*, 18(4):507–532, 1987.

[10] Dupačová. Stability and sensitivity analysis for stochastic programming. *Annals of Operations Research*, 27(1):115–142, 1990.

[11] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3/4):197–206.

[12] M.S. Daskin. α-reliable p-minimax regret: A new model for strategic facility location modeling. *Location Science*, 5(4):227–246, 1997.

[13] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494, 2006.

[14] M.A.H Dempster and R.T. Thompson. Parallelization and aggregation of nested benders decomposition. *Annals of operations research*, 81(1):163–188, 1998.

[15] H.I. Gassmann and S.W. Wallace. Solving linear programs with multiple right-hand sides: Pricing and ordering schemes. *Annals of Operations Research*, 64(1):237–259, 1996.

[16] H. Haarman, E. Den Exter, J. Van der Schaar, and W. Van Heijst. Crisis dominates the supply chain agenda in 2009, March 2009.

[17] D. Haugland and S.W. Wallace. Solving many linear programs that differ only in the righthand side. *European Journal of Operations Research*, 37(3):318–324, 1988.

[18] P. Kall and S. W. Wallace. *Stochastic Programming*. John Wiley and Sons, 1994.

[19] Lapide and Suleski. Supply chain optimization: Just the facts. Technical report, Advanced Manufacturing Research, Inc., 1998.

[20] J.M. Latorre, S. Cerisola, A. Ramos, and R. Palacios. Analysis of stochastic problem decomposition algorithms in computational grids. *Annals of Operations Research*, 166(1):355–373, 2009.

[21] I.J. Lustig, J.M. Mulvey, and T.J. Carpenter. Formulating two-stage stochastic programs for interior point methods. *Operations Research*, 39(5):757–770, 1991.

[22] J.M. Mulvey and A. Ruszczyński. A new scenario decomposition method for large-scale stochastic optimization. *Operations Research*, 43(3):477–490, 1995.

[23] J.M. Mulvey, R.J. Vanderbei, and S.A. Zenios. Robust optimization of large-scale systems. *Operations research*, 43(2):264–281, 1995.

[24] J. et al. Padgett. A comparison of carbon calculators. *Environmental Impact Assessment Review*, 28:106–115, 2008.

[25] D. Paraskevopoulos, E. Karakitsos, and B. Rustem. Robust capacity planning under uncertainty. *Management Science*, pages 787–800, 1991.

[26] C.B. Patel. *A multi-objective stochastic approach to combinatorial technology space exploration*. PhD thesis, Georgia Institute of Technology, August 2009.

[27] D. Poole. *Linear algebra: a modern introduction*. Brooks/Cole Pub Co, 2005.

[28] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, 1995.

[29] A. Ruszczyński. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35(3):309–333, 1986.

[30] A. Ruszczyński. An augmented lagrangian decomposition method for block diagonal linear programming problems. *Operations Research Letters*, 8(5):287–294, 1989.

[31] A. Ruszczyński. Some advances in decomposition methods for stochastic linear programming. *Annals of Operations Research*, 85(1):153–172, 1997.

[32] A. Ruszczyński and A. Świętanowski. Accelerating the regularized decomposition method for two stage stochastic linear programs. *European Journal of Operations Research*, 101(2):328–342, 1997.

[33] I. Schwartzmans. Macro planner quintiq business analysis, May 2009.

[34] S. Sen and J. Higle. An introductory tutorial on stochastic linear programming models, 1999.

[35] S. Solak. *Efficient solution procedures for multistage stochastic formulations of two problem classes*. PhD thesis, Georgia Institute of Technology, December 2007.

[36] S. Trukhanov, L. Ntaimo, and AJ Schaefer. Adaptive multicut aggregation for two-stage stochastic linear programs with recourse. Technical report, Working Paper, Texas A&M University, Department of Industrial and Systems Engineering, 2008.

[37] R.M. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[38] H. Vladimirou and S. Zenios. Stochastic linear programs with restricted recourse. *European Journal of Operations Research*, 101(1):177–192, 1997.

[39] J.H. Wilkinson. *Rounding errors in algebraic processes*. Dover Pubns, 1994.