Eindhoven University of Technology

MASTER

Modeling and solving the production scheduling problem at AC

Coset, R.P.J.

Award date:
2011

Modeling and Solving the Production Scheduling Problem at AC

Master thesis

Version 1.2.160 (public)
May 23, 2011

Student:
R.P.J. Coset

Supervisors:
prof. dr. ir. W.P.M. Nuijten
ir. J.W.M. Welberg

Preface

This master thesis is the result of my master project concluding the master Business Information Systems at Eindhoven University of Technology (TU/e). The project has been performed in a collaboration between TU/e and AC. Several people have contributed to the results of this master project and I would like to take the opportunity to thank them.

First of all I would like to thank my graduation supervisor Wim Nuijten for giving me the opportunity to conduct my master's project at AC. Moreover, thank you for the constant guidance, all the advice, constructive feedback, and patience during this project! Second I would like to thank my supervisor at AC Jan-Willem Welberg, for all the provided information, clear advice, and valuable feedback during my project. Furthermore I would like to thank my colleagues from AC for their information and feedback provided during this project. Especially I would like to thank Marcel Wiegerinck and Theo Timmer for providing me insight in the technical production process of AC.

Furthermore I would like to thank my friends for their loyalty, help and all the fun during my study at the TU/e. Without them I would never had such a great time. Last but not least I would like thank my family and girlfriend for their support during my study at the TU/e and especially during the last couple of months of this project.

Roel Coset
Eindhoven, May 2011

Abstract

In larger businesses one is often faced with a rather complex production scheduling problem. This is due to an abundance of, often conflicting, goals and constraints about production, inventory and service levels. The lack of decent planning support by existing *ERP*, *SCM* and *MES* systems often results in manual or spreadsheet solutions.

AC[1], a worldwide leader in it's product range, finds itself in such a position. In the current market there is high demand for the many different products produced by AC. With limited production capacity and expensive changeovers this results in a complex production planning problem.

An optimization approach has been proposed to address the production planning problem of AC. This optimization approach consists of an intermediate mathematical model, containing all necessary required information about the production and planning process. This intermediate model, i.e., the AC model, is mapped to the generic model of *IBM ILOG Plant PowerOps* (*PPO*) using the plugin framework of *PPO*. The results of the optimization approach showed a significant improvement of product availability for sales. Considering the impact of the initial stock and the uncertainty about the quality of the final stock, this improvement might only be applicable for the investigated warm-up period. Whether this improvements holds when a steady-state is achieved is subject to future research.

---

[1] The name of the company is anonymized in this public version of the document, to protect confidential information.

Contents

Chapter 1

Introduction

In the current market a higher demand for the products, produced by AC, exists than the production capacity of AC can handle. To better serve the market, AC tries to improve its internal processes. Since a better production planning would make better use of the limited production capacity; one of these improvements considers the planning process. This graduation project is the initial step in a larger research project by TU/e and AC as a partner. The goal of this initial project is to investigate an automated planning solution, to support the current planning process of AC and to identify possible shortcomings of existing approaches.

This document is structured in several chapters. First, information about AC is presented in Chapter 2. This includes some background information about AC, what AC produces, the planning and scheduling process and a more detailed motivation for this project. A literature study has been performed on automated planning and scheduling optimization in the process industry. The results of this literature study are presented in Chapter 3. The literature identifies the complexity of planning and scheduling problems in the process industry and an approach using optimization algorithms is suggested.

Chapter 4 checks if the planning problem found at AC matches the complexity of the problems in the literature, followed by a proposed planning solution in Chapter 5.

An implementation of this proposed solution has been analyzed in Chapter 6, followed by an evaluation of this planning and scheduling optimization approach in Chapter 7, leading to the final conclusion in Chapter 8.

Chapter 2

AC

The following sections will describe the product AC produces (Section 2.1), the global organization structure of AC (Section 2.2), the planning and production process of the products of AC (Section 2.3), and the motivation of AC for this research project (Section 2.4).

2.1 The main product

The information about products, produced by AC is considered classified information and has been omitted from this version of the document.

2.1.1 Product types

AC offers many different products to their customers. One of these product groups is the most important, as this product is both an end product and an input to the other products.

The different product types are defined by a specific product name. Since this product name has an explicit meaning, it will be shortly explained. A very common product is *1000 1680 f1000*. The first number defines the type of the product, this type is defined by the composition of the product. The second number defines the *titer* of the product in *dtex*. *Titer* is a measure and *dtex* is a unit for this measurement. It defines the weight of the product in gram per 10 kilometers, which more or less defines the thickness of the product. The last number is the number of *filaments* in the product. The product consists of a lot of small threads, one of this small threads is called a *filament*.

Every product has a couple of different packaging sizes, resulting in many different products of the same product. Figure 2.1 shows how all these different parameters impact the number of variants of one product. The following different variations are possible:

- *GGL vs. OGL: GGL* is a Dutch abbreviation for *equal product length.* A spool of the type *GGL* is a normal spool with a certain standard weight. It might be the case that during the production a thread snaps, resulting in an unfinished spool. These unfinished spools are flagged as *OGL*, which is a Dutch abbreviation for *unequal product length.*
- *Spool weight:* The length and the *titer* of the spool define the weight of the spool. There are different weights possible per product. Every product has a default spool

weight. However, some customers or after-treatments might have specific weight demands.

- *Spool size:* There are two different spool types, each with its own size.
- *Packaging:* By default, spools are packed in large pallets of about 90 spools. However, AC also offers smaller packages. For instance boxes of 2 or 4 spools.

For many products, the customers do not have specific demands about the spool weight, spool size or the packaging size. This results in the aggregation of different variants into one aggregated product, any possible variant in the aggregated product satisfies the customer's needs. However, some customers or converting operations do have specific demands, i.e., a customer may have machinery that only operates with type of spools or a converting operation can be more efficient with spools of 5 kilo instead of the default spool weight of 10 kilo. This results in more different aggregated products of the same product.

2.2 Global Organization Structure

AC is mainly located in The Netherlands. Besides some overseas sales offices and warehouses, the main offices and production are located in The Netherlands and Germany. AC is located in four locations: *Arnhem*, *Emmen*, *Delfzijl*, and *Wuppertal*.

Production facilities are located in *Delfzijl*, *Arnhem*, and *Emmen*. The production facility in *Delfzijl* produces the polymer from the monomers. Two small production facilities are located in *Arnhem*; a small pilot production plant and the pulp production facility *FSQ*. The main production facility is located in *Emmen*. Two production facilities, *FDQ1* and *FDQ2* are located here. *FDQ1* produces product out of the polymer. *FDQ2* produces pulp, staple fibers, and chopped fibers, together with the converted products.

Two offices of AC are located in *Arnhem* and *Wuppertal*. The main office is located in *Arnhem*. In these two offices the different marketing, sales, finance, human resources, purchasing and logistics departments are located. Some departments are gathered in one location, while others are split among these offices. An example of this is the sales department. The sales department of consists of different sub-divisions, called *resorts* (Section 2.2.1). These *resorts* are divided among the two offices.

2.2.1 Resorts

The sales department of AC is divided in eight different sub-divisions, called resorts. Every *resort* specializes in specific customer markets. Like in many other organizations, there exists some sort of competition between these resorts.

The sales department consists of the following *resorts*:

- Ballistics (QMB)
- Composites (QMC)
- Mechanical rubber goods (QMR)
- Tires (QMX)
- Friction (QMF)
- Protective (QMP)

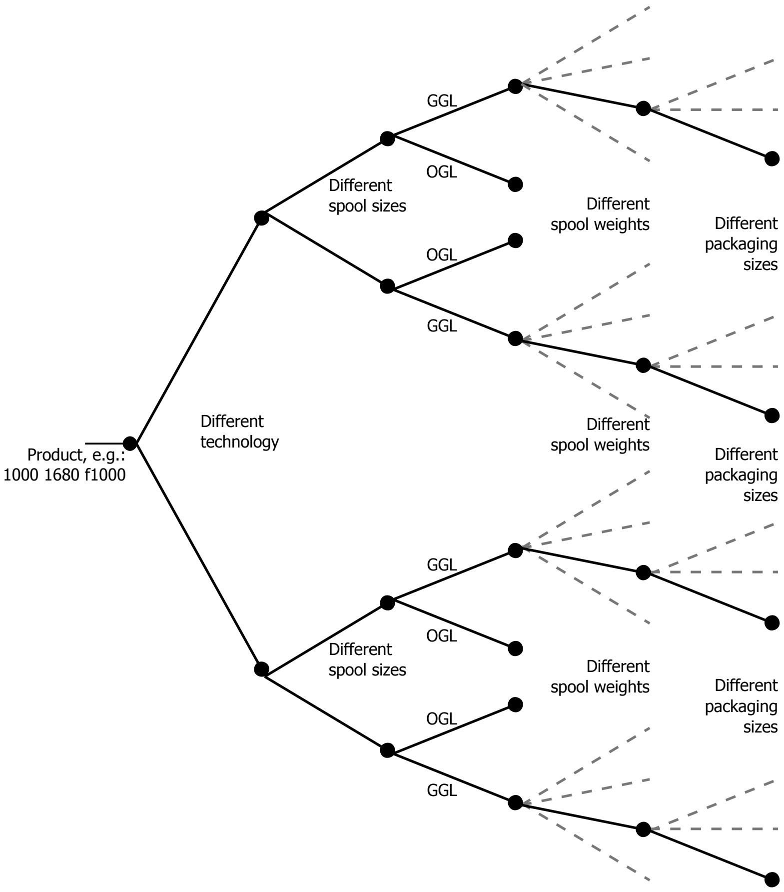

Figure 2.1: Tree, showing the possible different variants for one product.

- Optical fiber cables (QMO)
- Linear tension members (QMT)

2.3 Process description

AC is a large company, consisting of many different internal processes handling sales, transportation, etc. For the scope of this project, only the planning process (Section 2.3.1) and the production process (Section 2.3.2) are relevant.

All the processes that lead to finished products are dependent on the products produced by the spinning process and the spinning process suffers from limited capacity. This results in this project only covering the planning and production process of the spinning process in the flow, as it is considered the bottleneck in the product flow.

2.3.1 Planning process

The planning of the product production is done by one person at AC in Arnhem. The planner produces a planning for the next three months. This planning is handed to *MRP*-planners in Emmen in an *Excel* format. The planners in Emmen will insert this planning into *SAP R/3*.

Inputs: The planning process consists of a couple of inputs. The main inputs are briefly explained in the following list.

- *The forecast per aggregated product*: Since many products are in practice the same product, with only a different package quantity, different spool size or different product length per spool, the forecast of these products is merged into one forecast per aggregated product.
- *Current stock*: Some products, in the forecast, can partially be delivered from stock, resulting in a lower production requirement for that product.
- *Stock targets*: In a previous study, performed by *IBM* consultants, optimal stock targets have been calculated. Although influenced by production batch size, the stock targets are an indication to the planner about the production quantity of a certain product.
- *Active production*: The planning has to consider the current active production. This production will affect the future stock and might result in unnecessary changeovers if not taken into account.
- *List of constraints, technical and commercial*: Not every production line can produce or is allowed to produce a certain product. The planner needs to plan the production of products with respect to these constraints.

The current market situation shows a much higher demand for products than the production of AC can handle. This results in the following *extra* inputs to the planning process.

- *Allocation per resort*: Every resort has a production allocation in tons of a default product per month. This is calculated by the planner in actual machine hours. This allocation is a tool to “divide” the limited production amongst the different resorts.

This allocation is not a strict rule, but more a guideline. This results in room for negotiation between the different resorts.

- *Open orders*: The high demand forces the planner to lower the production of products below the forecast number. To decide where to cut, the confirmed customer orders can have priority over the other products.
- *Commercial strategy*: The commercial strategy of AC can help the planner to choose what products have priority over other products.

Planning process: With these inputs, the planner first plans the products that have very strong constraints. These constraints limit the possible planning choices, making it convenient to plan these products first. Then in an iterative process, the planner will fill the gaps with the other products in the forecast. In this planning process, the planner tries to produce an optimal planning with respect to the boundaries provided by the input, rush orders, transportation costs, and the number of changeovers.

Commercial constraints: Some customers of AC have specific requirements on the products they order. This involves packaging requirements, spool requirements, and technology requirements. This is further clarified in the following examples:

- The machinery of customer A can only handle spools of a specific type.
- Customer B would like to buy product x. However, he only accepts product x when it is produced with a certain production line.
- For the production of converted product y, 3.2 kilo product is required. Using spools with the default weight of 10 kilo would result in the waste of 0.4 kilo productİt would be more efficient to produce spools of 10 kilo, but with the converting operation in mind it is better to produce spools with the weight 9.6 kilo.

These commercial constraints do limit the aggregation of material numbers into one single aggregated product (Section 2.1.1). In a case where there exists a certain commercial constraint, two aggregated products are created for the same product type: one aggregated product for the specific material number satisfying the commercial constraint and the other for all the other material numbers of the same product type. Note, in the case of more different commercial constraints for the same product, this results in more than two aggregated products.

2.3.2 Production process

The product production process is a continuous process. This results in the fact that all different steps involved in the production of product happen in a continuous chain and do not require a separate planning. From the planning perspective, the production process can be seen as a black box .

For the production planning some information about the production process is relevant. In the following paragraphs relevant information about changeovers and technical constraints will be explained. The changeover time is relevant, since the changeover time differs between products. Not every production line can produce every product and not all products can be produced simultaneously, which makes the information about technical constraints relevant for the production planning.

Changeovers: When a production line needs to switch from producing product x to product y a changeover takes place. Since the production is a continuous process, the production line keeps running and produces B-quality product during most of the changeovers. This B-quality product is not wasted as it is required for the production of pulp. However, the required pulp ingredient can be spun more efficient when spun directly.

Specific changeover information is considered classified and has been omitted from this version of the document. Four different changeover types have been identified, i.e. a normal change, a blocking change, a soft change, and a spinneret change. These four changeover types all have a different changeover time.

Technical constraints: The technical constraints are documented in a so-called *constraints-file*. These technical constrains are considered classified information and have been excluded from this version of the document.

2.4 Problem description

The current market situation can be characterized by a higher demand for the products produced by AC than the production process can supply. AC does not like to turn down customers, resulting in a wide range of different products that have to be produced. However, the production of these different products with only a limited amount of production lines requires changeovers. These changeovers between those different products take time and lead to B-quality products. Considering the lost production time and high demand, this makes changeovers very expensive.

In order to optimize the product availability for sales, the planning of the production should lead to maximum effective production time. The many technical and commercial constraints, different changeovers, a wide range of different products and the high demand for these products results in a rather complex planning problem.

Chapter 3

Literature

There has been tremendous progress in planning and scheduling in the process industry during the last 20 years [6]. A small literature review has been performed to give an overview of the state-of-the-art planning and scheduling problems in the process industry.

In the process industry continuous and batch production systems can be distinguished [6]. Continuous production systems are often applicable to plants producing only a limited number of products, each in relatively high volumes. This production is typically performed by special purpose equipment allowing a continuous flow of materials with no clear defined start or end time. Alternatively, in batch production, small quantities of a large number of products are generally produced using multi-purpose equipment which are operated in batch-mode, i.e. there are well defined recipes specifying all different production steps, changeovers between recipes and production constraints.

The different planning levels in the process industry are often supported by an Advanced Planning System (APS). Three different planning levels are identified, i.e. network design, supply network planning and detailed production scheduling. Long term decisions are made in the network design phase, considering for instance plant acquisition, plant locations, and plant expansions. Mid-term decisions are made in the supply network planning phase, i.e planning, which provides the primary requirements for the final products to be produced at individual plants on the basis of demand data. The short term allocation of individual production resources over time to the production of the primary requirements is performed during the detailed production scheduling [8]. Planning and scheduling is part of company-wide logistics and supply chain management. However, to distinguish between planning and scheduling is often a rather artificial approach. In reality, the border lines between these areas are diffuse [6].

The goal of planning is an efficient utilization of the production and storage capacities. The planning horizon has to cover at least one seasonal cycle to be able to balance all demand peaks [8]. Time-indexed models, using a relative coarse discretization of time, e.g. a year, quarters, months, or weeks, are accurate enough in planning problems. Linear Programming (LP), Mixed Integer Linear Programming (MILP), and Mixed Integer Non-Linear Programming (MINLP) technologies are often appropriate and successful for problems with a clear quantitative objective function or quantitative multi-criteria objectives [6]. Planning problems are further discussed in Section 3.1.

Detailed production scheduling deals with the short-term allocation of resources over time to the production of the primary requirements determined by the supply-network planning [8]. In scheduling problems the focus on time is more detailed and may require even continuous time formulations. Furthermore, one faces rather (conflicting) goals and objectives: the optimal use of resources, minimal makespan, minimal operating costs, or maximal profit versus more qualitative goals such as reliability and robustness. Scheduling problems are usually NP-hard, no standard techniques are available and, actually, in many cases we are facing feasibility problems rather than optimization problems. Different solution approaches are found in the literature, e.g. mathematical optimization including MILP and MINLP, graph theory (GT), constraint programming (CP), and hybrid approaches integrating MILP and CP [6]. Scheduling problems are further discussed in Section 3.2.

3.1 Planning problems

Most of the planning problems in the process industry lead to MILP or MINLP models. Optimization algorithms in MILP and MINLP require a mathematical representation of the problem, describing the structure of the input and output data and variables, describing the constraints, and describing an objective function [4]. The mathematical MILP and MINLP models contain a number of building blocks: tracing the states of plants, modeling production, balance equations for material flows, transportation terms, consumption of utilities, cost terms and special model features. Using state-of-the-art commercial solvers, e.g., Fico Xpress by Fair Isaac or CPLEX by IBM ILOG, MILP problems can be solved quite efficiently [6]. More information about MILP and MINLP algorithms can be found in [4].

In [5] an example of a mathematical MILP model is formulated. This MILP approach combines strategic and operation planning in one model, capable of performing mid-term detailed analysis and long-term strategic analysis. The model describes many of the general building blocks of planning problems in the process industry. The different objective functions in the model allow for different operational and strategic analysis.

3.2 Scheduling problems

Solutions of scheduling problems known from literature are based on formulations of the detailed production scheduling problem as a time-indexed mixed-integer program depending on the time grid chosen. Some scheduling approaches successfully apply hybrid methods, involving mixed-integer and constraint programming [8]. The complexity of scheduling problems can easily exceed today's hardware and algorithmic capabilities. Using exact methods, such as MILP, in some cases it is not even possible to find feasible integer solutions [6].

Several scheduling methods are identified by [6]:

- *Processes and the state-task-network (STN) representation.* A STN models the different possible flows of material in a production process in a state-task diagram allowing for the optimization of the timing of the operations for each unit (i.e. which task, if any, the unit performs at any time during the time horizon); and the flow of material through the network. [7].
- *Decomposition: batching and batch scheduling.* The basic idea is to decompose the problem into two subproblems, i.e. batching and batch-scheduling. Batching determines the number and the size of the batches to be produced. Batch scheduling generates a feasible schedule and computes the start and end time [6]. An example of the decomposition approach is presented in [8]. A mixed-integer formulation

of the batching problem is solved approximately within seconds. During the batch scheduling step, a feasible schedule of minimal makespan is generated providing the start and end times of all tasks as well as the assignment of resources to the task [6].

  In [2] the scheduling problem is further decomposed. On the higher level, aggregate decisions are made determining the number, sequence and length of campaigns, i.e. a sequence of similar batches. The second level assigns an equipment unit to these batches. Changeovers and clean-out operations might result in conflicts in the assignment, therefore the third stage of the solution procedure resolves these resource conflicts with time-shifting procedures.

- *Heuristics and meta-heuristics:* A variety of heuristics are used to solve scheduling problems by simulating a given system and evaluating its objective function. These techniques are not problem specific and are based on generic principles and schemes, like genetic algorithms, simulated anealing and tabu search. All meta-heuristics have in common that they usually lack the proof of convergence and the proof of optimality [6].

Chapter 4

Problem identification

The problem motivation (Section 2.4) states that the planning problem at AC is rather complex. Literature (Chapter 3) shows that planning and scheduling problems in the process industry are usually NP-hard problems [6], that require optimization algorithms to find a possible planning solution.

To determine the problem complexity of the AC planning problem and to verify if this problem is part of the set of NP-hard problems, a mathematical proof needs to be constructed to prove the complexity of the problem. This proof requires a mathematical representation of the AC planning problem. This mathematical description, i.e. mathematical model, is presented in Section 4.1. Section 4.2 provides the complexity proof, showing that the decision variant of the AC planning problem belongs to the set of *NP-complete* problems.

4.1 Mathematical problem description

The mathematical description is divided into four parts. First the inputs of the planning problem are described in Section 4.1.1. Followed by a description of the output in terms of the input in Section 4.1.2. Then all the constraints, that constrain the possible outputs, are described in Section 4.1.3. Finally the objective function, that defines the quality of a planning solution, is defined in Section 4.1.4.

This mathematical description leads to the following definition of the AC planning problem.

Definition. AC planning problem

Instance: Input data as defined in Section 4.1.1 and a cost $C \in Z^+$.

Question: Is there a planning solution as defined in Section 4.1.2, respecting the constraints in Section 4.1.3 with a cost of at most C, i.e., for which $OF \leq C$, where OF is defined in Section 4.1.4?

4.1.1 Input

Time: The time unit in the mathematical model are hours, as specific production times, changeover times and constraints are all specified in hours.

The planning of the production is defined per time bucket t. A time bucket has the length of a fixed amount of time units, for instance one day or just one hour. Smaller time buckets result in a more accurate planning, but also result in more complexity. $t = 0$ is now. $t = 1$ is the first bucket of the planning and t_{max} is the last bucket in the planning, $0 \leq t \leq t_{max}$.

Forecasts and resorts allocations are specified per month m. Since months are not equal in size, this results in a different number of time buckets associated per month. $m = 1$ is the first month in the planning and m_{max} is the last month in the planning, $0 < m \leq m_{max}$.

The following functions are defined for the time measurements.

- b_{size} is the number of hours in a bucket.
- m_{max} is the last month in the planning.
- $nd(m)$ is the number of days in month m.
- $nb(m)$ gives the number of buckets in month m.

  $$nb(m) = \frac{nd(m) \times 24}{b_{size}}$$

- $fb(m)$ gives the first bucket of month m.

  $$fb(m) = \begin{cases} 1 & \text{if} \quad m = 1 \\ fb(m-1) + nb(m-1) & \text{if} \quad m > 1 \end{cases}$$

- t_{max} is the last time bucket in the planning.

  $$t_{max} = fb(m_{max}) + nb(m_{max}) - 1$$

Resort: Q is the set of different resorts within AC as described in Section 2.2. The function $ra(q, m)$ is the *resort allocation* for resort q in month m in hours.

Product: There are many different products within AC. These are mathematically represented with the following sets:

- P is the set of different products at an aggregated level. A single $p \in P$ is an aggregation of a number of products at SAP's material level.

  A product at SAP's material level is a packaging of spools, with a specific spool size, that contains a specific weight of a specific product. In SAP there exists a forecast at this product level. For many customers, the packaging, the spool size, or the spool weight does not matter. This results in the fact that the forecasts for some different products at the material level, are forecasts for the same product at the aggregated level.

  For every product $p \in P$ there exists a forecast in tons (1000 kg) per month. This forecast is a summation of the forecasts of material numbers in SAP corresponding to $p \in P$.

  The following functions are defined for $p \in P$ and $0 < m \leq m_{max}$:

  - $fc(p, m)$ gives the forecast of p for month m in tons.
  - $stockTarget(p)$ is the stock target for product p in tons.
  - $stDays(p)$ is the number of days of yearly sales that define the stock target.

  - $stc(p)$ are the costs per day of supply when the stock is below the stock target.
  - $initialStock(p)$ is the stock for product p at $t = 0$ in tons.
  - $profit(p)$ is the profit for product p per ton in euros.
  - $ndcModifier(p, m)$ is the modifier factor of the profit for product p in month m.
  - $ndc(p, m)$ are the non-delivery costs per ton for product p in month m.

    $ndc(p, m) = profit(p) \times ndcModifier(p, m)$
  - $ruFactor(q, p)$ is the resort usage factor. When product p is being produced, this factor defines for how much resort q is charged. For instance, when product p is produced for one hour and for resort q the $ruFactor(p, q)$ is 0.5, resort q will be charged for a half hour.
  - $rupb(q, p)$ is the resort usage per bucket. This is the number of hours charged to resort q for a bucket of production product p .

    $rupb(q, p) = ruFactor(q, p) \times b_{size}$

- R is the set of different product recipes. A single $r \in R$ is the recipe to produce a single $p \in P$. A single $p \in P$ might have more than one corresponding recipe $r \in R$.

  The following functions are defined for recipe $r \in R$:

  - *Product identification parameters:*
    - $pd(r) \in P$ is the product produced by r.
    - $type(r)$ is the integer type number of the product produced by r. This is the type number without the optional “D” in front of it.
    - $titer(r)$ is the titer of the product produced by r.
    - $filament(r)$ is the number of filaments of the product produced by r.
    - $ttf(r)$ is the type, titer and filaments identification of the product produced by r.
    - $sw(r)$ is the spool weight of the product produced by r.
    - $ss(r)$ is the spool size of the product produced by r.
  - *Production parameters:*
    - $tph(r)$ is the number of tons produced per hour by r.
    - $tpb(r)$ is the number of tons produced per bucket by r.

      $tpb(r) = tph(r) \times b_{size}$
    - $cph(r)$ are the production costs per hour for r.
    - $cpb(r)$ are the production costs per bucket for r.

      $cpb(r) = cph(r) \times b_{size}$
  - *Changeover and constraint parameters:*
    - The changeover and constraint parameters are considered classified information and have been omitted from this version of the document.

Resources: There are two different types of resources, *primary* and *secondary* resources, involved in producing products $p \in P$. The *primary* resources are the resources that actually produce the product and the *secondary* resources are resources that are constraining the primary resources with a certain capacity.

- H is the set of high-rise resources. Every $h \in H$ has a certain throughput capacity. $hrCap(h)$ defines the throughput capacity of $h \in H$ in kilos per hour.
- C is the set of cooling resources. Every $c \in C$ has a certain cooling capacity. $cCap(c)$ defines the cooling capacity of $c \in C$ in m^3 per hour.
- L is the set of production lines. Every production line is linked to one $h \in H$ and one $c \in C$. The following functions are defined for $l \in L$:
    - $name(l)$ is the identification of l.
    - $hrRes(l) \in H$ is the high-rise resource connected to l.
    - $coolRes(l) \in C$ is the cooling resource connected to l.
    - $minTp(l)$ is the minimum throughput required on l.
    - $hasKlitter(l) \in \mathbb{B}$ defines if l can perform *klit* operations.
    - $hasAA(l) \in \mathbb{B}$ defines if l can apply *AA-avivages*.
- All production lines depend on the evaporation capacity of the acid recycling, the total throughput on all production lines is limited by this capacity. *arCap* is the capacity of the evaporator in the acid recycling.

Changeovers: When recipes $r_1 \in R$ and $r_2 \in R$ are produced in sequence on the same production line, a changeover takes place. The time of the changeover is defined by the changeover type. There are four different types of changeovers (Section 2.3.2), each with its own length in hours.

For changeovers the following functions exist for $r_1, r_2 \in R$:

- $cType(r_1, r_2) \in \{no, soft, normal, block, spin\}$ are the changeover type between recipes r_1 and r_2.
- $ct(r_1, r_2)$ returns the changeover time between recipes r_1 and r_2.

Current production: At $t = 0$, for all $l \in L$ there exists an r currently being produced at l. As this information is relevant for possible changeovers, the following functions are defined for $l \in L$:

- $initR(l) \in R$ is the recipe currently being produced on l.

4.1.2 Output

A solution to the planning problem is an assignment of a recipe $r \in R$ to each time bucket t, $0 < t \leq t_{max}$ for every production line $l \in L$ respecting the constraints defined in Section 4.1.3

This solution can be represented in a table; an example planning output is shown in Table 4.1. The horizontal axis represents the time buckets t, $0 < t \leq t_{max}$. The vertical

	t_1	t_2	t_3	t_4
l_1	r_1	r_1	r_1	r_2
l_2	r_2	r_3	r_3	r_7
l_3	r_6	r_7	r_7	r_7

Table 4.1: Example planning output; $l \in L$ and $r \in R$

axis represents all the $l \in L$. Every cell in the table should be filled with a recipe $r \in R$, representing the recipe on the production line l in time bucket t.

The following functions are defined for production line $l \in L$, resort $q \in Q$, month m, $0 < m \leq m_{max}$, and time bucket t, $0 \leq t \leq t_{max}$ over the output:

- $r_{l,t} \in R$ assigned to the corresponding t and l in the solution, where $r_{l,0} = initR(l)$
- $prod(p, m)$ is the total production in tons of product p in month m according to the planning. This function takes care of changeovers by subtracting the changeover production loss from the amount that would have been produced in this bucket. This might result in a bucket with negative production. However, this will be corrected by the following buckets, producing the same product.

  $$prod(p,m) = \left(\sum_{l \in L} \left(\sum_{t_1 \leq t < t_n | p_{l,t} = p} tpb(r_{l,t}) - (tph(r_{l,t}) \times ct(r_{l,t-1}, r_{l,t})) \right) \right)$$

  $$\text{where } t_1 = fb(m), t_n = fb(m) + nb(m), \text{and } p_{l,t} = pd(r_{l,t})$$

- $st(p, m)$ gives the stock of p at the end of month m in tons. A negative stock represents a backlog at the end of the month.

  $$st(p,m) = \begin{cases} initialStock(p) & \text{if} \quad m = 0 \\ st(p, m-1) + prod(p,m) - fc(p,m) & \text{otherwise} \end{cases}$$

- $pn(p, m)$ is the production need for product p in month m in tons.

  $$pn(p,m) = (fc(p,m) + stockTarget(p)) - st(p, m-1)$$

- $ru(q, m)$ is the production usage of resort q for month m.

  $$ru(q,m) = \left(\sum_{l \in L} \left(\sum_{t_1 \leq t < t_n} rupb(q, p_{l,t}) \right) \right)$$

  $$\text{where } t_1 = fb(m), t_n = fb(m) + nb(m), \text{and } p_{l,t} = pd(r_{l,t})$$

4.1.3 Constraints

There are a number of technical constraints that limit the output. The technical constraints described in Section 2.3.2 have been converted into mathematical expressions in terms of the planning input and planning output.

Some of these constraints are considered classified information and have been omitted from this version of the document.

Global constraints:

- *Constraint enforcing enough subsequent buckets with the same recipe to prevent negative production due to the changeover production loss method.*

  For all $0 < t \leq t_{max}$ and all $l \in L$:

  $$r_{t-1,l} \neq r_{t,l} \Rightarrow 0 < \left(\sum_{x | t \leq x < t_n} tpb(r_{l,x}) - (tph(r_{l,x}) \times ct(r_{l,t-x}, r_{l,x})) \right)$$

  $$\text{where } \; t < t_n \wedge r_{t,l} \neq r_{t_n,l} \wedge (\forall y \mid t < y < t_n \wedge r_{t,l} = r_{y,l})$$

  If the planned recipe differs from the planned recipe in the previous time bucket, the sum of the total production should be greater than 0.

- *High-rise resource capacity constraint.* This constraint refers to technical constraints 1, 6, and 11 in Section 2.3.2.

  For all $0 < t \leq t_{max}$ and all $h \in H$:

  $$\left(\sum_{l \in L | hrRes(l) = h} tp(r_{l,t}) \right) < hrCap(h)$$

  The sum of the required throughput for the recipes planned on the production lines connected to a high-rise resource should not exceed the capacity of that high-rise resource.

- *Cooling resource capacity constraint.* This constraint refers to technical constraint 15 in Section 2.3.2.

  For all $0 < t \leq t_{max}$ and all $c \in C$:

  $$\left(\sum_{l \in L | coolRes(l) = c} cool(r_{l,t}) \right) < cCap(c)$$

  The sum of the required cooling for the recipes planned on the production lines connected to a cooling resource should not exceed the capacity of that cooling resource.

- *Evaporation capacity constraint.* This constraint refers to technical constraint 14 in Section 2.3.2.

  For all $0 < t < t_{max}$:

  $$\left(\sum_{l \in L} ar(r_{l,t}) \right) < arCap$$

  The sum of the total evaporation requirement from the acid recycling for all planned recipes at a certain time bucket should be lower than the evaporation capacity.

- *A klit-operation is only possible at certain production lines.* This constraint refers to technical constraint 9 in Section 2.3.2.

  For all $0 < t \leq t_{max}$ and all $l \in L$:

  $$klit(r_{l,t}) \Rightarrow hasKlitter(l)$$

  If a planned recipe requires a *klit*-operation, the production line should have the required *klit*-machinery.

- *AA-avivage application is only possible at certain production lines.* This constraint refers to technical constraint 10 in Section 2.3.2.

  For all $0 < t \leq t_{max}$ and all $l \in L$:

  $aa(r_{l,t}) \Rightarrow hasAA(l)$

  If a planned recipe requires the application of an *AA-avivage*, the production line should have the required *AA-avivage*-machinery.

- *Minimum throughput constraint.* This constraint refers to technical constraints 17, 18, and 19 in Section 2.3.2.

  For all $0 < t \leq t_{max}$ and $l \in L$:

  $tp(r_{l,t}) \geq minTp(l))$

  The required throughput of the planned recipe should be at least greater than the minimum throughput of the production line.

4.1.4 Objective function

The objective function of the AC planning problem is a combination of four different objectives with different weights, i.e. w_{ndc}, w_{st}, w_{ra}, and w_{pc}. The best planning output minimizes the outcome of this function.

$$OF = w_{ndc} \times O_{ndc} + w_{st} \times O_{st} + w_{ra} \times O_{ra} + w_{pc} \times O_{pc}$$

- *Non-delivery costs:* The objective of the planning solution is to minimize the difference between the production need and the actual production per month. This difference is measured as non-delivery costs, i.e. the money that would have been earned if the product was produced.

  $$O_{ndc} = \left(\sum_{p \in P} \left(\sum_{0 < m \leq m_{max}} ndc(p,m) \times (0 \uparrow (pn(p,m) - prod(p,m))) \right) \right)$$

  The sum of the non-delivery costs times the production shortage per month per product.

- *Stock target:* In order to account for uncertainties in the forecast, AC tries to keep the stock levels of all the products at a defined target. The planning solution should keep all the stocks above their required target. If the stock is beneath its target, the planning solution should be penalized for that.

  $$O_{st} = \left(\sum_{p \in P} \left(\sum_{0 < m \leq m_{max}} stc(p) \times \frac{0 \uparrow (stockTarget(p) - (0 \uparrow st(p,m))}{stDays(p)} \right) \right)$$

  The sum of the stock deficit costs times the number of days, the stock is below the target, per month per product.

- *Resort allocation:* As the production demand is higher than the actual production capacity, AC tries to divide the production capacity among the different *resorts* according to the *resort* allocation. An optimal solution should minimize the non-delivery costs, while dividing the production capacity as equal as possible according to the *resort* allocation.

$$O_{ra} = \left(\sum_{q \in Q} \left(\sum_{0 < m \leq m_{max}} 0 \uparrow (ru(q,m) - ra(q,m)) \right) \right)$$

The sum of the total resort allocation exceeding per resort per month in machine hours.

- *Production costs:* When the complete demand can be fulfilled and all the stocks are above their required target, the next objective is minimizing the production costs.

$$O_{pc} = \left(\sum_{l \in L} \left(\sum_{0 < t \leq t_{max}} cpb(r_{l,t}) \right) \right)$$

The sum of the recipe production costs per bucket per production line.

4.2 Complexity of the problem

A problem is an *NP-complete* problem, if i) for a given solution of the problem it can be checked in polynomial time that it is a valid solution of the problem, and ii) all instances of a known *NP-complete* problem can be mapped to an instance of the problem in polynomial time [1].

This section will show that the decision variant of the AC planning problem defined in Section 4.1 is *NP-complete.* First the following *NP-complete* problem is introduced:

Definition. Multiprocessor Scheduling

Instance: Set of T tasks, number of $m \in Z^+$ of processors, length $l(t) \in Z^+$ for each $t \in T$, and a deadline $D \in Z^+$.

Question: Is there a m-processor schedule for T that meets the overall deadline D, i.e. a function $\sigma : T \rightarrow Z_0^+$ such that, for all $u \geq 0$, the number of tasks $t \in T$ for which $\sigma(t) \leq u < \sigma(t) + l(t)$ is no more than m and such that, for all $t \in T$, $\sigma(t) + l(t) \leq D$?

In [1] it is shown that the Multiprocessor Scheduling Problem is *NP-complete.*

Theorem. The AC planning problem is *NP-complete*

Proof. The decision variant of the AC planning planning problem is in *NP* as for a given planning solution it can be checked in polynomial time that it respects all the constraints and it has cost $OF \leq C$ (Section 4.2.1). Furthermore, the *NP-complete* Multiprocessor Scheduling problem is a special case of the AC planning problem (Section 4.2.2).

4.2.1 Solution TA planning problem in *NP*

The solution of the AC planning problem is in *NP*; if a given solution can be checked in polynomial time, i.e. if all the constraints (Section 4.1.3) can be checked in polynomial time and if the objective function (Section 4.1.4) can be calculated in polynomial time for a given solution.

Both, constraints and objective function, do not have any exponential definition, resulting in the fact that all constraints can be checked in polynomial time and that the objective function can be calculated in polynomial time.

4.2.2 Mapping to an instance of the TA planning problem

It will be shown that the multiprocessor scheduling problem is a special case of the AC planning problem by mapping all instances of the multiprocessor scheduling problem to an instance of the AC planning problem, i.e. the inputs (Section 4.1.1) of the AC planning problem will be expressed in the instance variables of the described multiprocessor scheduling problem.

Time: The multiprocessor scheduling problem does only identify a deadline D, a positive numeric value. The lengths of the tasks in the multiprocessor scheduling problem are also defined as a positive numeric value. In the instance of the AC planning problem the time values are chosen, such that $t_{max} = D$.

- $b_{size} = 24$
- $m_{max} = 1$
- $nd(m) = D$

Resort: The multiprocessor scheduling problem does not identify resorts. Appropriate values have been chosen to make sure that resorts do not negatively influence the output.

- $Q = \{q\}$
- $ra(q, 1) = \infty$

Product: For every $t \in T$ of the multiprocessor scheduling problem there exists one $p_t \in P$ that corresponds to this t. If $t_1, t_2 \in T$ both correspond to $p_t \in P$ then $t_1 = t_2$ and if $p_t1, p_t2 \in P$ both correspond to $t \in T$ then $p_t1 = p_t2$.

The following values are defined for $p_t \in P$, such that the forecast of a product equals the desired task length and that the other values do not negatively influence the output.

- $fc(p_t, 1) = l(t)$
- $stockTarget(p_t) = 1$
- $stDays(p_t) = 1$
- $stc(p_t) = 0$
- $initialStock(p_t) = 0$
- $profit(p_t) = 1$
- $ndcModifier(p_t, 1) = 1$
- $ruFactor(q, p_t) = 1$

For every $p_t \in P$ there exists one $r_t \in R$. A product in the AC planning problem has several parameters that affect changeovers and constraints. The multiprocessor scheduling problem does not have these changeovers and constraints. To make sure that these parameters do not affect the output, values are chosen that make sure that specific AC constraints will not affect the output.

For $r_t \in R$ the following values are defined, such that producing one unit of p_t requires one time bucket in the AC planning problem and that the other values do not negatively influence the output.

- $pd(r_t) = p_t$
- $type(r_t) = 0$
- $titer(r_t) = 0$
- $filament(r_t) = 0$
- $ttf(r_t) = 0$
- $sw(r_t) = 0$
- $ss(r_t) = 0$
- $cph(r_t) = 0$
- $tph(r_t) = \frac{1}{b_{size}}$

Resources: The multiprocessor scheduling problem only has a fixed number, m, resources with a capacity of one. Resulting in the fact that all resources in the AC planning problem, except the production lines L, can be ignored. To make sure that these resources do not affect the output, all capacities are set at ∞.

- $H = \{h\}$
- $hrCap(h) = \infty$
- $C = \{c\}$
- $cCap(c) = \infty$
- $arCap = \infty$

The number of production lines $l \in L$ is equal to the number of processors m, i.e. $\#L = m$. The multiprocessor scheduling problem does not require any special resource features. To make sure that these features do not negatively influence the output, these features are set a the least constraining value, i.e. the values that allow everything. For all $l \in L$ the different values are defined:

- $name(l) = name$
- $hrRes(l) = h$
- $coolRes(l) = c$
- $minTp(l) = 0$
- $hasKlitter(l) = true$
- $hasAA(l) = true$

Changeovers: The multiprocessor scheduling problem does not require changeovers, resulting in $ct(r_1, r_2) = 0$ for all $r_1, r_2 \in R$.

Current production: The current production influences a possible changeover for the first recipe being produced on a production line. Since the changeover time is always zero for every instance of the multiprocessor scheduling problem, the current production can be any recipe. For all $l \in L$ the current production is defined as $initR(l) = r_x$, where $r_x \in R$.

Chapter 5

Proposed solution

It has been proven that the decision variant of the AC planning problem is *NP-complete* (Section 4.2). This complexity justifies the use of an optimization approach, suggested by the literature (Chapter 3). This optimization variant of the AC planning problem tries to find an solution that minimizes the function OF (Section 4.1.4).

This optimization approaches are supported by some powerful software tools that exist on the market. This planning software provides the planner with more insight and gives him more powerful tools in defining and calculating an optimal planning. However, this planning software requires a lot of specific information, that may be scattered across different files and databases. The most suitable way to make this information available to this software, is to transform the mathematical model (Section 4.1) into a “intermediate” model. This intermediate model will then be mapped to the more generic model of a planning tool.

The planning tool that is used in this process is *IBM ILOG Plant PowerOps* (*PPO*). *PPO* is an integrated production planning and detailed scheduling solution for use in the batch process industry [3]. The internal algorithms of *PPO* correspond to the ones identified in the literature (Chapter 3). The calculation of schedules in *PPO* consists of three stages. First a planning engine assigns recipes to production lines for every time bucket using a *MILP* approach. Followed by a batching stage, converting the planning output into batches of appropriate sizes using *MILP*, *CP*, and heuristic algorithms. Finally a scheduling stage schedules these batches at a specific starting time, respecting constraints and changeovers using *CP* and Large Neighborhood Search algorithms. *PPO* can easily be extended with external plugins. This plugin feature allows for a mapping from an intermediate model to the internal model of *PPO*.

The intermediate model is a direct transformation of the mathematical model (Section 4.1) into a data model. This data model is a collection of all required information in several tables. The data in the intermediate model needs to be imported into the internal model of *PPO*. This import, i.e. mapping, is performed by a special plugin extension of *PPO*, written for the AC model.

The planning and scheduling engines of *PPO* try to calculate the best schedule for the given input. The planning engine of *PPO* assigns recipes to production lines for every time bucket, conforming to the mathematical model (Section 4.1). However, the planning engine is unable to handle specific compatibility constraints and exact changeovers. The scheduling engine is required for that task. The recipe assignments are converted into batches. These batches are scheduled at a specific starting time, respecting constraints and changeovers. Since the scheduling engine does not assign recipes to a production line in a certain bucket, the output does not match the output previously described in the mathematical model. The output of the mathematical model reflects better the output of the manual planning process at AC. The manual planning process does not assign a specific starting time to batches, it assigns recipes to production lines in specific buckets, i.e. to specific days, respecting constraints and changeovers.

The quality of the schedule is measured with several key performance indicators. *PPO* tries to minimize these values according to defined weights. Section 5.1 will further describe the key performance indicators used in the AC model.

5.1 Performance indicators

Similar to the mathematical model (Section 4.1), *PPO* uses some performance indicators to measure the quality of a planning. The objective of *PPO* is to minimize the weighted sum of the performance indicators one selects. The planning and scheduling weights define the relation between key performance indicators and allow for an optimization of a planning in a certain direction. The following performance indicators are used in the AC model.

- Non-delivery costs (Section 5.1.1)
- Inventory deficit costs (Section 5.1.2)
- Setup costs (Section 5.1.3)
- Processing costs (Section 5.1.4)

These performance indicators are similar to the objectives in the objective function described in Section 4.1.4. Note that the resort allocation objective could not be directly translated into a performance indicator present in *PPO*. The plugin framework of *PPO* does allow for the specification of custom performance indicators. This makes it possible to for example add the resort allocation performance indicator, if it seems necessary in the future (Section 8.2).

5.1.1 Non-delivery costs

Non-delivery costs is a performance indicator that measures the demand not being fulfilled by the planning solution. A cost per ton is allocated to every demand. Every ton of demand not being fulfilled will add this cost to the total non-delivery costs.

In the AC model, these costs per demand are based on the contribution of the product. The contribution of a product is related to the profit margin of the product. If a demand is not fulfilled, the non-delivery costs is the money that would have been earned if the demand would have been fulfilled.

Within AC some products represent a strategic value that is higher than the actual money that is being earned with this product. The strategic value can be based on many aspects, for instance competition in the market, satisfying important customers, or some contracted customer orders. In order to make this strategic distinction between products, a modifier factor is introduced in the demand data. The original contribution of the product is multiplied by this factor. For instance, a modifier factor of 3 for a certain product would specify that this product has a strategic value that is three times more important than the actual contribution of the product.

The result of this modifier factor is that the planning and scheduling engines will respect the strategic value of a product. The downside of this implementation is that the non-delivery costs are not representing actual costs, i.e. money. Due to this implementation the actual gains with the planning and scheduling solution might be smaller. An export of the transactional data to Excel, allows for the calculation by hand of the non-delivery costs in real costs.

5.1.2 Inventory deficit costs

AC has defined optimal stock targets for their products, in order to account for uncertainties in the forecast. A planning and scheduling solution should take these targets into account and try to replenish the stock of products when there is available production time. The stocks that are below the stock target are measured with the inventory deficit costs performance indicator.

Stock targets are specified in days of the yearly sales. Resulting in a stock target of an amount in tons for a certain product. This target is derived using the days of yearly sales, i.e. the target corresponds to a number of days of sales. The performance indicator measures the number of days the stock of a product is below its target and multiplies that with a predefined cost per day. For instance for a certain product, the stock target is 20 ton, which corresponds to 4 days of its yearly sales and costs per day below the target is 50. If the actual stock is 10, the stock of that product is 2 days below its target. The inventory deficit costs of this product would be 100, i.e. 2 times 50.

5.1.3 Setup costs

AC is of the opinion that minimizing the number of changeovers is required to minimize the non delivery costs (Section 2.4). To be able to measure and verify this opinion the setup costs performance indicator is introduced.

The number of changeovers, i.e. the number of hours, is more important than the actual costs of a changeover. This results in the default costs of 1 per hour in the model data, making the setup costs equal to the number of changeover hours.

5.1.4 Processing costs

At the moment AC faces production capacity problems, making the non-delivery costs the most important performance indicator. In the future it might be the case that it is possible to fulfill the complete demand. When this is the case, a schedule that minimizes the production costs would be better than a schedule that ignores production costs. This results of the inclusion of the processing costs performance indicator. The current market situation does not make processing costs an issue, resulting in this performance indicator only being present for future purposes.

This performance indicator is an addition of all the recipe processing costs. If producing products with a certain recipe costs 10 Euro per hour and the recipe runs for 10 hours, the processing costs of this production would be 100 Euro. At the moment the recipe costs are all set at 0, making this performance indicator useless. If in the future this performance indicator is required, the model should be filled with the correct recipe costs.

Chapter 6

Analysis

The implementation of the AC model in *IBM ILOG Plant PowerOps* (*PPO*) provides results with performance indicators (Section 5.1.1). These performance indicators allow for model analysis, analyzing whether or not the planning and scheduling optimization in *PPO* with the AC model extension is a good solution to the problem identified at AC (Section 2.4). This is analyzed by comparing a real planning, made by the planner, to a calculated planning with the AC model (Section 6.2).

PPO has a planning and scheduling engine. Both engines try to find an optimal solution in a predefined calculation time. Before analysis can provide meaningful results, the quality of the calculated schedule should first be analyzed with respect to the calculation time (Section 6.1).

The ability to provide different weights for performance indicators and the ability to edit the data allow for some further what-if analysis of interesting topics within AC. The following analyses have been performed:

- Schedule quality over calculation time (Section 6.1)
- Manual schedule vs. calculated schedule (Section 6.2)
- Batch size of low demand products (Section 6.3)
- Minimizing setup costs vs. non delivery costs (Section 6.4)
- Impact of the initial stock (Section 6.5)

6.1 Schedule quality over calculation time

The planning and scheduling engine of *PPO* tries to find an optimal solution within predefined time bounds. It might have a great impact on the quality of the solution,

Performance indicator	Planning weight	Scheduling weight
Non-delivery costs	25	1
Inventory deficit costs	1	0
Setup costs	0	0
Processing costs	0	0

Table 6.1: Default planning and scheduling weights.

when these times are not well chosen. The algorithms used in the planning and scheduling engine provide a better solution over time, resulting in a solution with poor quality when the calculation time is too short. On the other hand, if the calculation time is set too high, computing power is being wasted. The goal of this analysis is to find optimal planning and scheduling calculation times for further analysis.

This analysis has been performed with a data set considering the demands of the months April, May and June of 2011. The planning weights of performance indicators are set to the default values (Table 6.1). Note that inventory deficit costs are not taken into account by the scheduling engine. The inventory deficit costs performance indicator puts too much stress on the scheduling engine, resulting in no usable scheduling output.

First the optimal planning time has been investigated. Only the planning engine has been used for this investigation. The non-delivery costs and inventory deficit costs have been calculated with different planning calculation times. The results of this analysis are represented in Table 6.2. These results show that after a calculation of 50 seconds the non-delivery costs are already at the lowest possible value. The inventory deficit costs does not show much improvement in calculation times beyond 100 seconds. Since the scheduling engine is not able to handle stock-deficit costs, the extra gain with more planning time is most of time wasted investment. Considering the relatively low improvement between higher planning times and the behavior of the scheduling engine, it can be concluded that using a planning calculation time of 200 seconds will provide results of acceptable quality.

With a planning time of 200 seconds, the optimal calculation for the scheduling engine has been investigated. The non-delivery costs and inventory deficit costs have been calculated with different scheduling calculation times. Since the scheduling engine can not handle inventory deficit costs, these costs are not being optimized by the schedule solution. This results in the fluctuations in inventory deficit costs over time. The generated production orders by the planning engine, that account for the inventory deficit costs, do not have a due date, with as result that these production orders are planned beyond the scheduling horizon. The scheduling horizon corresponds to the end of the last month in the input data. Since the time beyond the scheduling horizon is not available in practice, the production orders that start beyond this horizon have been removed from the solution. The results of this analysis are represented in Table 6.3. The results show that the non-delivery costs keep improving over time and that the inventory deficit costs stay at about the same level. From this we can conclude that higher scheduling time keeps resulting in better non-delivery costs. Since the difference between non-delivery costs at 5000 seconds and 10000 seconds is relatively low, 5000 seconds is an acceptable time to perform further analysis. Note that for the calculation of schedules, the difference of 700k for the extra 5000 seconds calculation time would be worth the investment.

Planning time seconds	**Non-delivery costs** million	**Inventory deficit costs** million
50	0	38.28
100	0	37.23
200	0	37.19
350	0	37.16
500	0	36.99

Table 6.2: Non-delivery costs with respect to planning time

Scheduling time seconds	**Non-delivery costs** million	**Inventory deficit costs** million
500	40.2	527.7
1000	39.1	533.5
2500	38.2	506.6
5000	37.8	551.5
7500	37.6	503.2
10000	37.1	505.7

Table 6.3: Non-delivery costs with respect to scheduling time

6.2 Manual schedule vs. calculated schedule

A planning solution, created by the planner using the planing process currently used to create a production planning (Section 2.3.1), has been manually constructed in *PPO*. This planning covers the months April, May and June of 2011 and has been constructed in the data file with the corresponding demands for these months.

The constraints checker in *PPO* noticed that some technical constraints (Section 2.3.2) were being violated in the manual schedule. Some of these violations were caused by missing data, erroneous data, and by exceptions to certain constraints. The constraints that were being violated will be listed below.

- The manual schedule violated some capacity constraints. It turned out that the planner was unaware of these technical constraints and never accounted for them in his production plans. These differences are handled by softening the applicable constraints, allowing the same behavior as the manual schedule.
- A certain product could only be produced by certain production lines in the model, while the real planning solution produced this product on a different production line. It turned out that the product violated the minimum throughput constraint. It turned out that the minimum throughput was violated by 0.001, probably a rounding error. This issue has been resolved to lower the minimum throughput of the production lines with 10 for the production lines with this minimum throughput .
- Some products contained errors in the recipe data and some products had new recipes that allowed more possible setups. The recipe data has been updated according to these errors and improvements.
- Some violations turned out to be exceptions to certain constraints. Including these exceptions in the model would require major modifications to the intermediate model and mapping as it is impossible to account for these exceptions in the current implementation of the model. Considering the low amount of exceptions and the huge effort to include these into the model, it has been decided to let the planner handle these exceptions by hand in the manual post processing.

With a planning time of 200 seconds and a scheduling time of 10000 seconds and the default weights (Table 6.1) the calculated planning for the same demand data has been constructed. Since results from the time analysis (Section 6.1) showed that the scheduling output keeps improving over time beyond 5000 seconds and this schedule needs to be compared with a real schedule, a scheduling time of 10000 seconds is used. The data required some minor adjustments, since there are some scheduled maintenance stops for some specific production lines in the months April and May. These maintenance stops have been added to the model data before the calculated planning was calculated. The construction of the calculated schedule required some iterations, since some errors in the recipe data violated some commercial constraints. The calculated schedule solution has some clear differences with the manual solution.

- The calculated schedule solution contains many gaps in the planning, especially on the production lines that share the high-rise process. The scheduling engine has problems scheduling production on the heavily constrained special production lines. The addition of the maintenance stop made it even worse. Some manual post processing is required to fill and adjust these gaps with some useful compatible production. The results of post processing are described in one of the following paragraphs.
- The calculated schedule solution contains many more changeovers, including much more spinneret changeovers. The calculated schedule solution contains about 25 spinneret changeovers. With some manual post processing it might be possible to lower this number of spinneret changeovers.

At first the calculated schedule does not seems to be of very high quality, however the performance indicators (Table 6.4) prove otherwise. The performance indicator with the highest weight, i.e. the most important performance indicator, shows a reduction of 32.6%, lowering the non-delivery costs with 12.7 million. The previous observation about the increased number of changeovers is confirmed by the setup time performance indicator, the amount of setup hours in the calculated solution is more than 4 times higher than in the manual planning. The inventory deficit costs do also show an increase in the calculated schedule. This is partially due to the scheduling engine not using this as a scheduling objective. When all the gaps on the special production lines are filled with useful production, the inventory deficit costs will probably be more in the range of the inventory deficit costs of the manual planning. The manual post processing steps shown in one of the following paragraphs gives some evidence of this assumption.

Non-delivery costs differences: An investigation at a product level to identify the differences has been performed. An Excel export gives more insight in the possible non-delivery costs of all products, i.e. the distribution of the contribution over all products. It turns out that three products are responsible for more than 50% of the total contribution in the monthly demand. Note that the contributions are modified to account for the strategic value.

With these three products the calculated schedule gains the most upon the manual planning. On the other hand plans the manual schedule more products that contribute less to the non-delivery costs, while these products are partially ignored by the calculated schedule. These products all have a relative low contribution margin, compared to the three products listed before.

Schedule	**Non-delivery costs** million	**Inventory deficit costs** million	**Setup time** hour
Manual	38.9	466.1	309
Calculated	26.2	524.7	1289
Manual post processed	18.6	517.8	1050

Table 6.4: Performance indicators for the manual and calculated schedule solution.

Schedule	Normalized non-delivery costs million
Manual	19.4
Calculated	16.1
Manual post processed	12.7

Table 6.5: Normalized non-delivery costs.

Since the differences in non-delivery costs are mainly caused by products that have their non-delivery costs modified to express this strategic value, it is investigated what the real non-delivery costs would be in both planning solutions. In the Excel export, the modified non-delivery costs are normalized to their original contribution values. The results in Table 6.5 represent real money, showing an improvement of 3.3 million, i.e. 17%, between the manual and the calculated schedule. This shows that the strategic value modifier had a huge impact on the non-delivery costs output.

Final stock differences: The differences between the inventory deficit costs have been analyzed. It turns out that both schedules have the same amount of products below their stock target. However, when the calculated planning is below the target it is much deeper below the target. It is also noticeable that the three products mentioned before have a final stock significantly higher than the manual planning and the stock target. Whether or not this behavior is positive or not, is subject to discussion. Since these products are at the moment constantly being produced, AC does not maintain high inventory levels of these products. Considering the impact of these products on the scheduling output, the higher stocks might result in more production of smaller products in the following months.

Post processing: The scheduling engine has some limitations affecting the quality of the scheduling output. These limitations require some post processing of the schedule output to improve the results. The impact of post processing the schedule output of the calculated planning has been analyzed. Two different post processing methods showed some improvements of the scheduling solution. Note that both methods have been applied independently of each other, if both methods are applied in sequence even better results could be achieved.

- The first post processing method involves manual editing the scheduling solution. The heavy constrained special production lines are not handled very well by the scheduling engine, resulting in many gaps in the output. By manually post processing the generated production orders of the calculated schedule on two special production lines, i.e. changing production orders into compatible ones and grouping similar production orders together, the non-delivery cost have been reduced from 26.2 million to 18.6 million. The inventory deficit costs have been reduced from 524.7 to 517.8 and number of setup hours have been decreased to 1050 hours.
- The second post processing method involves another run of the schedule engine, with some adjusted weights. The scheduling engine can not handle the simultaneous optimization of the non-delivery costs and the inventory deficit costs. This post processing method locks the generated production orders, of the calculated schedule, that account for the non-delivery costs, i.e. all the production orders before the scheduling horizon. Then in another run of the scheduling engine only the inventory deficit costs are optimized. A run of 2000 seconds results in a schedule output that

improved the inventory deficit costs from 524.7 million to 458.7 million. Although not part of the optimization, the non-delivery costs show some minor improvement from 26.2 million to 25.9 million. Since all the production orders are locked, this is improvement is unexpected and is probably just luck. The setup time dropped from 1289 hours to 1158 hours.

6.3 Batch size of low demand products

Within AC there are some low demand products that most of the time require a spinneret changeover before production and after production of this product. At the moment these products are produced only two or three times a year with quite large batches, producing enough to serve the demand of multiple months. An interesting analysis is to investigate how the planning and scheduling engine handles these products.

For this analysis demand data for the months January until June have been used. The planning engine of *PPO* decides in what time bucket what product is being produced and picks the required production quantity, considering available capacity. The scheduling engine then tries to schedule all the suggested products, taking account of setups and compatibility constraints. Since the planning engine decides the batch sizes and production quantities, first the output of the planning engine will be examined for two of these low demand products. A planning solution has been calculated using the default planning weights (Table 6.1) with a planning time of 500 seconds.

Figure 6.2 and Figure 6.1 show the stock coverages of these products. The red line is the stock target, the green area is the actual stock, the yellow bars are the demand points, the pink bars represent the fulfillment of the demand and the blue bars are production. These stock coverages do not really show the expected behavior of one large production batch for these products. Especially product Y shows more a monthly production from the point the initial stock has been depleted. From this can be concluded that *PPO* does not find it necessary to group these productions together. This is no indication that grouping these productions would result in better results, however with the current demand data *PPO* did not find a better planning solution when grouping these products.

6.4 Minimizing setup costs vs. non delivery costs

The textual problem description (Section 2.4) states that changeovers are very expensive, considering the lost production time and the high demand. This suggests that lowering the changeover time would result in better non-delivery costs. However, the comparison of the manual planning and the calculated planning in Section 6.2 shows different results. Four times more setup time results in 32.6% less non-delivery costs. The previous assumption that lowering setup time would result in better non-delivery costs is apparently not correct. This analysis tries to analyze the relationship between setup time and non-delivery costs.

In order to make non-delivery costs and setup time comparable, the costs per setup hour have been set to representable value per hour. This hourly cost has been derived from the average daily contribution per production line. With this modified data several schedules have been calculated using different weights for non-delivery costs and setup costs. The stock deficit costs have not been weighted in this analysis, as this extra planning weight might influence the results. With these different weights, a planning time of 200 seconds, and a scheduling time of 5000 seconds schedules for the demand data of April, May and

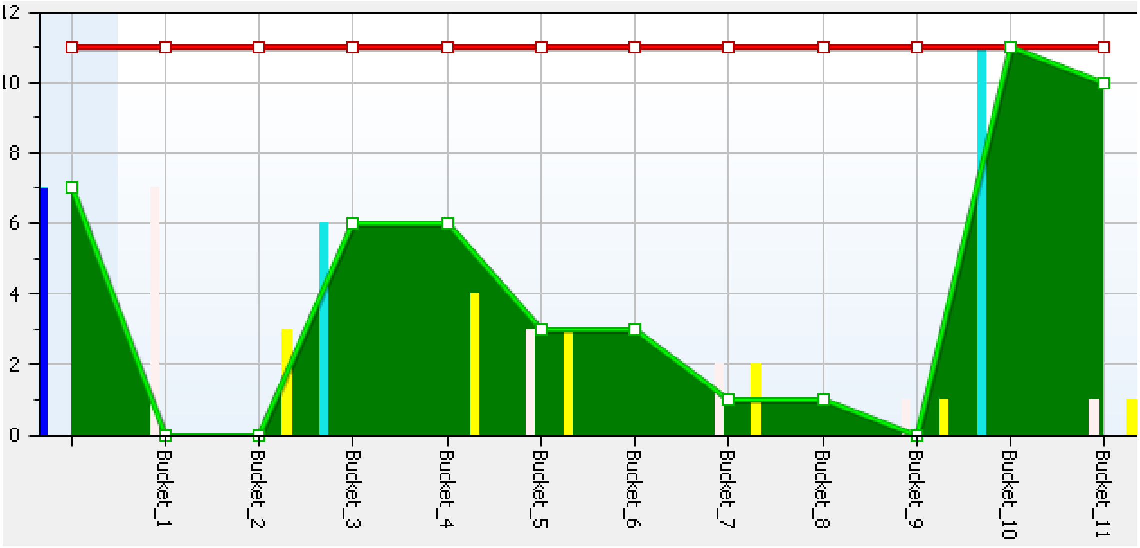

Figure 6.1: The stock coverage of product X.

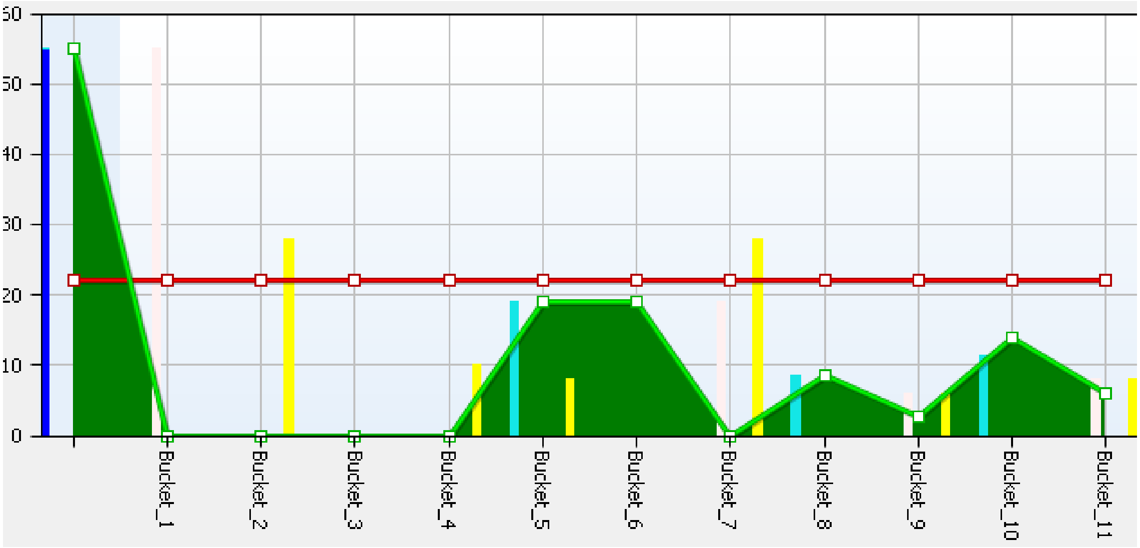

Figure 6.2: The stock coverage of product Y.

June have been calculated. Since the schedules are only compared with each other, a scheduling time of 5000 seconds is acceptable. The different non-delivery costs and setup times for these schedules are presented in Table 6.6.

The results of the analysis are a bit odd. The results do not show a real connection between the defined weights and the quality of the schedule. This defines a topic for future investigation. Some observations concerning the relation between non-delivery costs and corresponding setup times can be made. The schedules that have a low non-delivery costs have a corresponding low setup time (e.g. 12.5 - 87.5, 37.5 - 62.5, and 100 - 1), while the schedules with low setup times do not always have a corresponding low non-delivery costs (e.g. 25 - 75 vs. 62.5 - 37.5). From this we can conclude that there is some relation between setup time and non-delivery costs, but that lowering setup time does not always results in low non-delivery costs.

6.5 Impact of the initial stock

The calculated schedule solution in Section 6.2 contains many gaps and requires many changeover hours compared to the manual schedule. This makes it quite surprising that the non-delivery costs are that much better, considering the extra lost production time. The scheduling engine of *PPO* only tries to improve the non-delivery costs, since the scheduling engine can not handle the inventory deficit costs. This results in an assumed behavior where the initial stocks are being used to fulfill demands and are not replenished at the end.

To verify this assumed behavior, two analyses have been performed, using the demand data of April, May and June. In one analysis all initial stock levels have been set at zero ton, which equals to a stock deficit costs of 713 million. In the other analysis all initial stock levels have been set at their stock target, i.e. a stock deficit costs of 0. Two schedules have been calculated with a planning time of 200 seconds and a scheduling time of 5000 seconds, using the default weights. The corresponding performance indicators combined with a previous calculated schedule in Section 6.1 (5000 seconds), using the actual initial stock, are presented in Table 6.7. The difference between the non-delivery costs of the schedule with zero initial stock and the schedule with the actual initial stocks shows the impact of the initial stock on the non-delivery costs.

The manual planning shows the same behavior. When the initial stock is removed from the input data, the non-delivery costs and inventory deficit costs are approaching the costs

Weights non-delivery - setup	**Non-delivery costs** million	**Setup time** hour
1 - 100	132.5	51
12.5 - 87.5	22.8	735
25 - 75	29.9	604
37.5 - 62.5	20.4	715
50 - 50	23.8	789
62.5 - 37.5	29.8	1030
75 - 25	25.5	920
87.5 - 12.5	25.6	1119
100 - 1	22.8	697

Table 6.6: Table showing the results of the non-delivery costs vs. setup costs analysis.

of the schedule constructed with no initial stock, i.e. a non-delivery costs of 64.5 million and an inventory deficit costs of 554.4. Since the manual planning contains maintenance stops, these costs are not directly comparable.

Considering the high demand, it is obvious that the stock levels are low. Why maintain a stock that costs money, when I could sell these products? The final stocks are based on forecast data, not on actual customer orders. Uncertainties in this forecast data might result in a higher final stock than the scheduling solution indicates. Note that this uncertainty might also result in a lower final stock.

If the final stock levels turn out to be a problem, some changes could be made to the model to ensure a final stock:

- One of these changes would be to increase the production times of recipes by a certain percentage., resulting in more effective production time than *PPO* assumes. Since this lowers the production capacity, non-delivery costs will suffer from this. Another downside would be that the performance indicators would not provide any meaningful results.
- Increasing the demands of products with a certain percentage will result in more production of these products and eventually a higher stock. On the other hand will this probably lower the production of products with a lower non-delivery costs.
- Extra demand, accounting for final stocks, could be introduced. This demand should have an appropriate non-delivery costs, resulting in no interference with the real demand. Since this approach will change the non-delivery costs, the impact on the real non-delivery costs will be hard to measure.

These results show that the inventory deficit costs of the schedule with a zero initial stock is almost equal to the inventory deficit costs of the schedule with the actual initial stock. While the non-delivery costs of the schedule with the initial stock at the stock targets correspond to the non-delivery costs of the schedule with the actual initial stock. From this we can conclude that the initial stock has a large impact on the non-delivery costs. No stock results in much higher non-delivery costs. Since the stocks are not being maintained due to the limitations of the scheduling engine, this might be a problem over time.

Initial Stock	**Non-delivery costs** million	**Inventory deficit costs** million
Zero	51.8	579.8
At stock target	36.8	299.4
Actual stock	37.8	551.5

Table 6.7: Schedule quality with different initial stock levels.

Chapter 7

Evaluation

The analysis performed in Chapter 6 shows some of the possibilities of the AC model implementation in *IBM ILOG Plant PowerOps* (*PPO*). The quality of the results and the impact of the AC model will be discussed in the following sections. First the calculated schedules will be evaluated in Section 7.1. Followed by an evaluation of the model and its limitations in Section 7.2. Finally the scheduling engine of *PPO*, that calculates the schedules will be evaluated in Section 7.3.

7.1 Calculated schedules

The analysis performed in Chapter 6 provide some interesting results, especially the comparison of the manual schedule and the calculated schedule (Section 6.2) provides some nice discussion topics. The improvement in non-delivery costs sounds great, but is it? This improvement comes with a cost, i.e. more changeover hours and more product stocks below their specified stock target. Note that post processing does improve both performance indicators.

- It has been previously been assumed that lowering the changeover time would result in better product availability for sales (Section 2.4). The difference between changeover time in the manual and calculated schedule solution contradicts with this assumption. The analysis performed to see whether or not lowering setup times would result in better non-delivery costs (Section 6.4) showed that there is indeed a relation between total changeover time and non-delivery costs, but that lowering changeover time does not always improve the non-delivery costs.
- The higher inventory deficit costs in the calculated schedule are more alarming, considering the impact of the initial stock. Since the schedule engine is unable to handle inventory deficit costs, the calculated schedules only try to minimize the non-delivery costs. The production orders to improve the inventory deficit costs, suggested by the planning engine, are shifted beyond the scheduling horizon.

  Although not confirmed by comparing the final stock data (Section 6.2), the better non-delivery costs might be caused by initial stocks being used to satisfy demand and using production capacity to satisfy more other demands, while stocks are being depleted. Considering the impact of the initial stock (Section 6.5) the good results might deteriorate over time, when a steady state has been achieved.

7.2 Model

The calculated schedules have been created using the mathematical model described in Section 4.1. A model is a simplified representation of the real world, resulting in not everything being accounted for in the model. In the real world there are exceptions and other factors influencing the schedule output. Extending the model with these exceptions and factors might be unpractical or even impossible, as not everything can be specified in mathematical rules or would result in the model losing its simplicity. The analysis showed some limitations of the model:

- Technical constraints
- Modeled planning inputs
- Input data

7.2.1 Technical constraints

The comparison between the calculated schedule and manual schedule (Section 6.2) showed that the manual schedule violated some technical constraints. One violation originated from exceptions. It is not convenient to model these exceptions, resulting in more restrictions to the schedule solutions than there are in the actual planning process. This is an example that planning and scheduling remains a human task. Manual interaction on the generated schedules is an important factor in the planning process, as humans can more easily handle these exceptions and might easily see obvious improvements.

The manual schedule violated constraints, unknown by the planner. Since the planning tool allows for checking of constraints that are not taken into account by the manual planning process, this gives extra insight into what is possible and what is not. This might result in easier and better communication between planning and production.

7.2.2 Modeled planning inputs

Section 2.3.1 lists the inputs to the planning process. Not all these planning inputs are covered by the model. For instance the *resort allocation* and *open orders* are not covered by the intermediate model. With as consequence that the outputs of the planning and scheduling engines do not account for these factors. Manual adaption of the scheduling output is required to account for these extra inputs.

7.2.3 Input data

The model consists of lots of data about AC e.g. recipe data, resource data, constraint data, and demand data. If this data is not correct, the output of the scheduling calculation will not be correct either. The evaluation and comparison of the manual and calculated schedule (Section 6.2) showed some of these errors in the data.

- Resource and technical constraint data in the model have been gathered from the *constraints file*. This file, an Excel file, also contained outdated information and was not really structured. This makes the correctness of this constraint data plausible.
- Recipe data has been extracted from another Excel file. This file contained information about all products that have been produced on production lines and includes

production quantity information, i.e. tons of A-quality product per day. However, this quantity information does not involve steady state data but includes the startup period. Resulting the data being polluted by changeovers.

- The demand data is constructed from an SAP export. The SAP configuration of AC works at a different product level than most of the AC employees. SAP works at a material level, identifying every different variant of a single product. While within AC the products are identified with an aggregated identifier, combining multiple products in one. This aggregation is not present in SAP, resulting in many operations with SAP being exported to Excel. The exported demand data from SAP is combined into aggregated data using custom tools within Excel and Access. This export and manipulation of the data giving room for possible errors.
- Commercial constraint data was not clearly documented. Most of this knowledge was only in the head of the planner.

7.3 Scheduling engine

The analysis of Chapter 6 showed that the scheduling engine of *PPO* has some issues with the AC model.

- The AC model requires the inventory deficit costs performance indicator to make sure that the stocks of their products are at the required stock target. The scheduling engine can not handle this performance indicator in combination with the AC model, resulting in some strange errors from one of the internal algorithms of the scheduling engine. The absence of this performance indicator results in the scheduling engine only using the non-delivery costs performance indicator to optimize schedules. With the consequence that many of the production orders, suggested by the planning engine, affecting the inventory deficit costs are scheduled beyond the scheduling horizon.

  This problem could be tackled by introducing extra demands equal to the stock targets at the end of the schedule. However, this would require a suitable non-delivery costs for these demands, not interfering with the real demand. Since this approach will change the non-delivery costs, the impact on the real non-delivery costs will be hard to measure.

- Some special production lines allow for the simultaneous production of two products within some heavy constrained bounds. The setup feature requirement within *PPO* did not allow for a range specification, resulting in a model with extra recipes for all possible combinations. The planning engine of *PPO* does not have issues with this implementation, suggesting the right combinations to the scheduling engine. The scheduling engine on the other hand finds it hard to select the corresponding compatible recipes. Resulting in many gaps in the schedule solution, while many of these products have another variant that does match the product on the other production line. Manual post processing provides means to solve these compatibility issues.

  Sometimes the scheduling engine does select the correct compatible recipes, resulting in a much better scheduling output. However, this behavior is not consistent, which is inconvenient.

- The production orders, suggested by the planning engine, might cause compatibility or capacity issues. The scheduling engine solves these issues by introducing delays between production orders. In the cases of the special production lines this might result in a schedule covering twice as many months. The scheduling engine will not find any solution, as this extra long schedule does not fit within the available buckets. This issue has been tackled by adding extra buckets beyond the scheduling horizon.

Another solution, using the planning capacity reduction factor, has been investigated. However, the capacity required a reduction of at least 20% before the scheduling engine was able to find solutions. This amount of reduction was unacceptable and resulted in low quality scheduling outputs.

Chapter 8

Conclusion

The conclusion of this project has been split in two parts. One part contains the actual conclusion (Section 8.1) and the other part proposes future work (Section 8.2).

8.1 Conclusion

The current market situation can be characterized by a higher demand for the products produced by AC than the production capacity can handle. This was the motivation of AC to invest in improvement of its internal processes. This project covered one of these processes, i.e. the planning process. The production planning should lead to maximum effective production, optimizing the product availability for sales.

The algorithmic complexity of the planning problem of AC turned out to be *NP-complete*, resulting in an optimization approach for the planning problem of AC. This optimization approach consists of an intermediate mathematical model, containing all necessary required information about the production and planning process. This intermediate model, i.e. the AC model, has been mapped to the generic model of *IBM ILOG Plant PowerOps* (*PPO*). The AC model allowed for the automated calculation of optimized schedules. The product availability for sales, i.e. the fulfillment of the demand, is being measured using non-delivery costs. The non-delivery costs represent the money that have would been earned if the demand would have been fulfilled.

The AC model has been tested, analyzed and evaluated. An analysis comparing a calculated schedule solution with a schedule solution of the current manual planning process, showed a huge improvement of the non-delivery costs, reducing the non-delivery costs by 17%. However, the schedule engine of *PPO* was unable to optimize inventory deficit costs, resulting in a schedule only accounting for non-delivery costs. Further analyses over a much longer period of time should indicate whether or not this improvement is only applicable to the warm-up period or to the steady-state period.

Besides automated scheduling, the AC model extension of *PPO* provides an extra set of tools to the planning process. The extension provides means for the planner to see the impact of his decisions, allowing him to measure the quality of his planning solutions with the performance indicators provided by the AC model. *PPO* would give him the ability to measure the quality of a planning solution in money, rather than the number of changeovers. Ignoring the automated scheduling features of *PPO*, this ability would also provide invaluable information, i.e. support, to the manual planning process.

8.2 Future work

The conclusion (Section 8.1) states that it is unsure whether or not the scheduling solution is a real improvement or just some short term improvement. Due to limited capabilities of the scheduling engine in *PPO*, schedules are only being optimized for non-delivery costs. This observation results in the following proposed future work:

- Further long-term analysis should be conducted to find whether or not the measured improvements are part of the warm-up period or a steady-state period. If the measured results are indeed part of the warm-up period, the results should be updated with results of the steady-state period.
- The scheduling engine of *PPO* had major problems with the AC model, resulting in optimizing a scheduling solution only to non-delivery costs. The shortcoming of the scheduling engine was expected as improving the the topic of a STW proposal TU/e has submitted with AC as a partner. This graduation project was an initial step in this larger research project and confirms that additional research is required to properly solve the scheduling problem at hand.
- Due to time constraints in the project, several custom performance indicators have not been implemented. Some future work would include implementing these performance indicators. One performance indicator that measures the equal distribution of the production capacity over the resorts, using the *resort allocation*. And another performance indicator that measures the non-delivery of strategic important products, rather than abusing the non-delivery costs performance indicator for that purpose.
- Not all inputs of the planning process (Section 2.3.1) are accounted for in the AC model. The AC model could easily be extended with some of these extra inputs, e.g., considering transportation cost (boat vs. plane) and open orders.

Bibliography

[1] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

[2] Grunow, M., Gnther, H.-O., and Lehmann, M. Campaign planning for multi-stage batch processes in the chemical industry. *OR Spectrum 24* (2002), 281–314. 10.1007/s00291-002-0098-y.

[3] IBM. Information about IBM ILOG Plant PowerOps. `http://www.ibm.com/software/integration/sca/plant-powerops/about/`.

[4] Kallrath, J. Mixed integer optimization in the chemical process industry: Experience, potential and future perspectives. *Chemical Engineering Research and Design 78*, 6 (2000), 809 – 822. Process Design.

[5] Kallrath, J. Combined strategic and operational planning an milp success story in chemical industry. *OR Spectrum 24* (2002), 315–341. 10.1007/s00291-002-0102-6.

[6] Kallrath, J. Planning and scheduling in the process industry. *OR Spectrum 24* (2002), 219–250. 10.1007/s00291-002-0101-7.

[7] Kondili, E., Pantelides, C., and Sargent, R. A general algorithm for short-term scheduling of batch operations. *Computers & Chemical Engineering 17*, 2 (1993), 211 – 227. An International Journal of Computer Applications in Chemical Enginnering.

[8] Neumann, K., Schwindt, C., and Trautmann, N. Advanced production scheduling for batch plants in process industries. *OR Spectrum 24* (2002), 251–279. 10.1007/s00291-002-0100-8.