IMPROVING THE SCHEDULING ALGORITHM OF LIMIS PLANNER

MASTER THESIS INDUSTRIAL ENGINEERING AND MANAGEMENT

Student:	Roel Kikkert s0141178 Industrial Engineering and Management University of Twente
First Supervisor :	Dr. ir. J.M.J. Schutten School of Management and Governance Dept. IEBIS University of Twente
Second Supervisor:	Dr. P.C. Schuur School of Management and Governance Dept. IEBIS University of Twente
Company Supervisor:	Ir. H. ten Brug Limis

2012/2013

Preface

This report presents the results of the final project for the Production and Logistic Management track of the Master Industrial Engineering and Management at the University of Twente. This report is also an advisory report for Limis, regarding the improvement of the scheduling algorithm in Limis Planner. This research aims at developing and evaluating scheduling algorithms that could improve the current way of scheduling.

Of course, I could not have done this project without the help from others. First of all, I thank Ir. H. ten Brug as my supervisor at Limis for his ideas and support. I also thank the other colleagues of Limis for the help and interest during my research.

I thank my supervisors from the University of Twente Dr.ir. J.M.J. Schutten and Dr. P.C. Schuur for the useful comments and tips to give my research and corresponding report the appropriate structure and content.

Finally, I thank my family and friends for their interest in my proceedings and their support during my graduation project.

Roel Kikkert.

Management Summary

Limis is a small company in Enschede that offers the software package Limis Planner that gives insight in the planning of a company. The focus is on production companies with a complex production structure. Limis Planner consists of different modules that contain all aspects of planning in a production company. The objective of this research is to improve the scheduling algorithm in Limis Planner which is formulated as follows:

> ***The objective of this research is to develop an algorithm or a set of algorithms that schedules jobs in a good way in a short computation time***

We study the literature to get more insight in planning, especially in the different ways of scheduling appropriate for production companies with a job shop structure and associated aspects. There are different scheduling algorithms that could be used in a job shop structure. The current way of scheduling is based on different Priority Rules that could be set per resource and schedules the operations based on a parallel generation scheme. This algorithm gives advantages, but there are also missing functionalities.

We have drawn up the following list of requirements of the scheduling algorithm based on the advantages of Limis Planner and the missing functionalities that are based on the literature and the experiences with the use of Limis Planner:

- A schedule must be created in a short computation time.
- The algorithm should give good schedules.
- The algorithm should be consistent.
- The way of scheduling should be simple and logic.
- The algorithm should be suitable for a lot of different customers.
- It must be able to reach multiple objectives at the same time.
- The algorithm should prevent that operations with high priority and short processing time are processed after an operation with low priority and long processing time that is a few moments earlier available.
- The scheduling algorithm should deal with employees that are employable on multiple alternative resources.
- The scheduling algorithm should deal with operations that could be processed on multiple alternative resources.

Based on the literature, we evaluate different scheduling algorithms that are used for job shop scheduling. The scheduling algorithms that are compared in this research are given below with a short explanation:

- Priority Rules prioritize the operations that should be scheduled. In case of a serial generation scheme, a Priority Rule determines the next operation that should be scheduled. This operation is scheduled as early as possible. In case of a parallel generation scheme, the first start date is determined and the operation with the highest priority that can start at this date is scheduled.

- Sampling methods are multi pass procedures which mean that the best schedule is selected from several solutions. We treat three sampling methods: Random Sampling (RS), Biased Random Sampling (BRS), and Regret Based Biased Random Sampling (RBRS). RS gives each operation in the decision set the same probability, while BRS and RBRS give operations different probabilities; each operation could be selected as next one, but operations with a higher priority have more probability. RBRS uses the concept of regret for the priority values which is based on the number of regret that is created when an operation is not scheduled.
- Simulated Annealing improves a initial schedule and consists of a number of iterations. At the start of each iteration, the current schedule is compared with a randomly chosen neighbor schedule. If the neighbor schedule gives a better result, this one is accepted as the current schedule and the next iteration is started. If the neighbor schedule gives a worse solution, this schedule is accepted with a certain probability.
- Tabu Search is in many ways similar to Simulated Annealing, but the basic difference lies in the mechanism used for approving a candidate schedule. This is based on a deterministic process instead of a probabilistic one that is used in Simulated Annealing. At each iteration, the best solution from the neighbor solutions is accepted as the current solution
- Shifting Bottleneck Heuristic schedules the resources one by one by determining the bottleneck resource and schedule this resource. After that the scheduled resources are rescheduled based on release dates and precedence relations and the next bottleneck resource is scheduled as long as not all resources are scheduled.
- Constraint Satisfaction schedules operations based on constraints that are determined. If the solution meet the constraints the schedule is accepted.

We evaluates these scheduling algorithms on different criteria which is showed in Table 1.

	Priority Rules with parallel generation scheme	Priority Rules with serial generation scheme	Random Sampling	Biased Random Sampling	Regret Based Biased Random Sampling	Simulated Annealing	Tabu Search	Shifting Bottleneck Heuristic	Constraint Satisfaction
Short computation time	++	++	+	+	+	- -	- -	+	-
Good schedules	+	-	- -	-	++	++	++	+	- -
Simple and logic	++	++	++	++	+	- -	-	-	-
Suitable for a lot of different customers	++	++	-	++	++	-	-	-	-
Multiple objectives at the same time	+	+	++	++	++	++	++	+	+
High priority operations are scheduled first	+	++	- -	-	+	+	+	-	-
Employees can operate multiple alternative resources	+	+	+	+	+	+	+	+	+
Operations on multiple alternative resources	++	++	++	++	++	++	-	- -	+

Table 1

We select the Priority Rules with a parallel generation scheme, Regret Based Biased Random Sampling, and Simulated Annealing for further research based on this evaluation. We test these scheduling algorithms for two different cases.

Table 2 presents the results of Case I. The job tardiness, utilization, operation tardiness, setup time, and computation time are given as results and the objective is to minimize the job tardiness. RBRS give the best results as showed in the table.

Algorithm	Job Tardiness	Utilization	Operation Tardiness	Setup Time	Computation Time
RBRS with 1000	49.40	49.84	235.95	585.7	850.2
RBRS with 100	59.90	50.12	310.45	585.35	89.6
SA with 0.99	60.80	47.75	364.95	572.8	1162.8
SA with 0.95	64.95	48.73	380.05	582.25	228.6
EDD	69.45	49.75	365.40	586.95	1
SPT	1134.95	52.09	6190.90	584.7	1
SST	1639.85	50.32	8658.30	499.9	1

Table 2

Table 3 presents the results of Case II. The objective in this case is to minimize the sum of the job tardiness and the setup time. SA gives the best results in this case as showed in the table.

Algorithm	Objective	Job Tardiness	Utilization	Operation Tardiness	Setup Time	Computation Time
SA with 0.99	1243.7	59.85	39.05	549.8	1183.85	1390.8
SA with 0.95	1523.2	59.7	38.15	715	1463.5	277.2
RBRS with 1000	1911.45	31.45	30.60	361.55	1880	1297.8
RBRS with 100	1944.1	29.1	29.82	291.2	1915	131
EDD	2013.8	38.3	29.62	357.7	1975.5	1
SST	2438.75	1177.75	37.61	6966.65	1261	1
SPT	4067.1	2127.6	31.38	11284.05	1939.5	1

Table 3

Based on this research, we conclude that Regret Based Biased Random Sampling with a parallel generation scheme is the most suitable algorithm for Limis Planner. However, RBRS does not completely meet the requirement to reach multiple objectives at the same time as showed in Case II. RBRS can deal with multiple objectives, but it is difficult to create a priority in situations in which there are unrelated objectives. Simulated Annealing could be a solution for this problem, but the disadvantage of SA is that it is difficult to use it in the right way, so it could be used, but only when the knowledge of the algorithm is available. Otherwise, the generated schedules are not better than RBRS in that situations.

RBRS does also not completely meet the requirement that the algorithm should prevent that operations with high priority and short processing time are processed after an operation with low priority and long processing time that is a few moments earlier available. RBRS could be extended with a serial generation scheme to solve this problem, because with this way of scheduling, the high priority operations are scheduled first.

Based on this, the conclusions are as follows:

- Regret Based Biased Random Sampling with a parallel generation scheme is the most suitable scheduling algorithm for Limis Planner in the most situations.
- A serial generation scheme is most suitable for cases with a large difference in importance of jobs, so when it is important to process a few jobs on time and it does not matter when other jobs are ready.

Table of Contents

Chapter 1 Introduction

This report describes a project done at Limis in the context of a master thesis for the Production and Logistics Management track of Industrial Engineering and Management at the University of Twente.

This chapter introduces the research. It starts with the most important background information of Limis that is presented in Section 1.1. This section gives an idea what kind of company Limis is and shows more about the subject of this thesis. Section 1.2 gives an introduction to the problem and Section 1.3 presents the organization of this thesis.

1.1. Background of Limis

Limis is a small company located in Enschede that develops and implements planning software called Limis Planner. The mission of Limis is to have the best automated solution for complex planning problems in a standard software package that is profitable for clients. This mission is associated with important values of Limis which are: smart, clarity, result, and simplicity.

Limis Planner is especially suitable for production companies with a complex production structure. The current customers of Limis are production companies and also the focus for new customers is in this industry. Limis is active in different production sectors, such as metal, wood, plastic, nutrition, and discrete manufacturing.

The small scale of the company is one of the strong points of Limis. This makes direct communication within the company possible; also communication lines with users of Limis Planner are short.

There are a lot of different planning software packages that could be used for planning. Excel and Enterprise Resource Planning (ERP) are examples of often used systems. These packages work well as long as the planning is relatively simple. If the production environment is too complex to plan with Excel or ERP, Limis Planner can be used. Limis Planner can handle a complex production environment; it gives more insight in the planning and improves it.

Limis Planner gives a 24/7 up to date insight in the planning of a production process. This results in increased reliability, shorter throughput times, and less time needed for planning. Furthermore, using Limis Planner gives savings on raw material, machinery, employees, and other production costs.

Limis Planner could be used as a stand-alone package, but it is also possible to use Limis Planner as expansion of a current system for the planning. Most of the times, the customer has something in use for planning that is not functional enough for the whole planning, but could be used for delivery of input data for Limis Planner. If this is the case, Limis Planner should be linked to that system.

Planning is the heart of the company and each part of the organization should use Limis Planner to improve their results. Limis has developed the concept Smart Planning to provide clarity. The principles of Smart Planning are the following ones:

- Limis Planner should be used in each department. Customers and suppliers should also be involved in the project to get a better planning for all parties. Appendix A gives a more extensive overview of the functionality of Limis Planner for the different stakeholders. However, it is possible to start with only using the basic functionalities of Limis Planner. For example, only the planner can use it for planning.

- Improving of planning is a continuous process and not something that can be started with the press of a button. The current systems could be used for the delivery of input data for Limis Planner by linking these systems to Limis Planner. This is necessary to make the available information useful for Limis Planner. When there is no system in use, the company should be organized in such a way that the required input data for Limis Planner is available.

- The planning in the organization becomes more and more efficient and reliable.

1.2. Problem introduction

A lot of years ago, Limis has developed planning software that contains a scheduling algorithm that schedules the operations in a certain sequence on the resources. This algorithm is the basis for the planning software that is used nowadays. Limis has a lot of different customers and each customer wants other functions in Limis Planner. Therefore, the software has been extended with new functions a lot of times, but these functionalities were added one by one to solve practical problems. Limis wants to develop a new planning software package, because Limis has the idea that these functionalities could be incorporated in a more efficient way.

Limis Planner gives customers insight in their planning and corresponding aspects, but Limis has the idea that the algorithms used in Limis Planner does not give the best possible schedules. Therefore, Limis wants an algorithm in Limis Planner that gives better schedules. Retaining a short computation time is a requirement which gives the following objective of this research:

> ***The objective of this research is to develop an algorithm or a set of algorithms that schedules jobs in a good way in a short computation time***

We have used the manufacturing planning and control architecture from Zijm (2000) to determine the scope of this research. This architecture can be used as positioning framework for managerial problems in order to indicate key problem areas. There are three hierarchical levels of control and three managerial levels as shown in Appendix B. The objective of this research is to find an algorithm that improves the scheduling. Therefore, capacity planning on operational short-term level is the position in the framework.

Several research questions should be answered to achieve the research objective. First of all, the literature should be studied to get a theoretical basis for our research. We start with literature about planning in a production company to get a general overview. After that, we focus on scheduling and especially literature about job shop scheduling, because the production structure of customers of Limis consists of a job shop structure. We also need different algorithms that could be used to solve scheduling problems in job shops and in which way these algorithms could be evaluated. This results in the following research question with corresponding sub questions:

1) ***What is written in academic literature about planning and scheduling?***
    a) What is written about planning in production companies?
    b) What is written about job shop scheduling?
    c) Which scheduling algorithms could be used to solve job shop problems?
    d) In which way could these scheduling algorithms be evaluated?

The goal is to improve the way of scheduling in Limis Planner, therefore we need the current way of scheduling in Limis Planner. First of all, we want the different functionalities of Limis Planner to get a global insight in the way of working of Limis Planner. After that, the focus is on the way of scheduling in Limis Planner, so which scheduling algorithm is used. We need the advantages and missing functionalities of the current algorithm on which we can determine the requirements of the algorithm. We also want to know the characteristics of the production process of the customers of Limis, because the scheduling algorithm should fit with these. This results in the next research question with corresponding sub questions:

2) ***What is the current way of scheduling in Limis Planner?***
    a) What are the functionalities of Limis Planner?
    b) Which algorithm is used in Limis Planner for scheduling?
    c) What are the advantages of the current algorithm?
    d) What are the missing functionalities in Limis Planner?
    e) What are the characteristics of the production process of customers of Limis?
    f) What are the requirements for the scheduling algorithm in Limis Planner?

Finally, we want to know which scheduling algorithm is the best one for Limis Planner. We select different scheduling algorithms from the literature that are suitable for Limis Planner based on the requirements for the scheduling algorithm. The algorithms are tested to compare the results of the different algorithms and decide which one is most suitable for Limis Planner. This gives the last research question with corresponding sub questions:

3) ***What is the desired way of scheduling in Limis Planner?***
    a) Which scheduling algorithms are fulfilling the requirements and could be suitable for Limis Planner?
    b) Which scheduling algorithm gives the best results?

1.3. Organization of the thesis

This section presents the organization of the thesis and gives a short introduction to each chapter.

First of all, Chapter 2 presents an overview of relevant scientific literature. This literature study reviews the planning in production companies and introduces job shop scheduling. This chapter also shows different scheduling algorithms that are used for job shop problems and it presents aspects that should be evaluated to compare the selected scheduling algorithms.

Chapter 3 describes the different modules in Limis Planner and gives more information about the current scheduling algorithm. This chapter discusses the advantages and the missing functionalities of the current way of scheduling in Limis Planner. It also presents characteristics of the production process of the customers of Limis. At the end, it gives the requirements of a scheduling algorithm for

Limis Planner based on these advantages, missing functionalities, and characteristics of the production process.

After that, Chapter 4 discusses which algorithms from the literature study are suitable for Limis Planner based on the requirements from Chapter 3. Chapter 4 also presents the results of the algorithms for different cases.

Finally, Chapter 5 gives the conclusions of this research. This section also presents recommendations for further research and the applicability of this thesis.

Chapter 2 Literature overview

This chapter describes relevant literature about planning and scheduling to solve the first research question “*What is written in academic literature about planning and scheduling”*

Section 2.1 introduces the differences between planning and scheduling. Section 2.2 gives a general overview of the planning in manufacturing companies and Section 2.3 gives an introduction to scheduling, especially to job shop scheduling. Section 2.3 also presents different methods that solve job shop scheduling problems and gives more information about the evaluation of these algorithms.

2.1. Difference between planning and scheduling

There is often confusion about the differences and similarities of planning and scheduling. Originally, planning has to deal with the generation of activities to reach a goal and scheduling has to deal with assigning activities to resources and time. Nowadays, the clear separation between planning and scheduling has disappeared in practice. Both planning and scheduling deal with finding a sequence of activities to reach goals and assigning tasks to resources and the main differences are in the purpose and results.

The purpose of planning is related to the company’s long-term strategy and has as goal to create a production environment that is capable to meet the overall goals. There should be enough capacity to deal with the incoming demand, so there should be a balance between the available capacity and the acceptance of new orders and determination of delivery dates of these orders. The purpose of scheduling is to create a sequence of operations on the resources that meets the delivery dates that are determined in the planning (De Boer, 1998).

The results of planning are rough plans for a longer period of time where activities are assigned to departments, while the results of scheduling are detailed schedules for a shorter period of time where activities are assigned in a certain sequence to machines (Barták, 1999).

We clarify the difference between planning and scheduling with the hierarchical planning framework of De Boer (1998) that divides a multi-project planning in four phases. These levels are based on a strategic, tactical, and operational level as shown in **Figure 2.1**.

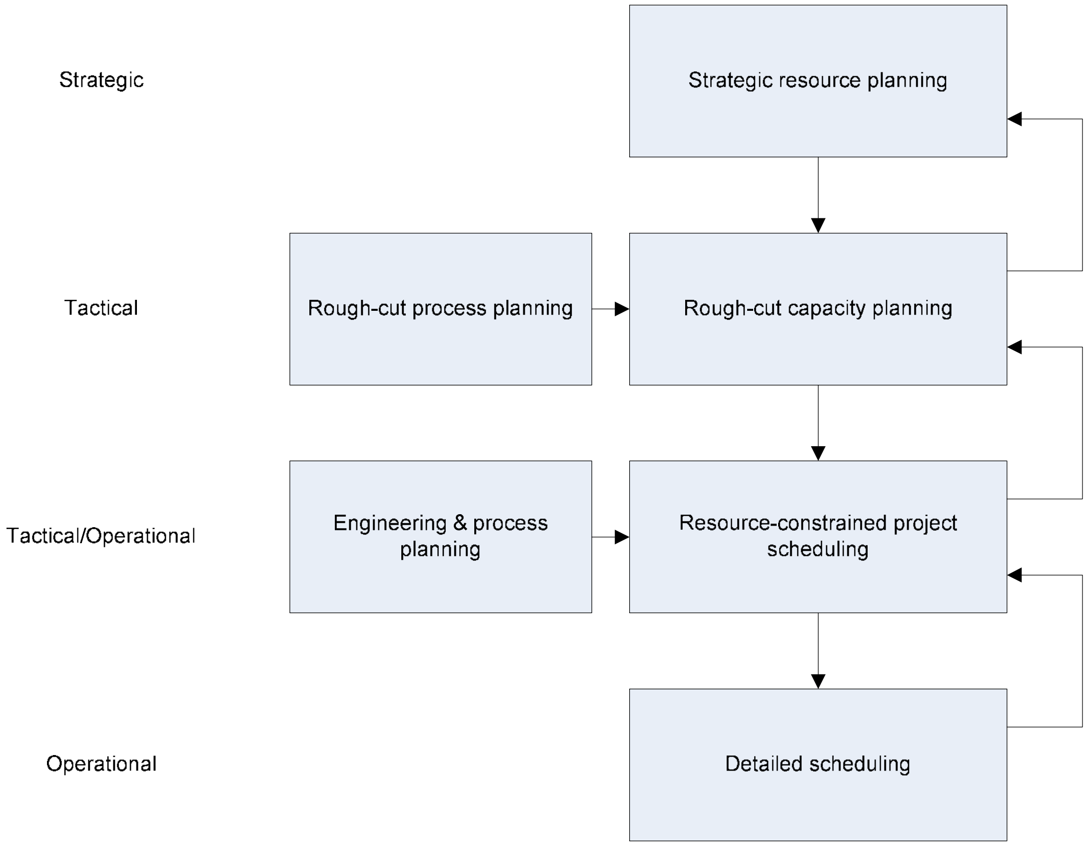


Figure 2.1 Hierarchical planning framework (De Boer, 1998)

First of all, global resource capacity levels are determined at the highest planning level that is called strategic level. Strategic decisions about resource capacity are made based on the company's long-term strategy, so the function of strategic planning is to establish a production environment that is capable to meet the company's overall goals. Usually, the planning horizon varies from one to several years (De Boer, 1998).

Tactical planning has allocating sufficient resources to deal with the incoming demand as goal. Decisions about regular and non regular capacity usage levels are made at the Rough-Cut Capacity Planning (RCCP) level. Decisions such as overtime work and outsourcing belong to this category, for example. Acceptance of new orders and determination of delivery dates are also decisions that are made in this phase. There are two RCCP methods: resource driven and time driven. Resource driven means that all resource capacity levels are fixed and the goal is to minimize the maximum lateness of the projects by using regular capacity. Time driven means that deadlines are specified for projects and the goal is to minimize the use of non regular capacity. Input for RCCP is generated by performing a rough-cut process planning. Rough-cut process planning breaks up a project into a network of work packages with estimated resource utilization and time duration. Usually, the planning horizon covers half a year or more (De Boer, 1998).

Work packages are divided into smaller activities in the next planning level, the resource-constrained project scheduling. Duration and resource utilization are determined for these activities and also precedence relations are included. Resource-constrained project scheduling determines at which moment activities are performed and assigns resource groups to these activities based on the detailed scheduling. The planning horizon varies from several weeks to several months (De Boer, 1998).

These activities are scheduled in the correct sequence during the detailed scheduling. Detailed scheduling is defined as the allocation of resources over time to perform a collection of tasks. The resulting schedules describe the sequencing and assignment of operations to resources (De Boer, 1998).

We treat an example to clarify the difference between planning and scheduling. We may have a marketing plan with the expected quantities and release times of products based on a forecast of the market and current orders. This marketing plan is for a longer period of time and serves as input for the planning of production. Production planning generates a production plan with a sequence of activities that satisfies the quantities and times from the marketing plan. The production plan consists of routes of activities and the availability of required materials. Also the production plan is for a longer period of time. Finally, the production scheduling allocates the activities from the production plan to individual resources over time. The resulting schedule is for a shorter period of time (Barták, 1999).

2.2. Planning

This section gives an overview of planning in production companies. Section 2.2.1 explains different types of production methods and Section 2.2.2 presents the manufacturing planning and control architecture of Zijm (2000).

2.2.1. Production methods

The production method of a company is relevant for planning and scheduling, because the way of planning should be dependent of the type of production in a company. Production methods are classified along two dimensions. The first criterion is the logistic product/market structure; the internal organization is used as second classification criterion (Zijm, 2000). The possible product/market structures are the following ones:

- Make and Assemble To Stock (MATS) is the production method for the majority of the suppliers of consumer products. Products are manufactured based on demand forecasts and customers are not actively concerned in the manufacturing process. These products can rapidly be delivered from inventory. Companies that produce food and electronic equipment are examples of companies with MATS manufacturing systems.

- Make To Stock, Assemble To Order (MTS/ATO) is used when a large variety of different products is made from a limited number of components. The components are produced to stock without direct interference of the customer. Only the final assembly is based on customer orders. Producing in this way prevents high final product inventories, while products could be delivered in a relatively short time. Manufacturing of cars and trucks are examples of MTS/ATO manufacturing systems.

- Make To Order (MTO) is used when small quantities of diverse end products are produced where diversity also exists on component level. The production is completely dependent on the customer order, only materials are often procured based on forecasts. Most metal working factories belong to MTO manufacturing systems.

- Engineer To Order (ETO) is used when both the product and process are specific for a customer. Design of the product is based on specifications of the customer and the customer is actively concerned in the whole process. The production of a ship is an example of an ETO manufacturing system.

The second classification dimension concerns the internal structure of the manufacturing and assembly system. There are three basic structures that might be used stand alone or combined in a more hybrid structure (Fogarty et al, 1990). The basic structures are the following ones:

- An internal structure with dedicated flow lines is the most simple structure. This could be simple processes where products are processed on one machine, but also processes where products follow a more or less common route along a number of workstations. A sufficiently large volume and a limited product variety are the primary criteria to set up a dedicated flow line. A characteristic of dedicated flow lines is that required resources are especially designed for specific processes, most of the times.

- A job shop structure is a more complex internal structure. Systems that are typically designed to manufacture a variety of products in usually small quantities, are called job shops. In a job shop, most products require an unique set-up and sequencing of processing steps. Job shops are often characterized by a functional layout and are process oriented.

- On-site manufacturing is the most complex structure and is characterized by the fact the required resources are transferred to one place instead of the other way around. On-site manufacturing relates to large projects as the realization of complex infrastructural works.

We end up with a significant variety of manufacturing system classes when the various criteria are combined and hybrid structures are included. Dedicated flow lines are commonly used in large-scale productions, so in MATS environments. Jobs shops can be often found in MTS/ATO and MTO environments. Finally, on-site manufacturing is used in ETO environments, most of the times.

2.2.2. Manufacturing planning and control architecture

The main objective of the manufacturing planning and control architecture is to give an overview of the planning and control in production companies. The emphasis in this architecture is on an integration of technological and logistics planning. Figure 2.2 depicts a general architecture for manufacturing planning and control that is divided in nine different modules.

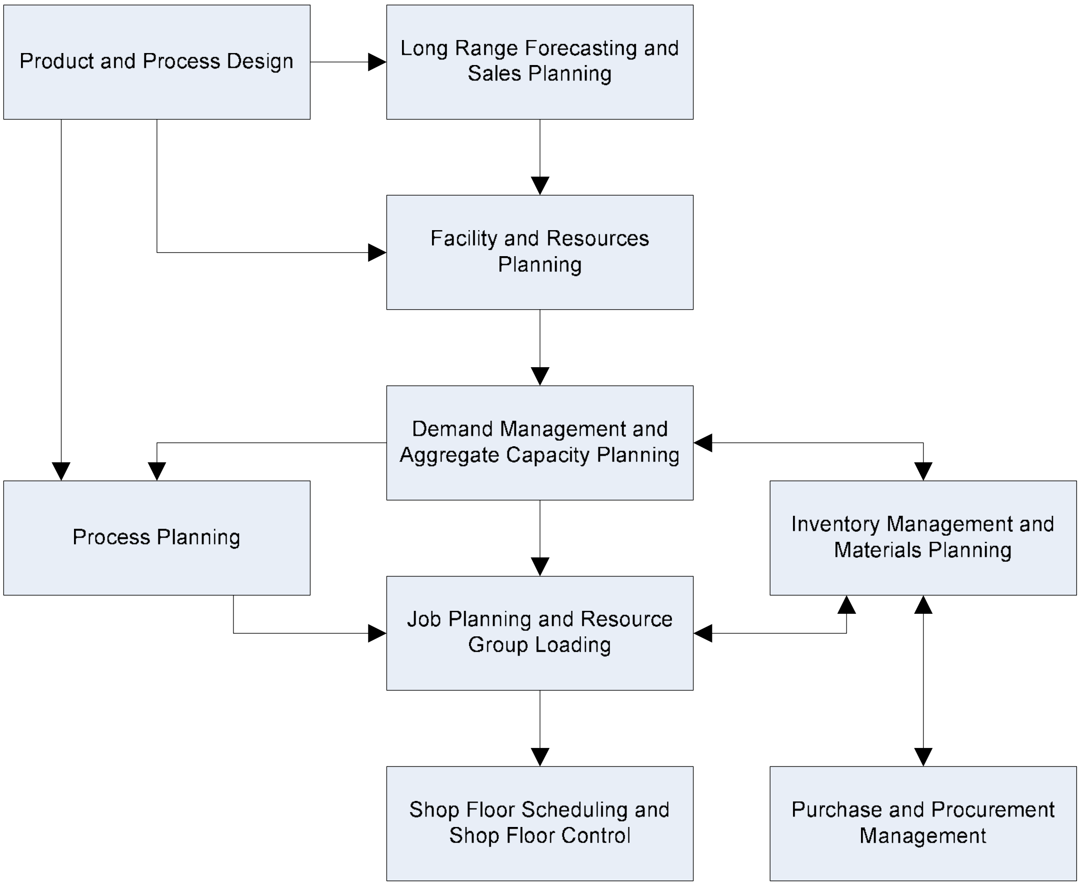


Figure 2.2 A manufacturing planning and control architecture (Zijm, 2000)

Product and process design

Product and process design should be the most important function in a company, because this is the raison d'être. Most of the times, product and process are designed simultaneously, in order to get products that are distinctive, but can be made in an efficient way. There are different methods available for product and process design.

In an MTO environment, it is important to organize design of a product and process in a way that the most steps are standardized. Postponing decisions as long as possible is a general guideline to reach this without losing flexibility during the latter process. When product and process have a lot of standardized operations, specified machines could be used, because these are suitable for more products. More universal machines should be used for specific products that are produced in small quantities. Otherwise, the machine is useless after the production of that specific product (Zijm, 2000).

Methods that could be used to standardize the production process are the assembly evaluation method (Miyakawa & Ohashi, 1986), design for assembly method (Boothroyd & Dewhurst, 1990), and the producibility and processability evaluation methods (Miyakawa, 1991) and (Takahaski et al., 1989). The goal of these methods is to simplify product and corresponding process in order to reduce costs in this way.

Long range forecasting and sales planning

This function deals with the long term estimation of the development of the market and the achievable market share in a period. It also deals with the corresponding planning of sales in that period for the different products. Regression models are often used for forecasting. These models are based on observed relations between causal factors and the realized sales volumes (Makridakis et al., 1998).

Sales planning is often based on forecasting, but in an MTO environment it is simpler, most of the times. The production of orders covers longer time periods, so forecasting is not needed, because the sales is known for the coming period. In the long run, sales in an MTO environment depends on specific customer relations for which forecasting methods have less value (Zijm, 2000).

Facility and resources planning

Required facilities and resources should be planned to produce wished sales volumes. Therefore, the technological designs and sales planning are needed as input. Required resources are also specified in this module to ensure that planned sales volumes can be realized (Zijm, 2000).

There are a lot of models that support layout planning. These models focus on minimization of costs of the flow between various departments (Kusiak & Heragu, 1987). There are also more product-focused models as cellular manufacturing that could be used for lay out planning (Wemmerlov & Hyer, 1989).

Throughput, lead time, quality, and costs are the main criteria in planning of facility and resources. At this level, closed queuing networks models could be used to evaluate the impact on the overall performance of alternative equipment and an alternative layout (Suri et al., 1993).

Demand management and aggregate capacity planning

Demand management includes short term demand forecasting, its translation into prospective orders, and order acceptance. The order acceptance function is based on aggregate capacity planning which means the synchronization of production requirements and the available resource capacities. Also the planning of additional shifts during certain periods and the decision to outsource production of some components are part of aggregate capacity planning. To generate realistic delivery dates for customers, a clear insight into the relations between available capacity of resources, possible workload and resulting lead times is essential (Zijm, 2000).

In an MTO environment, demand management is primarily related to customer order management. Costs and due dates are relevant aspects in this case. The due date should not be only based on processing time, but there should be also reserved time for technical order specification, engineering, and process planning activities. Time does not permit a detailed engineering and process planning phase before order acceptance. Therefore, order acceptance is based on the expected impact of the order on the resource utilization and capacity. Linear programming methods could be used for this (Zijm, 2000).

Process planning

Process planning specifies all technical information that is needed before processing. In process planning, a distinction is made between macro and micro. Macro process planning concerns the selection of production routes and the global estimation of processing times, while micro process planning operates at a more detailed machine level (Zijm, 2000).

Within an MATS and MTS/ATO environment, process planning is performed during the process design phase, but this is not the case in an MTO environment. In order to speed up the process planning activities, databases are needed of possible processes, machining methods, and tool combinations that could be used. There are computer aided process planning systems that automatically generate a process planning when this data is available (Zijm, 2000).

Job planning and resource group loading

When the orders have been accepted, the jobs of the orders might be planned at resource level. In this phase, the planning consists of an allocation of jobs to resource groups. Also jobs could be split in more jobs what is called lot splitting or jobs could be combined to one job what is called batching. Loading of the resources groups aims at matching the required and available capacity within each resource (Zijm, 2000).

An often used procedure is the MRP-based time-phasing procedure. This is a good procedure as long as the load might be kept relatively stable. There are different models available to get realistic lead times, these models combine capacity loading and material control in one system (Buzacott & Shanthikumar, 1993). In particular, it is easy to establish release and due dates for jobs at resource level if there is a clear overview of inventory, capacity, and internal lead times (Zijm, 2000).

Inventory management and materials planning

Inventory management plays an important role at both aggregate and detailed level. When production plans are smoothed, inventories naturally arise as temporary capacity stocks. Also producing in batches has influence on the inventory, because this means higher inventories costs. Finally, decisions about safety stocks are relevant in this module, because safety stocks leads to higher inventory costs, but also to less moments of material shortage (Zijm, 2000).

An often used system in materials planning is Material Requirements Planning (MRP). The key contribution of MRP is to ensure that materials are available for processing and to recognize the situation when this is not the case. The generalized Kanban model is a method that gives insight in the interaction between resource capacities, inventories, and lead times (Buzacott, 1989).

Purchase and procure management

The purchase and procure management function takes care of procurement of all materials purchased from suppliers. This module is related to material planning, because purchase of material is dependent of the inventory and material planning. Purchase and procure management deal with aspects of vendor selection on a strategic level, definition of purchase contracts on a mid-term tactical level, and determination of batch sizes on operational level (Zijm, 2000).

Hence the trade-off between fixed and holding costs should be made. Therefore, the economic order quantity and the time dependent equivalents can be used (Silver et al., 1998). Contracts with suppliers over long periods, specification of minimum and maximum delivery quantities, as well as reliable delivery times are opportunities to decrease the fixed costs (Zijm, 2000).

Shop floor scheduling and shop floor control
Detailed scheduling of jobs takes place on this level. The sequence of the operations is determined for individual resources and the goal is to meet the due dates that are set in the planning. Shop floor control deals with the monitoring and diagnostics of the jobs, reporting on quality aspects, and signaling major disruptions (Zijm, 2000).

Most of the times, universal machines are used for production in an MTO environment. This means that a lot of different jobs needs the same machine. Therefore, most scheduling problems are job shop scheduling problems. We treat the job shop scheduling problem extensively in Section 2.3.2. Each job has its own route in a job shop and the jobs must be sequenced on the required resources. This means a lot of different possible schedules (Zijm, 2000). We treat scheduling methods for job shop problems in Section 2.3.4.

2.3. Scheduling

Section 2.3.1 gives a general introduction to scheduling and Section 2.3.2 introduces job shop scheduling. Section 2.3.3 gives information about scheduling in practical situations and Section 2.3.4 presents different algorithms that solve scheduling problems. Finally, Section 2.3.5 gives criteria to evaluate the scheduling algorithms.

2.3.1. General introduction to scheduling

Scheduling is a decision-making process that is used in many production companies. It deals with the allocation of tasks on resources over time periods with as goal to optimize one or more objectives to obtain the best possible system performance. Machines, tools, and employees are resources in a production company and the different jobs in a production process are the tasks that should be scheduled. Examples of objectives are minimization of the completion time of the last task and the minimization of the number of tasks completed after their due date (Kempf et al., 2000).

In most cases, scheduling must interact with many other functions within a company. These interactions are typically system-dependent and take place within a computer network. However, there are also situations in which information exchange between scheduling and other decision making functions occurs in meetings or through memos. The translation of orders to jobs and the size of inventory are examples of exchangeable information between scheduling and other functions within a manufacturing company (Pinedo, 2012).

Modern factories often employ elaborate manufacturing information systems involving a computer network and various databases to exchange all information between the different departments. Local computers, workstations, and data entry terminals are connected to a central server, and may be used to retrieve data from various databases and to enter new data. In most cases, the scheduling system is filled with information in this way and the output of the scheduling system also serves as input for other systems. Large companies use Enterprise Resource Planning (ERP) systems that control and coordinate all information in their divisions and different databases. Sometimes, it does not only coordinate the own company, but also customers and suppliers (Pinedo, 2005).

Another popular system that is widely used to support scheduling is the Material Requirements Planning (MRP) system. The raw materials should be available at specified times after the set up of a schedule which should be done in conjunction with an MRP system. MRP systems contain a bill of material of each job with the required parts for production and it keeps track of the inventory of these parts. Furthermore, an MRP system determines the timing of purchases of each material. MRP systems take these decisions based on scheduling, but it also serves as input for scheduling in case of too late deliveries of material (Pinedo, 2012).

Scheduling in manufacturing systems is characterized by a variety of factors. There are a different number of resources with each different characteristics and configurations as level of automation, the type of material handling system and so on. The differences in these characteristics means different scheduling models that are related to the different production methods mentioned in Section 2.2.1 (Pinedo, 2005):

- The first class of models contains the project scheduling models. A project contains different stages and consists of a number of jobs that may be subject to precedence constraints. This means that a job cannot be performed before another job is finished. Minimizing the completion time of the last job is often the objective of a project scheduling model, so minimizing the completion time of the project.
- The second class of models includes single machine, parallel machine, and job shop models. The jobs in a single and parallel machine environment consist of one operation that might be performed on any of the available machines. A job in a job shop environment consists of multiple operations that have to be produced on multiple machines. Most of the times, the objective in these models is to minimize the completion time of the last job, the number of jobs that finished too late or the total tardiness.
- The third class of models focuses on production systems with automated material handling. A job consists of a number of operations and a material handling or conveyor system controls the movement of jobs as well as the timing of processing on the different machines. Maximizing the throughput is the objective of these models.
- The fourth class of models are the lot scheduling models that are used for a continuous production and demand. In this class, there are a variety of different products that are produced in large quantities. Minimizing the total changeover and inventory costs is the goal of these models, most of the times.

An important characteristic of a scheduling model is the machine configuration that could be divided in four basic ones (Pinedo, 2005). These basis machine configurations are mentioned below:

- A single machine model consists of one machine and all jobs should be scheduled on that machine.
- A parallel machine model consists of multiple machines that have the same functionality. All jobs could be scheduled on all machines.
- A flow shop model consists of multiple machines with different functionalities. All jobs needs multiple operations on a number of different machines in the same sequence. The machines in a flow shop are set up in series and when a job is completed on a machine, the job joins the queue at the next machine.

- A job shop model also consists of multiple machines with different functionalities and also the jobs need multiple operations on a number of different machines. The difference with the flow shop model is that not all jobs visit the same machines in the same sequence.

2.3.2. Classical job shop problem

The problem of scheduling jobs in a production company is often described as a job shop problem. A job shop consists of m machines $M_1, M_2, \dots, M_m$ and a set of n jobs $J_1, J_2, \dots, J_n$ that needs to be scheduled on these machines. Each machine is available from time 0 onwards and can process at most one job during an unit of time. Each job J_j consists of a set of operations $O_{1j}, O_{2j}, \dots, O_{n_j,j}$ where n_j gives the number of operations of job J_j. Operation O_{ij} can only be processed after the completion of operation $O_{i-1,j}$, where $i = 2, \dots, n_j$ and operation O_{1j} is available from time 0 onwards. Each operation O_{ij} needs continuous processing on machine u_{ij} during a given non-negative time p_{ij}. Most of the times, the objective is to find a schedule that minimizes the time needed to process all jobs, so minimizing the makespan (Schutten, 1998) and (Yamada & Nakano, 1997).

Often, a graph is used to represent a simple job shop problem. Each operation O_{ij} has a node v_{ij} with weight p_{ij}. The graph has also two auxiliary nodes s and t, both with weight 0. These nodes are the start and end note and are connected with respectively the first and the last operation of each job. The graph has a conjunctive arc $(v_{ij}, v_{i+1,j})$ $(i = 1, \dots, n_j - 1)$ for each pair of consecutive operations O_{ij} and $O_{i+1,j}$. The graph has two disjunctive arcs for each pair of operations O_{ij} and O_{kl} that must be processed on the same machine; these operations cannot be processed simultaneously. To get a feasible solution, one of the two disjunctive arcs should be selected between operations O_{ij} and O_{kl}, because one of these is processed before the other one. After this, the arcs that are not necessary could be removed to get more overview (Schutten, 1998) and (Yamada & Nakano, 1997).

Table 2.1 presents data of a simple example of a job shop with 3 machines and 3 jobs, where each jobs consists of 3 operations. For example, job 1 should be processed first on machine 1, then on machine 2, and finally on machine 3. The processing time of job 1 is 2, 4, and 6 on respectively machine 1, 2, and 3. Figure 2.3 shows the corresponding graph of this problem. Figure 2.4 shows how these jobs could be scheduled and Figure 2.5 shows the corresponding graph of this schedule.

J_j	u_{1j}	u_{2j}	u_{3j}	p_{1j}	p_{2j}	p_{3j}
J_1	M_1	M_2	M_3	2	4	6
J_2	M_2	M_3	M_1	3	3	5
J_3	M_3	M_1	M_2	4	5	6

Table 2.1 Data for Example 1

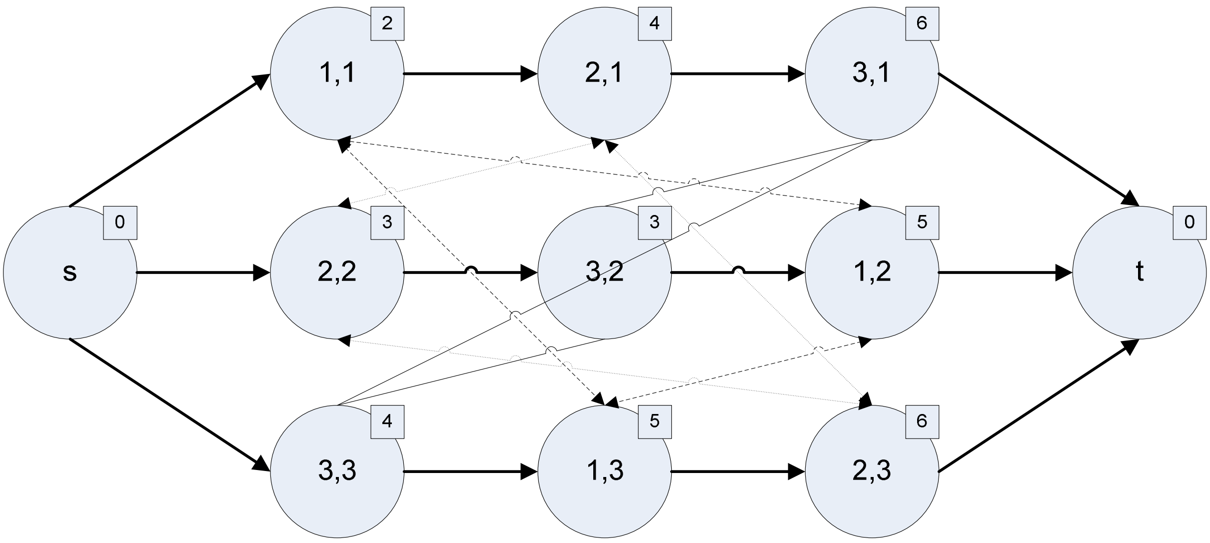


Figure 2.3 Graph representing problem of Example 1

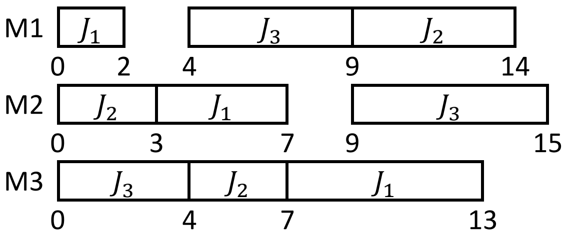


Figure 2.4 Schedule of solution Example 1

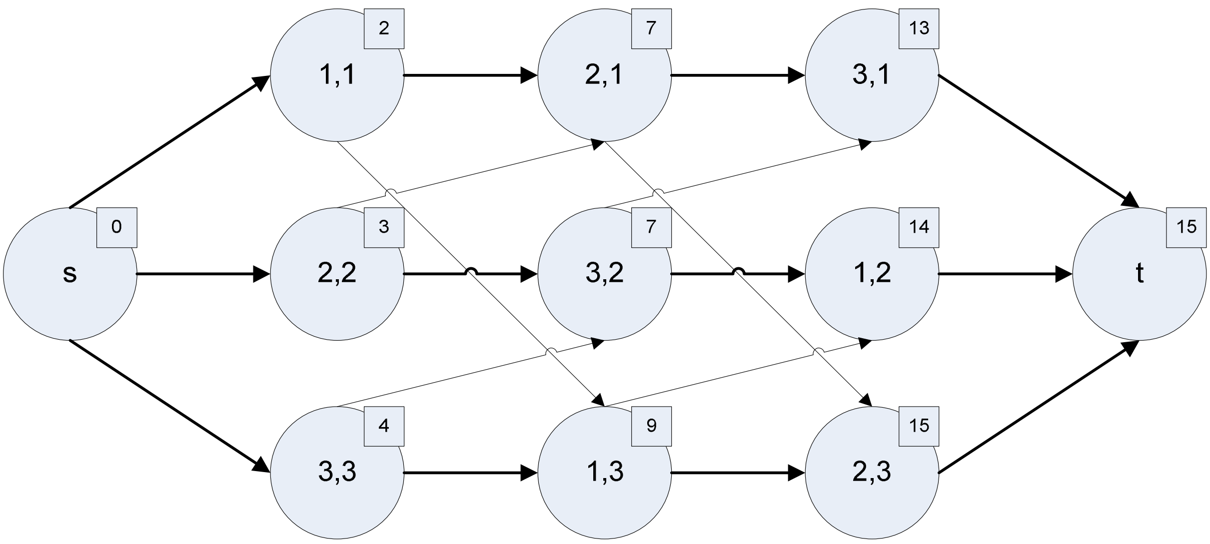


Figure 2.5 Graph representing solution Example 1

2.3.3. Scheduling in practical situations

In practice, the problems are not as simple as in the classical job shop problem presented in Section 2.3.2. In the classical job shop problem, all jobs become available for processing at the same time, but this is not always the case in practice. Therefore, each job has a release date which is the first possible date that the job could be processed (Schutten, 1996).

In the classical job shop problem, the goal is to minimize the makespan. In practice, each job has a due date that must be taken into account. The due date of a job represents the agreed completion date. Most of the times, completion after the due date is allowed at the expense of a penalty (Schutten, 1996).

Often, a machine is not directly available for a new operation after processing an operation. This could be the case for cleaning or other preparations and is called set up time. Set up times are ignored in the classical job shop problem, but are relevant in practice. The set up time is often dependent of the sequence of the operations and has influence on the duration of the production process (Pinedo, 2005).

There is also transportation time between operations to transport the product from machine to machine. Transportation time is not the same for each displacement and therefore relevant for a good way of scheduling. Transport between machines could be also carried out in batches (Schutten, 1998).

In practice, there are also machines with the same functionality. When this is the case, jobs could be processed on multiple alternative machines for the same operation which has influence on the way of scheduling (Schutten, 1998).

Most of the times, an operation does not only need a machine, but also other resources as tools and employees. There are two different approaches that deal with multiple resources. These are the centralized approach and the decentralized approach. In the centralized approach, every resource is seen as a machine that needs to be scheduled. In the decentralized approach, resources are grouped based on the fact that a group of resources may be restrictive. This could be done with a Flexible Manufacturing Cell (FMC). An FMC consists of a parallel machine group and a set of unique tools that could only be used by the machines of the FMC (Schutten, 1998) and (Meester & Zijm, 1993).

Not all machines have the same availability times. Some machines do not need employees and produce 24 hours a day. Other machines work 8 hours per day, because these machines needs employees who are only available during office hours. Sometimes, machines are also unavailable in case of maintenance. The period that a machine is not available is called down time of which are two types. We call a down time preemptive if an operation may start before and finish after it. For example, a weekend could be a preemptive down time, because it is often allowed that an operation starts on Friday afternoon and finishes on Monday morning. A down time is non-preemptive if each operation needs to be completely processed either before or after a down time. For example, maintenance is often a non-preemptive down time (Schutten, 1998).

Each job is a chain of operations in the classical job shop problem, but the job routings may be divergent or convergent in practice. For example, a convergent job routing occurs when components are assembled. A divergent job routing occurs when components are produced from one piece of raw material. The first steps are the same ones, but after separation each product has its own route. This means that it is possible to have multiple direct predecessors (Schutten, 1998).

2.3.4. Methods to solve scheduling problems

Scheduling problems could be modeled as combinatorial optimization problems in which there is a finite number of solutions. So a way to find the best solution is to enumerate all possible solutions and save the best one, but this could be only done in acceptable computation time for small problems. Due to the practical importance of combinatorial problems, many algorithms are developed to tackle these problems. These algorithms could be classified as either exact or approximate algorithms. Exact methods give an optimal solution, while approximation methods give an approximation of an optimal solution. Exact methods need much time to solve the problem and this leads to computation times that are too high for practical purposes. Therefore, the approximation methods get a lot of attention, the last years. The goal of the approximation algorithms is to find a good solution in less time instead of finding an optimal solution (Blum & Roli, 2003).

Exact optimization methods are not relevant in this research, because the job shop problems in this research are too large to solve with optimization methods. Therefore, we only threat different approximation methods.

Priority rule based scheduling

Priority rule based scheduling is an often used algorithm, because it is a fast and easy method to use. The priority rule based heuristics consists of two components, a Generation Scheme and a Priority Rule that are both mentioned in the sections below.

Generation Schemes

We distinguish two Generation Schemes: the Serial Schedule Generation Scheme (SSGS) and the Parallel Schedule Generation Scheme (PSGS). Both methods generate a feasible schedule by adding activities from a decision set that contains all schedulable activities (Kolisch, 1996).

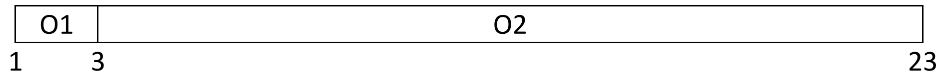
SSGS consists of N stages, where N is the number of activities to be scheduled. There are two activity sets associated with each stage: the set of scheduled activities S_n and the set of activities that are available for scheduling D_n. In each stage, an activity is selected from the decision set based on a Priority Rule and this activity is scheduled at the earliest precedence and resource feasible start time. The selected activity is removed from the decision set and added to the scheduled set. After that, a new decision set is generated and a new activity is selected from the decision set. SSGS finishes when all activities belong to the scheduled set (Kelley, 1963).

PSGS is a time oriented scheme and consists of at most N stages. In each stage, a set of activities is scheduled and each stage n has an associated schedule time t_n. Due to the schedule time t_n, the set of scheduled activities is divided into two subsets. Scheduled activities that are completed up to the schedule time are in the complete set C_n and those that are scheduled, but not completed are in the active set A_n. The decision set D_n contains all the unscheduled activities taking into account the precedence and resource constraints (Bedworth & Bailey, 1982).

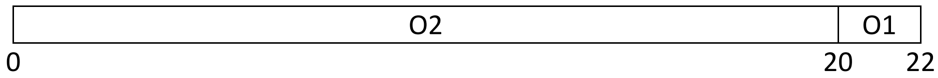
In each stage, a partial schedule is created with the activities of the complete and the active set. The schedule time in a stage is the earliest completion time of the activities in the active set of the previous stage. Each stage consists of two different steps. The first step is the determination of the schedule time. After that, the activities with a finish time equal to the new schedule time are removed from the in-process set. These activities are put into the complete set and a decision set is generated with activities. The second step is the selection of an activity based on a priority rule. This activity is scheduled at the current schedule and is removed from the available set and put into the in-process set. The second step is repeated until the available set is empty. The parallel scheme finishes when all activities are in the complete or active set (Bedworth & Bailey, 1982).

The advantage of a serial generation scheme is that operations with high priority are always scheduled first. A serial generation scheme prevents that an operation with a long processing time and low priority is scheduled before an operation with a short processing time and high priority that becomes a few minutes later available. For example, there are two operations that must be scheduled. The first one is an operation with high priority and a processing time of 2 that is available from time 1. The second one has low priority and a processing time of 20 and is available from time 0.

The serial generation scheme schedules the two operations as follows:

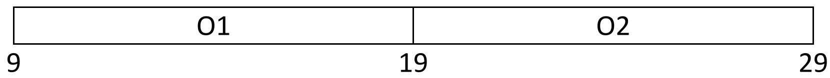


The parallel generations scheme schedules the two operations as follows:

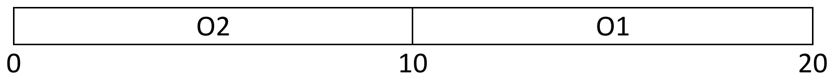


The serial method has as disadvantage that there are a lot of idle moments on the resources while this is not needed. For example, there are two operations that must be scheduled. The first one is an operation with high priority and processing time of 10 that can be scheduled on a resource from time 9, while an operation with low priority and a processing time of 10 can be scheduled from time 0.

The serial generation scheme schedules the two operations as follows:



The parallel generation scheme schedules the two operations as follows:



Neither of the schemes outperforms the other one as showed with the examples. Both schemes create the best schedules in certain cases. It is dependent of the importance of priorities which generation scheme is most suitable. The serial generation scheme is more suitable for cases with a few operations that must be processed on time and where tardiness for other operations is not a real problem, so when the results of a few operations are important. The parallel generation scheme is more suitable when the results of all operations are important, so when the total result is important. However, in general the parallel generation scheme performs better than the serial method in cases with a lot of activities (Bedworth & Bailey, 1982) & (Sprecher et al., 1995).

Priority Rules

Priority Rules prioritize the operations that should be scheduled. In case of a serial generation scheme, a Priority Rule determines the next operation that should be scheduled. This operation is scheduled as early as possible. In case of a parallel generation scheme, an operation is selected from the waiting operations when a machine is available. In this case, the machine stays only idle when there are no operations that are waiting. There are a lot of different Priority Rules. In theory, the priority of an operation could be based on everything; some well known ones are mentioned below.

The Earliest Due Date first (EDD) rule schedules operations in increasing order of due dates. So the operation with the earliest due date is scheduled as first one. The EDD rule is mostly used to minimize the maximum lateness (Pinedo, 2005).

The Earliest Release Date first (ERD) rule schedules operations in increasing order of release dates. So the operation with the earliest release date is scheduled as first one. Most of the times, the ERD rule is used to minimize the variation in the waiting times of the jobs at a machine (Pinedo, 2005).

The Minimum Slack first (MS) rule is a variation of the EDD rule. If a machine is available at time t, the remaining slack of all the operation is calculated as follows: $\max(d_j - p_j - t, 0)$, where d_j is the due date, p_j the processing time, and t the current time. The operation with the lowest slack is scheduled as first. This rule minimizes due date related objectives (Pinedo, 2005).

The Weighted Shortest Processing Time first (WSPT) rule uses the weight (w_j) and the processing time (p_j) of operations. The operation with the highest ratio of weight over processing time is processed as next one, so the operations are ordered in decreasing order of w_j/p_j. This rule could be used to minimize the weighted sum of completion times (Pinedo, 2005). When all the jobs have the same weight, the WSPT rule is reduced to the Shortest Processing Time first (SPT) rule. The SPT rule schedules operations with the shortest processing time as first ones (Dominic et al., 2004).

The Longest Processing Time first (LPT) rule orders the operations in decreasing order of processing times. So the operation with the longest processing time is processed as next one (Dominic et al., 2004). This rule balances the workload over the machines in an environment with parallel machines, because it is easier to balance the workload when there are only operations with a short processing time at the end (Pinedo, 2005).

The Least Flexible Job first (LFJ) rule is used in cases with parallel machines that are not identical and with operations that could only be produced on a specific subset of these machines. The operation that has the fewest processing alternatives is scheduled as first one (Pinedo, 2005).

The Least Work Remaining (LWR) rule gives the highest priority to an operation from a job that is almost completed. This rule could be especially used in job shops to promote the flow of the orders during the production (Dominic et al., 2004).

The Shortest Setup Time first (SST) schedules the operation with the shortest setup time first. The setup time is dependent of the last scheduled operation on the machine. This rule could be used to minimize the setup times (Pinedo, 2005).

Sampling

Sampling methods are multi pass procedures which mean that the best schedule is selected from several solutions. We treat three sampling methods: Random Sampling (RS), Biased Random Sampling (BRS), and Regret Based Biased Random Sampling (RBRS). RS gives each operation in the decision set the same probability, while BRS and RBRS give operations different probabilities; each operation could be selected as next one, but operations with a higher priority have more probability.

RBRS uses the concept of regret for the priority values which is based on the number of regret that is created when an operation is not scheduled (Kolisch, 1996).

RS is the most simple sampling method and consists of random selection of an activity from the decision set. The decision set contains all activities that could be scheduled in a feasible way. A number of possible schedules is generated and the best one of these is selected. The schedules found with RS are in general worse than the schedules found with the other sampling methods which is plausible, because the activities are randomly selected. RS is only used if other sampling methods could not be used, so in cases where priorities cannot be determined in a good way (Baker, 1974).

BRS is introduced by Cooper (1976). It also consists of a random selection of an activity from the decision set, but not all activities in the decision set have the same probability to be selected as next scheduled operation. This probability could be calculated by dividing the priority value of the activity by the sum of the priority values of all activities in the decision set which is showed in the following formula:

$$p(j) = \frac{v'(j)}{\sum_{j' \in D_n} v'(j')} \qquad (j \in D_n; v'(j) > 0)$$

The priority should be used if activities with more priority have higher values, otherwise the inverse of the priority should be used as showed in the following formula:

$$v^{'}(j) = \begin{cases} \frac{1}{v(j)} \text{ if high priority activities have lowest values} \\ v(j) \text{ if high priority activities have highest values} \end{cases} \qquad (j \in D_n)$$

Alvarez-Valdaz & Tamerit (1989) proposed a modification of BRS when high priority activities have lowest values to allow priority values equal to zero. M has to be large enough to guarantee that all modified priorities are nonnegative.

$$v^{'}(j) = \begin{cases} M - v(j) \text{ if high priority activities have lowest values} \\ v(j) \text{ if high priority activities have highest values} \end{cases} \qquad (j \in D_n)$$

RBRS is introduced by Drexl (1991). It also uses priority values, but these priority values are based on the concept of regret values. The regret value is defined as the difference between the actual value and the worst case value that might result from selecting another activity as showed in the following formula:

$$v'(j) = \begin{cases} \max V(D_n) - v(j) \text{ if high priority activities have lowest values} \\ v(j) - \min V(D_n) \text{if high priority activities have highest values} \end{cases} \qquad (j \in D_n)$$

The regret value should be modified with the following formula:

$$v''(j) = (v'(j) + \varepsilon)^{\alpha} \qquad (j \in D_n)$$

The value for ε guarantees a value unequal to zero and should be a positive value, otherwise the activities with priorities of zero could not be selected. The value for α should be also a positive one and makes it possible to reduce or increase the differences between the modified priorities. A high value of α gives more probability to schedule an operation with high priority. After that, the probability could be calculated by dividing the priority value of the activity by the sum of the priority values of all activities in the decision set as showed in the following formula:

$$p(j) = \frac{v"(j)}{\sum_{j' \in D_n} v"(j')} \qquad (j \in D_n; v(j) > 0)$$

Local Search Algorithms

These algorithms are improvement algorithms and start with a feasible initial schedule and try to obtain a better schedule by changing the current one with a simple change. A initial schedule could be created by one of the Priority Rules, for example. The exchange of two sequencing operations or the displacement of one operation to another place in the schedule are examples of simple changes. The resulting schedule is called a neighbor solution. A schedule is a neighbor of another schedule if it could be obtained through a well defined modification of the other one. At each iteration, a local search algorithm evaluates one or multiple neighbor solutions which is dependent of the algorithm. After that, the candidate solution is accepted or rejected based on a criterion which also vary per local search algorithm (Pinedo, 2005).

The design of the neighborhood is an important aspect of the local search algorithms. When there is only one machine, the neighborhood could be defined as all the schedules that could be obtained by a single adjacent pairwise change. So there are $n-1$ solutions in the neighborhood of the original schedule. A larger neighborhood is got, when a random job is taken and inserted in another position in the schedule. Then, there are $n(n-1)$ neighbors, because each job could be inserted in $n-1$ other positions. The neighborhood of a schedule in a more complicated environment is yet more complex (Pinedo, 2005).

A disadvantage of the local search methods is that they ignore a large part of the solution space in some cases, because only neighbor solutions are compared.

Simulated Annealing

Simulated Annealing (SA) is an algorithm based on a combination of ideas from the not directly related sciences combinatorial optimization and statistical physics. In combinatorial optimization, the algorithm is used as a generalization of the iterative improvement approach to combinatorial optimization problems. In statistical physics, it is used to get the state with the lowest energy level (Laarhoven et al., 1992).

The SA procedure consists of a number of iterations. At the start of each iteration, the current schedule is compared with a randomly chosen neighbor schedule. If the neighbor schedule gives a better result, this one is accepted as the current schedule and the next iteration is started. If the neighbor schedule gives a worse solution, this schedule is accepted with a certain probability (Kirkpatrick et al., 1983).

This probability is calculated with the following formula: $e^{\left(\frac{b-d}{c}\right)}$ where c is a positive control parameter which is decreased during the execution of the algorithm. This makes it possible to go to worse solutions and gives the opportunity to move away from a local optimum and find a better solution later on. The values b and d are the solution values of respectively the current schedule and the neighbor schedule. Since c decreases, the probability of moving to a worse solution in later iterations decreases. Also the probability to go to significantly worse solutions is low, because high differences between b and d are accepted with a low probability (Kirkpatrick et al., 1983).

There are several stop criteria to prevent too many iterations. First of all, the decreasing steps of c could be enlarged. Another option is to let the procedure run for a specified number of iterations or let it run until no improvement is showed for a specified number of iterations (Pinedo, 2005).

Tabu Search

Tabu Search (TS) is a metaheuristic method that is developed to deal with combinatorial optimization problems. TS is in many ways similar to SA, but the basic difference lies in the mechanism used for approving a candidate schedule. In TS, this is based on a deterministic process instead of a probabilistic one that is used in SA. At each iteration, the best solution from the neighbor solutions is accepted as the current solution (Glover, 1986).

To prevent cyclical changes, a tabu list is made with mutations that are not allowed. This could be pairs of jobs that may be not interchanged, for example. Each iteration, the tabu list is updated with the last exchange. The other ones are pushed down one place and the last one is often removed, because the tabu list has a fixed length, most of the times. It is important to give the tabu list a right number of entries. When this number is too small, cycling may occur, while the search may be unduly constrained when the number is too large (Pinedo, 2005).

A disadvantage of TS is that potentially good solutions are discarded, because they are on the tabu list. Therefore aspiration criteria could be used; these criteria overrule tabu statements and include otherwise neglected solutions. An often used aspiration criterion is allowing solutions that are better than the current best solution (Glover, 1986).

Shifting Bottleneck Heuristic

The Shifting Bottleneck Heuristic (SBH) is an approximation method that is characterized by a disjunctive graph that is used to capture the relations and dependencies between jobs on different machine groups. The nodes of the graph represent the process steps of the jobs on certain machine groups. Conjunctive arcs between the nodes are used to model the routes of a job. Disjunctive arcs are required in order to represent pending scheduling decisions among the jobs on a given machine. The processing time of the jobs are represented on the nodes (Adams et al., 1988).

In SBH, there are two sets of machines denoted by M and M_0 that are respectively the set with all machines and the set of machines that are scheduled. In each iteration, the SHB defines a schedule for another machine and this machine is added to M_0. The first step is the selection of a machine u that is not part of M_0. This bottleneck machine could be found by calculating for each operation on an unscheduled machine the earliest possible starting time and the minimal delay between the end of the operation and the end of the complete schedule based on the fixed schedules on the machines in M_0 and the conjunctive arcs. A schedule is determined for each unscheduled machine and the machine with the largest maximum lateness is the bottleneck machine. For this machine, an optimal schedule is calculated based on the fixed schedules for the machines in M_0. After that, disjunctive arcs are removed and the machines in M_0 are optimally rescheduled based on release dates en precedence relations. When this is done, the next bottleneck machine is selected (Adams et al., 1988).

Constraint Satisfaction

Constraint Satisfaction (CS) presents relations between variables in the form of constraints. It assumes n decision variables $(x_1, \dots, x_n)$ and let D_j denote the set of allowable values for decision variable x_j, so the domain (Brailsford et al.,1999).

Formally, a constraint is a mathematical relation that implies a subset S of the set $D_1 \times D_2 \times \dots \times D_n$ such that if $(x_1, \dots, x_n) \in S$, then the constraint is said to be satisfied. A mathematical function could be defined as follows: $f(x_1, \dots, x_n) = 1$ and this function only applies when the constraint is satisfied.

This gives the following constraint satisfaction problem:

$$f_i(x_1, \dots, x_n) = 1 \qquad i = 1, \dots, m$$

$$x_j \in D_j \qquad j = 1, \dots, n$$

CS is typically solved via a tree search algorithm. Each node in the search tree corresponds to a partial solution and going from one node to another one is done by assigning a value to a variable. The selection of the next variable is done by variable selection heuristics and the assignment of its value is done by value assignment heuristics (Garrido et al., 2008).

When a value is assigned to a variable, inconsistent values of unassigned variables are deleted. This is called consistency checking or constraint propagation. For a variable x, the current domain $\delta(x)$ is the set of values for which no inconsistency could be found. When a current domain is empty, because all value are removed, a so-called dead-end has been reached. In such a case, one or more assignments of variables have to be undone and alternatives have to be tried out. An instance is solved if every variable has a value and an instance is infeasible if for a variable in the root of the tree no values are remaining to be tried (Pinedo, 2005).

Most of the times, there are a lot of allowed solutions in scheduling and the best one should be selected, so therefore constraint satisfaction is not the best algorithm to use in scheduling. It could only be used when there are constraints that allow a small number of schedules.

2.3.5. Qualification of scheduling algorithms

The algorithm should be easy to use, so simplicity is the first criterion. There are two different aspects where simplicity is relevant in this case: simplicity of the scheduling algorithm itself and simplicity of using the scheduling algorithm, so the simplicity of Limis Planner.

Using the scheduling algorithm should be simple, because Limis Planner is a software package that is sold to a company and the planner of that company should work with it. Besides the planner, other personnel should understand the way of working of Limis Planner, in case of absence of the planner. So there are a lot of different people that should work with Limis Planner. This means that the way of scheduling should be easily understandable and it must be possible to learn it in a short time (Stoop & Wiers, 1996) and (Cordeau et al., 2002).

The scheduling algorithm should also be simple, because it is difficult to have a scheduling method that needs a lot of parameters with changing values for different situations. In that case, the parameters must be adapted to the situation which needs exact knowledge of the scheduling algorithm. Parameters could be used, but should be easily understandable in that case. Otherwise it is too difficult to use the algorithm in an optimal way and the possibility of inefficient schedules is too large (Stoop & Wiers, 1996) and (Cordeau et al., 2002).

Simplicity of the scheduling algorithm is also important for the marketability of Limis Planner. The goal of Limis is not to have the best scheduling algorithm, but to have the algorithm that is the best one to sell. Points that are most relevant in this way are logic and performance. A potential customer cannot judge if a given schedule gives the shortest makespan and least lateness, but they can judge the logic of a schedule. A schedule that looks logically impresses a customer more than a schedule that does not, but logicality does not always mean better results. This also applies to the scheduling algorithm; a customer is more inclined to buy Limis Planner if the algorithm schedules the jobs in a logic way (Stoop & Wiers, 1996) and (Cordeau et al., 2002).

Flexibility is another criterion that is used to evaluate scheduling algorithms. The algorithm should be able to deal with different objectives and various constraints that occur in real-life applications. It is also relevant that the scheduling algorithm can deal with changes that occur in practice. This could be disturbances as order cancellations, delayed raw materials or broken machines, but also rush orders or modifications in the appointments with customers. In that case, the scheduling method should be able to make a new schedule in short time that delivers no problems in the production process (Pinedo, 2012) and (Cordeau et al., 2002).

Also speed of the algorithm, expressed in the computation time that is needed, is an important criterion; customers want less computation time for a good schedule. Importance of computation time depends on the planning level at which the problem is solved and on the degree of accuracy required. Sometimes, a schedule is needed within a few minutes while in other cases it is allowed to have a schedule after a few days (Stoop & Wiers, 1996) and (Cordeau et al., 2002).

The scheduling algorithm should give good schedules which is also called accuracy which means the difference between the value of the created schedule and the value of the optimal schedule. The optimal value of a schedule is unknown in general, but the values of the scheduling algorithms could be compared with each other to measure which scheduling algorithm gives good schedules. There are different values that can be used to measure the accuracy of the algorithms (Cordeau et al., 2002).

Reliability is the first measure of accuracy and could be defined as the ability to deliver at the right time. Most of the times, reliability is measured with tardiness. Tardiness is the amount of time a job or an operation is finished after the due date. There are different ways to evaluate tardiness. The number of jobs or operations with tardiness, the maximum tardiness of a job or operation, or the total lateness of all jobs or all operations are examples of methods to measure the reliability of a scheduling algorithm (Cavalieri et al., 2007).

Accuracy might also be measured with the total setup time. Setup time is the amount of time needed to prepare the machine for the next operation. A high setup time suggests that a significant part of the production process is used for setups which is not efficient. Sometimes, setup times could not be prevented, but in an efficient schedule the setup time should be as low as possible (Pinedo, 2012).

The utilization of resources is also a measure of accuracy. A high utilization suggests an efficient schedule. The utilization could be measured as the utilization during the total time that resources needed to produce all jobs, but also as the utilization during the time that each resource needs to process operations on that resource (Cavalieri et al., 2007).

Transportation time is another measure of accuracy, this is the amount of time needed to transport a product from a resource to the next resource for another operation. Logically, there is time needed for transport, but these times can vary when a job could be processed on different machines or in different sequences (Pinedo, 2012).

Consistency is another evaluating criterion which is related to accuracy, As a rule, customers will prefer a scheduling algorithm that performs well all the time rather than one that may perform even better most of the times, but very poorly in some situations (Cordeau et al., 2002).

2.4. Summary

This section gives a brief summary of the literature presented in the sections above.

The production structure is relevant for the best way of planning and scheduling. There are different production structures that are divided based on the logistic product/market structure and the internal organization. This gives the following most common combinations of structures:

- Dedicated flow lines in Make and Assembly to Stock environments
- Jobs shops in Make To Stock/Assembly To Order and Make To Order environments
- On-site manufacturing in ETO environments

The manufacturing planning and control architecture of Zijm (2000) gives an overview of the planning and control in production companies and in which way the planning should be organized.

Scheduling is a decision-making process that is used in many production companies. It deals with the allocation of tasks on resources over time periods with the goal to optimize one or more objectives to obtain the best possible system performance. Most of the times, there is a job shop scheduling structure which consists of jobs that needs different machines in a different sequence.

Solution methods for scheduling problems are divided in optimal and approximation methods. Only approximation methods are relevant in this research, because the job shop scheduling problems are too large to solve with optimal methods. The investigated approximation methods for job shop scheduling are the following ones:

- Priority Rules Based Scheduling
- Sampling
- Simulated Annealing
- Tabu Search
- Shifting Bottleneck Heuristic
- Constraint Satisfaction

The different scheduling algorithms are evaluated on the following criteria:

- Simplicity
- Flexibility
- Speed
- Accuracy
- Consistency

Chapter 3 Limis Planner

The previous chapter presents literature about planning and scheduling, but the current way of scheduling should be evaluated to give advice about a new scheduling algorithm. Therefore, this chapter answers the second research question "*What is the current way of scheduling in Limis Planner".*

Section 3.1 introduces the different modules of Limis Planner and Section 3.2 describes the current way of scheduling. Section 3.3 describes the characteristics of the production process of the customers of Limis. At the end, Section 3.4 discusses the advantages and disadvantages of this way of scheduling in Limis Planner.

3.1. Modules in Limis Planner

Limis Planner is developed for production companies to get more insight in their planning. Limis Planner consists of different modules, whereby each module covers a part of the planning process. There are also modules that have a support function for the whole planning process. Section 3.1.1 shows the build-up of Limis Planner and Section 3.1.2 describes the module User Management. Section 3.1.3 presents the Work Preparation module and Section 3.1.4 gives more information about the Plant Manager. Section 3.1.5 describes the module Limis Web and Section 3.1.6 presents the module Timing. Section 3.1.7 presents the To-do-list and Section 3.1.8 gives more information about the Engine. Finally, Section 3.1.9 shows the Rapport Viewer and Section 3.1.10 presents the module Material Planning.

3.1.1. Build-up of Limis Planner

Limis Planner has been developed for manufacturing companies to get insight in the planning with minimal effort. Limis Planner consists of a capacity planner that provides a long term planning for the sales and a short term planning for the production. Besides the capacity planner, Limis Planner consists of different modules that support the production process. **Figure 3.1** shows the concept for the business process used by Limis.

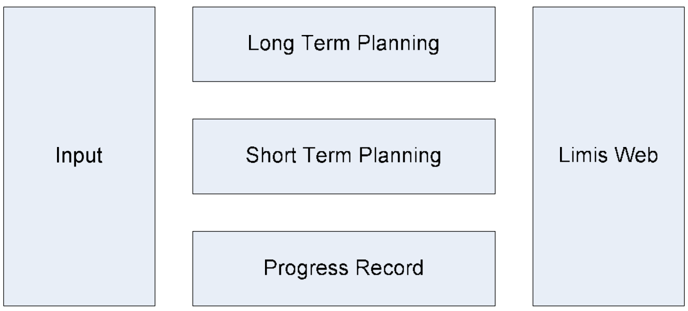


Figure 3.1 Concept business process

The long term planning contains the rough planning of orders and is based on expected processing times. The expected processing time does not exactly contain the time that is needed for processing, but also extra time that is included as buffer to take into account unpredictable events as breakdowns of machines. By means of a bucket planning, a capacity check is done and this capacity check results in expected delivery dates and serves as input for the acceptance of orders.

Short term planning contains the detailed scheduling of jobs to resources in a time window. Scheduling in Limis Planner is based on Priority Rules. It is possible to use the same rule for all machines, but using different rules is also an option. When a machine is free, an operation is picked from the operations that are waiting for processing, based on one of the preset Priority Rules. This results in detailed schedules of the production of all jobs.

The situation in practice does not always correspond with the planning, so Progress Record is used to ascertain the actual completed operations. This gives an overview about what actually happens in the production progress. It measures the efficiency of the prior estimates and whether the employees process the orders in the time as planned. Also unpredicted events as breakdowns could be registered.

The input module creates the production orders. These production orders are combined in jobs, so each job consists of one or multiple operations. There are two methods to enter the required data in the system. The first option is to enter data manually and the second option is to import data from information systems that are used in the company.

Limis Web gives employees, suppliers, and customers 24 hours per day and 7 days per week the possibility to request information about the progress of the production via Internet. Limis Web consists of a set standard web pages that can be easily added to the website of the company. These web pages give information about current orders and the planning, for instance. It is an option to give relations limited opportunities, because it is not desirable that customers can get information about suppliers, for example.

3.1.2. User Management

Limis Planner has a module User Management that gives each user the required information and rights. It is possible to give employees limited rights and in that way Limis Planner can be used in the whole company, while a few people have the rights to change things. It is also possible to work with different databases within Limis Planner which could be useful when there are production processes that are unrelated.

Regulation of relations with customers and suppliers is also part of User Management. All relevant basic information could be added in this module and there could be made a distinction between active and passive relations.

3.1.3. Work Preparation

Data that is needed before the start of the production could be entered in the Work Preparation module. New orders are created in this module with related delivery dates and also the changes in composition of jobs could be made in Work Preparation. Jobs are linked to orders, and extended with operations on the different resources. All required data can be added in this module, so the required resource and the sequence in which the jobs should be processed, for example.

This module also regulates the way of presentation, so the way in which data is showed to the user. For example, the exact machine on which the operation is processed could be shown in the To-do-list, but it is also an option to show this per machine group.

3.1.4. Plant Manager

The Plant Manager module visualizes the planning by means of a bucket planning. It shows which part of the capacity is intended for the production of the current orders and which part is available for new orders. Plant Manager presents this in a general overview and in a detailed elaboration. Expected delivery dates on operation, job, and order level can be delivered based on this and the bucket planning is also used as input for acceptance of new orders.

The general overview contains for each machine a bar that has a color dependent of the available capacity. When a machine has capacity available, the bar is green for that day and when there is more capacity needed than available, the bar is red. In the case of unavailability of machines, the bar is grey. This occurs if machines needs maintenance, for example. Plant Manager uses this way of presentation, because it shows problems in the production process in a split second.

Figure 3.2 presents an example of a general overview of the bucket planning of a workplace with three machines. The workplace has enough capacity available to fulfill the production in the first coming days. Whether the workplaces do not have capacity available, this does not mean that none of the machines in the workplace has capacity available. Therefore, Plant Manager also gives the capacity of each machine.

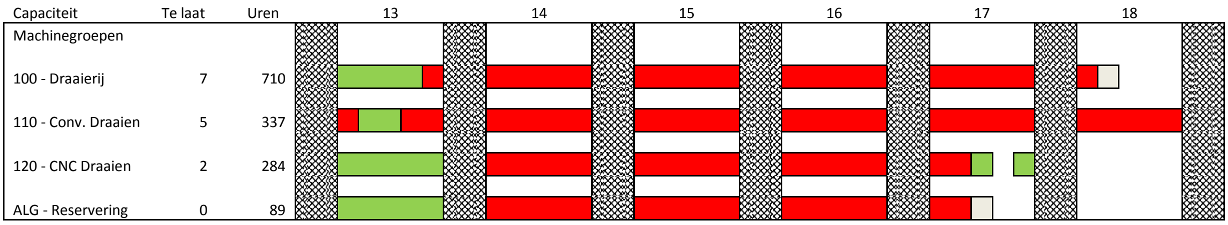


Figure 3.2 Overview capacity in Plant Manager

The detailed capacity overview gives the exact values of the bucket planning. It shows the capacity, planned capacity, and the available capacity for each machine per week. Figure 3.3 shows a screenshot of the details in the bucket planning. For example, machine 100 has a total capacity of 135 in week 13 and there are 140 hours planned this week which means that the capacity is overbooked with 5 hours. Together with the shortage of 7 hours from the week before, it gives a total overload of 12 hours.

MgNr/WpNr	Week:	<	13	14	15	16	17	18
100	Kal.:	-	135	135	135	135	135	135
Draaierij	Plan:	7	140	327	96	72	61	4
	Vrij:	-7	-12	-204	-165	-102	-28	103
100/110	Kal.:	-	55	55	55	55	55	55
Conv. Draaien	Plan:	5	86	90	49	64	36	4
	Vrij:	-5	-36	-71	-65	-74	-55	-4
100/120	Kal.:	-	60	60	60	60	60	60
CNC Draaien	Plan:	2	44	182	26	6	25	0
	Vrij:	-2	14	-108	-74	-20	15	75
100/ALG	Kal.:	-	20	20	20	20	20	20
Reservering	Plan:	0	10	55	21	3	0	0
	Vrij:	-	10	-25	-26	-9	11	31

Figure 3.3 Detailed information about capacity

Plant Manager presents the planning of an order with a Gantt Chart. The Gantt Chart shows the different planned operations with the corresponding processing times. It also shows the actual and delivery date to give insight in progress of an order. The yellow column is the actual date and the green column is the delivery date. Each bar is a different operation and the color gives information about the status of this operation. A blue bar means that the operation is planned or in processing. A red bar shows that the operation is blocked which is the case when there is not enough material available to produce this operation. The bar is grey when the operation is completed and the black bar means that the task is outsourced. The red arrows shows the expected completion date of the operation if this deviates from the planned completion date. **Figure 3.4** shows an example of a Gantt Chart.

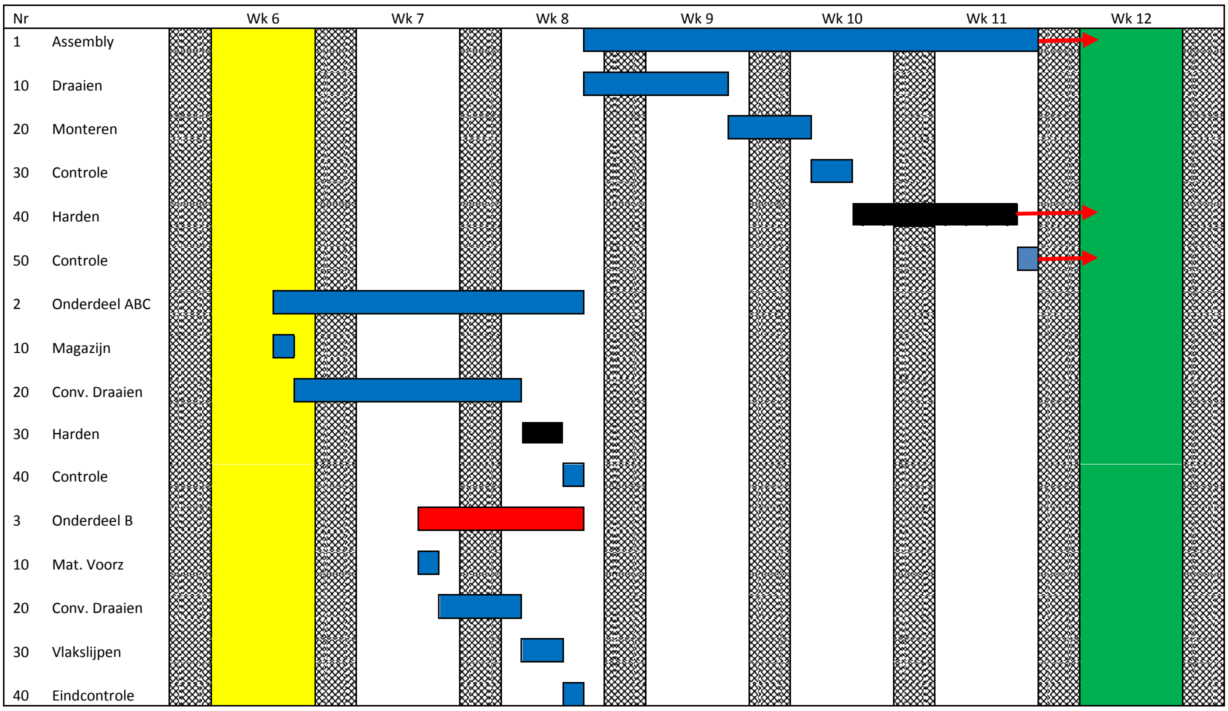


Figure 3.4 Gantt Chart of an order

3.1.5. Limis Web

Limis Web gives employees, suppliers, and customers 24 hours per day and 7 days per week the possibility to request information via Internet. Limis Web consists of a set standard web pages that could be easily added to the website of the company. This web pages give information about current orders and the planning.

3.1.6. Timing

This module manages the working hours of employees, the overtime hours and hours off. The available hours of each employee are filled in and are used as input for the planning. The working hours of the employees could be linked to jobs which gives an overview of the proceedings of an employee. This gives the opportunity to see the performance of each employee which can be used to evaluate these employees. Also the differences in progress between practice and planning are defined in this module to give insight in the quality of planning.

Timing could be done manually, each employee enters the required data in a computer and the progress is calculated based on this data. Another option is to use barcodes with corresponding scanners that can be used to register the progress. Each employee gets a barcode and a scanner, and also all jobs get barcodes. Each employee must scan his barcode on the order if the employee finished the job. In this way, the progress is calculated automatically which is more reliable than the manual way.

3.1.7. To-do-list

A To-do-list is used to make the schedule visible on the work floor. The To-do-list shows all operations with required information for the employees. It presents the information in the way it is determined in the work preparation module. In general, each employee has a To-do-list with operations that should be performed that day. The To-do-list also makes delays visible on the work floor, because operations that are scheduled after the delivery date are placed on the To-do-list of the day that is the delivery date. In this way, delays could not be overlooked.

3.1.8. Engine

The Engine contains the production model of the company. All resources and employees with corresponding capacities are defined in the engine, but also things as production hours and days are recorded in the Engine. Summarized, all data that remains the same for a long period is recorded in the Engine and should be changed in the Engine.

3.1.9. Report Viewer

This module is used to present overviews and plays an important role in almost all modules. From the Report Viewer, each overview could be saved as a separate file and opened in another software package. The Report Viewer is used to show all data in an orderly way.

3.1.10. Material Planning

The module Material Planning regulates the flow of material from and to the warehouse. This module contains, besides the picking of the goods, also the needed raw material for the production of an operation. This data could be entered manually which take a lot of time or it could be linked to a database that automatically assigns the required raw materials to an operation.

This module also checks if the required materials are available at the planned start time of operations. It shows by means of a red bar in the project overview and a cross in the detailed material planning if this is not the case. The expected inventory turnover is calculated based on the purchase of material and the planning of projects. This module also gives an ordering advice for materials, taking into account safety stocks, order sizes etc.

3.2. Current way of scheduling

The current way of scheduling is based on Priority Rules with a parallel generation scheme which means that the first possible start date is determined and the operation with the highest priority that can start at this date is scheduled. Operations that are waiting for processing on a machine are scheduled according the Priority Rules in the current way of scheduling. Several rules could be used to schedule these operations. Limis Planner uses multiple rules per machine to prevent equal scores. The different rules are ordered and are used one for one until there is an operation to schedule. It is an option to have a different order of Priority Rules per machine or work place. Scheduling in Limis Planner is based on a parallel generation scheme which means that at the moment a machine is available an operation is selected from all operations that might be scheduled on that resource as explained in Section 2.3.4.

The Highest Weight (HW) rule schedules the operation with the highest weight from the operations that are waiting for processing on that machine. Operations can have a weight from zero to nine, where operations with a weight of nine have the highest priority. The HW rule gives the opportunity to execute orders quickly through the production process which is useful in the case of rush orders.

The Earliest Release Date (ERD) rule schedules the operations based on the release date of the operations. From the waiting operations, the operation with the earliest release date is scheduled as first one. This rule is used to minimize the variation in the waiting times.

The Earliest Due Date (EDD) rule schedules operations based on due dates. When a machine is free, the operation with the earliest due date is scheduled from the waiting operations. The EDD rule is used to process according the given delivery dates, which gives higher reliability or at least the lowest deviation from the delivery date.

The Shortest Processing Time (SPT) rule schedules operations based on processing time. From the operations that are waiting, the operation with the shortest processing time is scheduled as first one. The SPT rule is used to minimize the average completion time.

The Remaining Processing Time (RPT) rule also schedules the operations based on the processing time. The difference with the SPT rule is that the processing time of the complete order is taken into account. From the waiting operations, the operation for which the job has the longest remaining processing time when that operation is completed is scheduled first. The RPT rule is also used to minimize the average completion time.

The Work In Next Queue (WINQ) rule schedules operations based on the expected waiting time of the successor of the operation. This waiting time is calculated as the sum of the processing times of the operations that are waiting at the machine on which the successor must be performed. From the waiting operations, the operation whose successor has the shortest waiting time, is scheduled first. The WINQ rule is used to maximize the utilization rates of the machines and could be used if there are expensive bottleneck machines in the company.

The Same Material Code (SMC) rule schedules operations based on the material code. Operations with the same material code are processed in sequence, because there is no setup time between these operations. From the waiting operations, an operation with the same material code is scheduled as next one. This rule is used to minimize the changeover time.

The Maximum Lateness (ML) rule schedules operations based on the current lateness of the operation. From the waiting operations, the operation with the most lateness is scheduled first. The ML rule is used to minimize the maximum lateness.

The On Time (OT) rule also schedules operations based on the lateness, but this rule schedules first the operations that could be processed on time. This rule is used to maximize the number of operations that are processed on time.

3.3. Characteristics of customers

The most relevant characteristics related to the production process of the customers of Limis are the following ones:

- The customers of Limis have a job shop structure which means that the jobs consist of multiple operations and the sequence in which a job uses the resources is different for the jobs.
- The customers works with a lot of jobs and operations.
- The jobs vary in size which means that some jobs consist of two operations while other jobs contains twenty operations, for example.
- There are large differences in the length of processing times of operations. The most operations need less than a hour, but there are operations that need multiple days.

- Employees have capabilities to operate different machines.
- There are operations that could be processed on multiple alternative resources.

3.4. Discussion

The current way of scheduling in Limis Planner has advantages, but there are also disadvantages and missing functionalities. Section 3.4.1 presents the advantages and Section 3.4.2 gives the disadvantages and corresponding missing functionalities of this way of scheduling. At the end, Section 3.4.3 presents the requirements of the new algorithm for Limis Planner.

3.4.1. Advantages of Limis Planner

Selling Limis Planner is the goal of Limis and potential customers are more likely to buy Limis Planner when the way of scheduling is logical. Priority Rules are easily understandable and schedule jobs in a simple and logic way, so simplicity and logicality are advantages of the current way of scheduling.

The current way of scheduling gives a schedule in a short computation time, so it immediately shows what changes in capacity or availability mean for progress of the production process. It also gives in large cases a schedule in a short computation time.

Limis Planner can also be easily customized to concrete problems of the customer. There are a lot of Priority Rules implemented in Limis Planner and it is possible to add new rules if the customer wants this as long as the customer has the data for these rules available. For example, operations could be sorted on product group in the case of high set up costs at a certain workplace, while the operations could be sorted on due date if lateness is the problem. The Priority Rules can be used in the same hierarchy within the whole company, but can also vary per resource or group of resources. Therefore, Priority Rules are suitable for a lot of different customers which is an advantage of Limis Planner.

The current way of scheduling can deal with employees that are employable on multiple resources and operations that could be processed on multiple alternative resources.

Summarized, the advantages of Limis Planner are the following ones:

- Simple and logic way of scheduling.
- There is a short computation time needed to create a schedule.
- Suitable for a lot of different customers.
- Can deal with employees that are employable on multiple resources.
- Can deal with operations that could be processed on multiple alternative resources.

3.4.2. Disadvantages and missing functionalities

Limis Planner uses Priority Rules that could be set per resource and Limis customizes these rules for each customer. Each resource needs one or more Priority Rules, so for each resource should be determined which rule is the best one. It is difficult to have a good flow in a production process if different Priority Rules are used, because different rules are in conflict with each other.

Another disadvantage of Limis Planner is that it does not use an objective function. It schedules the operation that is the best one for that moment, but does not take into account the final schedule. When a resource is available, an operation from the waiting ones is scheduled based on one of the Priority Rules. This gives the disadvantage that sometimes an operation with a long processing time and low priority is scheduled before an operation with high priority, because this job is ready for scheduling a few minutes later on.

Limis Planner only gives one possible schedule based on the selected Priority Rules. It makes no sense how often a Priority Rule is repeated, it always give the same solution and it is more likely that hundred solutions contains one good schedule. The Priority Rules could be set in general, per workgroup or per resource, so it is possible to give different schedules, but this should be done manually and has a completely different objective.

The Priority Rules that are used in Limis Planner has as objective to minimize one performance measure and cannot reach several objectives simultaneously. For example, it is not possible to find a the balance between tardiness and set up time, while it is probable that this gives better schedules than a minimization of one of these. Limis Planner can only try to minimize the tardiness or the set up time.

Summarized, the disadvantages of Limis Planner are the following ones:

- Using of different Priority Rules do not promote the flow.
- Limis Planner creates one schedule.
- Operations with low priority might be scheduled before operations with high priority when they are earlier available.
- Limis Planner does not use an objective function to get a good schedule.
- Limis Planner cannot reach several objectives simultaneously.

3.4.3. Conclusion

We have studied the literature about planning and scheduling and we have investigated the current functionality of Limis Planner. Based on these, we determine different requirements that the scheduling algorithm must meet. These requirements are a collection of the current advantages of Limis Planner and the most important missing functionalities. The requirements are the following ones:

- A schedule must be created in a short computation time.
- The algorithm should give good schedules.
- The algorithm should be consistent.
- The way of scheduling should be simple and logic.
- The algorithm should be suitable for a lot of different customers.
- It must be able to reach multiple objectives at the same time.
- The algorithm should prevent that operations with high priority and short processing time are processed after an operation with low priority and long processing time that is a few moments earlier available.
- The scheduling algorithm should deal with employees that are employable on multiple resources.
- The scheduling algorithm should deal with operations that could be processed on multiple resources.

Chapter 4 Solution Approach

The previous chapters give an overview of relevant literature about planning and scheduling and about the current way of scheduling. This chapter describes the decisions we have made to answer the last research question "*What is the desired way of scheduling in Limis Planner"*

Section 4.1 discusses which scheduling algorithms should be selected based on the requirements presented in Section 3.4.3 and Section 4.2 justifies the choices that are made to use the algorithms. Section 4.3 presents the cases and the results of these cases.

4.1. Discussion of algorithms

This section discusses which of the scheduling algorithms presented in Section 2.3.4 are most eligible as scheduling algorithm for Limis Planner based on the requirements that we have determined in Section 3.4.3. We give each algorithm a score from -- to ++ to evaluate in which way the algorithm meets each requirement. We cannot exactly evaluate each requirement, because scores of some requirements are dependent of the results of testing. We give these requirements an expected score based on literature. Table 4.1 shows the scores of the algorithms on the requirements and we explain the scores below the table per requirement.

	Priority Rules with parallel generation scheme	Priority Rules with serial generation scheme	Random Sampling	Biased Random Sampling	Regret Based Biased Random Sampling	Simulated Annealing	Tabu Search	Shifting Bottleneck Heuristic	Constraint Satisfaction
Short computation time	++	++	+	+	+	- -	- -	+	-
Good schedules	+	-	- -	-	++	++	++	+	- -
Simple and logic	++	++	++	++	+	- -	-	-	-
Suitable for a lot of different customers	++	++	-	++	++	-	-	-	-
Multiple objectives at the same time	+	+	++	++	++	++	++	+	+
High priority operations are scheduled first	+	++	- -	-	+	+	+	+	-
Employees can operate multiple alternative resources	+	+	+	+	+	+	+	+	+
Operations on multiple alternative resources	++	++	++	++	++	++	-	- -	+

Table 4.1 Score for different requirements

A schedule must be created in a short computation time.

The Priority Rules give schedules in the shortest computation time, because there is only one time a schedule created based on one by one selection of all operations which could be done in seconds (Pinedo, 2005).

Sampling also creates one schedule in a short computation time, because creating a schedule in works in the same way as the Priority Rules. (Regret Based) Biased Random Sampling determines a probability, but this does not significantly increase the computation time. There are more iterations needed to reach a good schedule which means a longer computation time. In general, Sampling methods find good solutions in an acceptable computation time (Kolisch, 1996).

The local search algorithms Simulated Annealing and Tabu Search need more computation time, because they improve an initial schedule by comparing the neighbor solutions. SA compares a random neighbor and TS compares all neighbor solutions. There are a lot of neighbor solutions in the case of operations that could be performed on multiple alternative resources which means a lot of computation time needed for one iteration, so in this case TS needs a lot more computation time than SA for an iteration. However there are less iterations needed to get a good solution, because the best change that is not on the tabu list is chosen instead of a random change as with SA. In general, there is a lot of computation time needed to get good schedules with Simulated Annealing and Tabu Search for large cases (Kamboj & Sebgyota, 2009).

The Shifting Bottleneck Heuristic schedules each iteration a bottleneck resource and reschedules all scheduled resources after that. Therefore, the computation time is dependent of the number of resources and the number of times the resources are rescheduled during an iteration. In general, the SBH finds good schedules in a relatively short computation time (Schutten, 1996).

Constraint Satisfaction search a schedule that meets the constraints. Therefore, the computation time is dependent of the constraints that are determined. In the most cases, CS needs a lot of computation time to find a good schedule (Barták et al., 2004).

The algorithm should give good schedules

In general, the algorithms that need more computation time also give better schedules. The Priority Rules only give a good schedule when there is an objective that corresponds with the Priority Rule that is used or when there are multiple objectives that can be combined in one Priority Rule. The parallel generation scheme gives better results for the whole collection operations, while the serial generation scheme gives better results for the operations with high priority (Bedworth & Bailey, 1982).

Regret Based Biased Random Sampling gives the best results from the Sampling methods and in general, these results are good. Only cases with multiple objectives that are unrelated give bad results, because these cannot be combined in the priority. In that case, the priority should be determined based on one of the objectives (Kolisch, 1996) and (Bedworth & Bailey, 1982).

The local search algorithms give good results when there is enough computation time available. Also the Shifting Bottleneck Heuristic gives good results in general. Especially in cases with bottleneck resources, SBH is a suitable scheduling algorithm. Constraint Satisfaction accepts an schedule that meets the constraints which is not a good one, in general (Pinedo, 2005).

The way of scheduling should be simple and logic.

The Priority Rules are the most simple scheduling algorithms. The operations are scheduled in a certain sequence that is dependent of the Priority Rule that is used which is a simple and logic way of scheduling (Pinedo, 2005).

The Sampling methods are also simple and logic, because the operations are scheduled in the same way as the Priority Rules. The only expansion is the probability that is used to schedule operations (Kolisch, 1996).

Simulated Annealing is less simple and logic than the other scheduling algorithms, because the random way of working is not intuitive. It also works with temperatures and an acceptance value which is not a simple way of scheduling (Kirkpatrick et al., 1983).

Tabu Search is also a bit more complex; selecting the best neighbor solution that is not on a tabu list is logic, but not the most obvious way of scheduling (Pinedo, 2005).

The Shifting Bottleneck Heuristic is a bit more complex, but scheduling the bottleneck resource as first one is logic; only the rescheduling process is a bit more complex (Adams et al., 1988).

Constraint Satisfaction is easily understandable, but difficult to use in scheduling, because the goal is to find a good schedule and CS search an allowed schedule. The constraints should be specific to let these schedules the same (Barták et al., 2004).

The algorithm should be suitable for a lot of different customers.

The Priority Rules are suitable for different customers. There are different Priority Rules that can be used in different situations dependent of the requirements of the customer. The Priority Rules works in all situations as long as the used Priority Rule corresponds with the objective. The Priority Rules can be used in the same way for all customers and does not need a lot of knowledge (Pinedo, 2005).

Regret Based Biased Random Sampling is also an algorithm that is suitable for a lot of customers, because the priorities can be based on an aspect or multiple aspects that are related to the objective function. RBRS works in all situations as long as the used priorities are related to the objective. RBRS does not need a lot of knowledge and can be used for different customers in the same way (Kolisch, 1996).

Simulated Annealing can be used for a lot of different customers, but there is knowledge of the algorithms needed to determine the different values that are needed. The chosen temperatures could be vary in different situations and also the neighbor structure could be vary in each situation. SA does not create good schedules when not the right neighbor structure and values are used. Therefore, it is difficult to use SA for different customers in a right way without knowledge of the algorithm (Senthiil & Selladurai, 2011) and (Kirkpatrick et al., 1983).

Tabu Search also needs knowledge to use it for different customers, because the schedules are less good without a right length of the tabu list and the ideal length can be vary in different situations. Also the determination of the neighborhood is important for the results and needs specific knowledge, so it is difficult to use TS for different customers without knowledge of the algorithm (Pinedo, 2012).

The Shifting Bottleneck Heuristic can be used for different customers, but there is also knowledge of the algorithm needed to use it for different objectives (Pinedo, 2012).

The constraints in Constraint Satisfaction can be determined in the way the customers wants, so also CS could be used for different customers. However, this is difficult, because good values for the constraints should be determined and these values are different in each situation (Barták et al., 2004).

It must be able to reach multiple objectives at the same time.
The way in which Priority Rules are suitable to reach multiple objectives is dependent of the way in which the objectives could be combined in one Priority Rule. The most general objectives could be combined in a Priority Rule, so Priority Rules are able to reach multiple objectives at the same time (Pinedo, 2005).

The Sampling methods could also deal with multiple objectives as long as the probabilities can be based on priorities that are related to the objectives. Besides, the Sampling methods use an objective value to determine which schedule of the created schedules is the best one (Kolisch, 1996).

Simulated Annealing and Tabu Search are suitable to reach multiple objectives at the same time, because these objectives can be combined in one value that is used as the objective value (Pinedo, 2005).

The Shifting Bottleneck Heuristic can deal with multiple objectives, because the way of scheduling of the bottleneck resource can be related to the objectives. Also the way of rescheduling the resources can be related to multiple objectives. The SBH is especially suitable to minimize the total makespan and less suitable for multiple objectives (Schutten, 1996).

Also Constraint Satisfaction is able to reach multiple objectives, but only by determining maximum or minimum values for the objectives in the constraints. This is difficult, because the right values should be determined for these constraints (Barták et al, 2004).

The algorithm should prevent that operations with high priority and short processing time are processed after an operation with low priority and long processing time that is a few moments earlier available.
Priority Rules with a serial generation scheme schedules the operations based on the priorities, so first the operation with the highest priority. This means that operations with a high priority are scheduled as early as possible. With a parallel generation scheme, the operation with the highest priority that can be scheduled at the first start date is scheduled (Kolisch, 1996) and (Bedworth & Bailey, 1982).

The Sampling methods give the best schedule based on an objective value, so the best schedule does probably not contain a lot of operations with high priority that are scheduled after an operation with long processing time and low priority. The Sampling methods do not allow situations in which a resource is idle while operations are available if a parallel generation scheme is used. In cases that the first operation is an operation with low priority and long processing time, this operation is always scheduled, also while there becomes a high priority operation available in a few minutes. If the Sampling methods use a serial scheme, operations with high priorities are scheduled first (Cooper, 1974).

Simulated Annealing and Tabu Search moves an operation to another place and evaluate the objective value of this schedule. When an operation with low priority and long processing time is scheduled before one with high priority and this sequence is changed, it is probable that this gives a better objective value. This means that these algorithms in the most cases prevent situations with an high priority operation after a low priority operation if the moments of availability are almost the same (Pinedo, 2012).

Constraint Satisfaction gives a schedule that satisfies the constraints, so it is dependent of the constraints whether the schedules contains high priority operations that are directly scheduled after low priority operations with long processing times (Pinedo, 2005).

The Shifting Bottleneck Heuristic schedules the operation on the bottleneck resource that is the best for the total result, so the schedule created with SBH does not contain a lot of such situations (Schutten, 1996).

The scheduling algorithm should deal with employees that are employable on multiple different resources.
All algorithms can deal with employees that are employable on multiple alternative resources (Pinedo, 2005).

The scheduling algorithm should deal with operations that could be processed on multiple alternative resources.
The Priority Rules and Sampling methods can perfectly deal with operations that could be processed on multiple alternative resources, because the operations are one by one scheduled on a resource where an available resource can be determined on that moment (Pinedo, 2005) and (Kolisch, 1996).

When operations can be processed on alternative resources, Simulated Annealing is also suitable to deal with this, because an operation can be also scheduled to another resources in a neighbor solution. Tabu Search is less suitable to deal with this, because this means a lot of neighbor solutions, so a lot of computation time that is needed to do one iteration (Pinedo, 2012).

The Shifting Bottleneck Heuristic is less suitable to deal with alternative resources, because this algorithm schedules the operations per resource, so it needs an allocation of operations to resources before the algorithm starts (Schutten, 1996).

Constraint Satisfaction can also deal with operations that could be processed on multiple resources (Pinedo, 2005).

Conclusion
Priority Rules are simple and need a small computation time. They can also deal with operations and employees that could be performed on multiple alternative resources, so we use the Priority Rules as the first scheduling algorithm in our research. In general, the parallel generation scheme gives in large cases better solutions for the whole collection of operations than the serial one. We want to give a schedule that is the best one for all operations and not only for the high priority operations, so we decide to use a parallel generation scheme.

Sampling methods are also relatively simple methods that give a schedule in short computation time when the number of iterations is small, so this algorithm meets the requirement of a small computation time. Sampling methods also use an objective value and can deal with multiple objectives at the same time. Also multiple employability of employees and operations that could be processed on multiple alternative resources give no problems with these scheduling algorithms.

Regret Based Biased Random Sampling is the best-scoring sampling method, so this is our second scheduling algorithm.

We select Simulated Annealing from the local search algorithms, because this method is more appropriate in large cases than Tabu Search. SA compares the current solution with one neighbor solution while TS compares all neighbor solutions and selects the best one taking into account the tabu list. In cases with a lot of jobs and when operations could be processed on multiple alternative resources, there are a lot of neighbor solutions. Therefore, we select Simulated Annealing as our third scheduling algorithm.

The Shifting Bottleneck Heuristic is not really simple and cannot easily deal with operations that could be performed on multiple alternative resources, because SBH is based on scheduling the resources one by one where the operations are determined to a resource. In that case there is another algorithm needed to assign operations to a resource. Therefore, the Shifting Bottleneck Heuristic is not a suitable scheduling algorithm for Limis Planner.

Constraint Satisfaction is a method that works in situations with a lot of constraints. The goal of this method is to find a solution that is feasible and not to find the best solution. In scheduling, there are typically a lot of feasible solutions, so CA is also not a suitable scheduling algorithm for Limis Planner.

4.2. Justification of choices

Section 4.2.1 explains more about the objective values. Section 4.2.2 presents more about the Priority Rules and Section 4.2.3 gives the details of Regret Based Biased Random Sampling. Finally, Section 4.2.4 presents more about Simulated Annealing.

4.2.1. Objective values

Delivering at agreed delivery times is an often used goal of scheduling, so tardiness is the first objective of this research. There are two ways of measuring the tardiness. The first option is to measure the percentage that is delivered on time and the second option is to take the total tardiness. Neither of them is always better; both give a distorted performance in some situations as shown in the following situations.

The first option evaluates a schedule with 9 too late orders, each with a lateness of 100 better as an order with 10 too late orders, each with a lateness of 1 time unit. On the other hand, the second option evaluates a schedule with 9 too late orders, each with a lateness of 1 better as a schedule with 1 order that is 10 time units too late.

We choose the second way of calculating the tardiness, because this way of calculating tardiness does not contain completely wrong situations and is more stable.

The second objective is the total setup time which is calculated as the sum of the setup times of all operations.

The last objective is the average utilization of the resources and is evaluated by calculating the ratio between the processing time that is needed to process all operations on a resource divided by the completion date of the last operation on that resource.

The objective values could be evaluated in different ways which is dependent of the production structure in the company. The following ways of using objective values are relevant for the customers of Limis and therefore for our research:

- Comparing multiple objective values in a hierarchical way is the first way of determining an objective value that could be used for customers of Limis. Multiple objectives are compared, but there is one objective that evaluates the quality of the schedule. A schedule is better if the objective value is better and worse if the objective value is worse. Only when the objective value is equal the next objective is compared. This way of comparing should be used in the cases with one main objective.

- Comparing an objective value that is based on multiple parameters is also an option that is suitable for some customers of Limis Planner. This means that the objective value consists of different parameters with each a certain weight. This way of comparing should be used in the cases with multiple main objectives.

4.2.2. Priority Rules

Figure 4.1 presents a flow chart of the way of working of the Priority Rules with a parallel generation scheme. Selecting a Priority Rule is the first step in this algorithm, so a choice should be made between the Priority Rules. We use the EDD rule, because this rule has as objective to minimize tardiness. We also use the SPT rule, because this rule has as objective to minimize the average completion time. Finally, we use the SST rule, because this rule has as goal to minimize the total setup time.

After that, the first possible start date for each operation should be determined and based on this, the start date is determined and the next operation is selected. When all operations are scheduled, the objective values are calculated and the scheduling algorithm ends, otherwise the first start date is determined again and the next operation is scheduled.

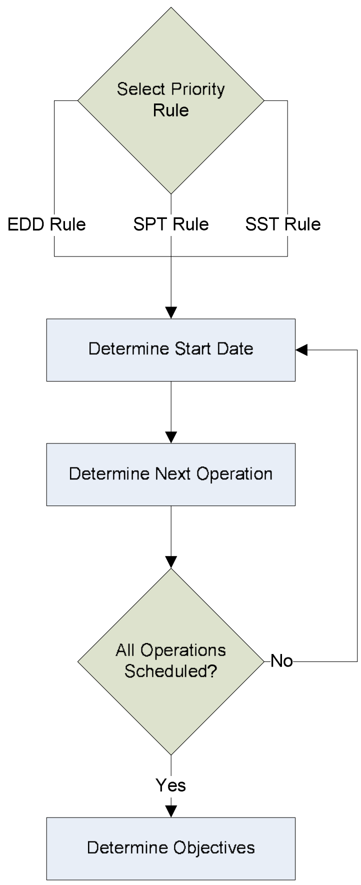


Figure 4.1 Flow Chart of Priority Rules with a parallel generation scheme

Figure 4.2 presents a flow chart of the way of working of the Priority Rules with a serial generation scheme. The difference with the parallel generation scheme is the moment of determining the next operation. Now, this is done before the determination of the start date.

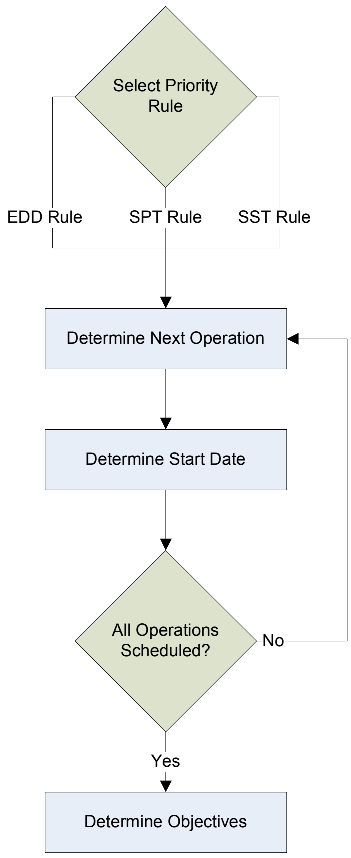


Figure 4.2 Flow Chart of Priority Rules with a serial generation scheme

4.2.3. Regret Based Biased Random Sampling

Figure 4.3 presents the flow chart of the way of working of Regret Based Biased Random Sampling.

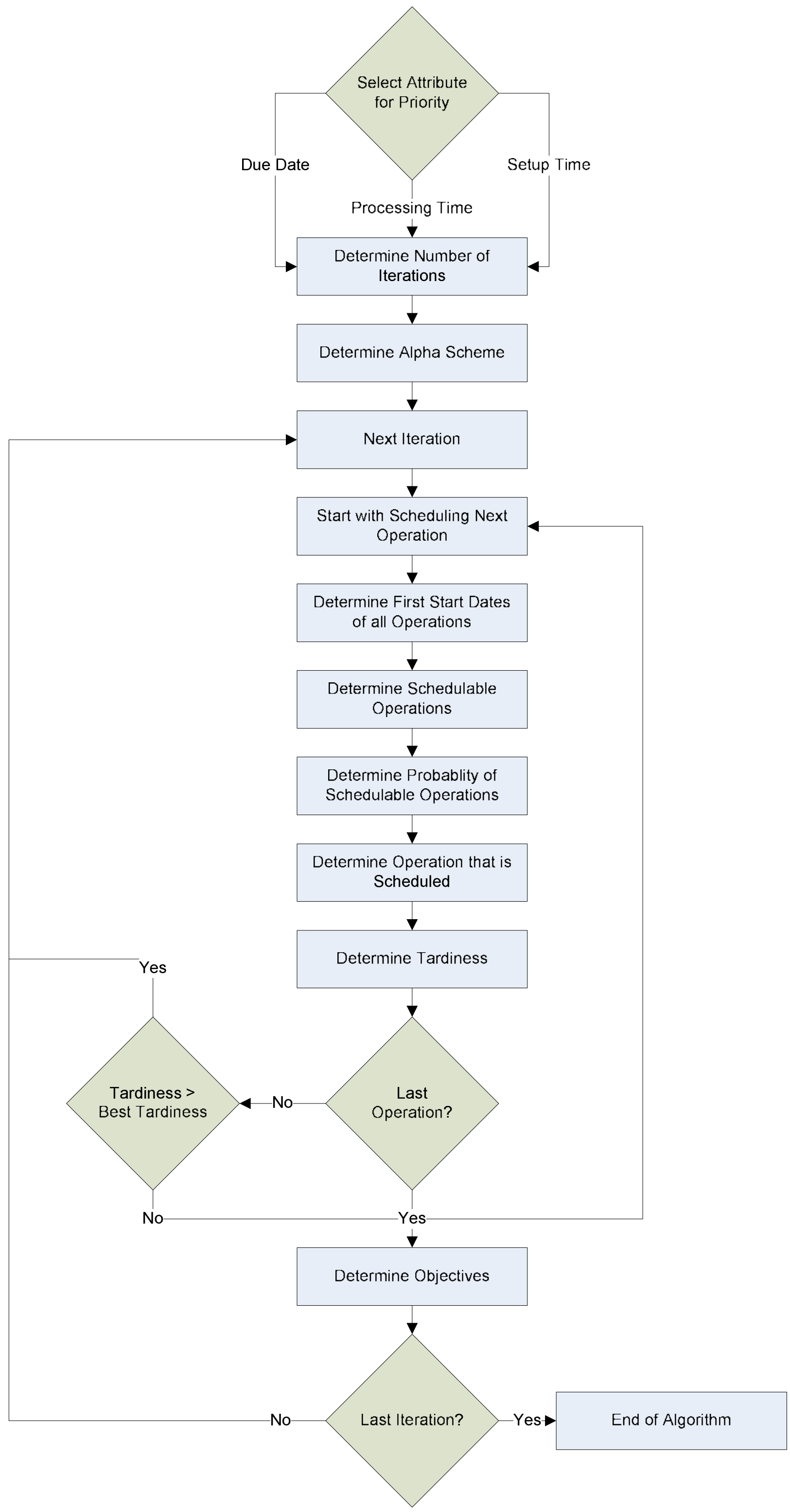


Figure 4.3 Flow Chart of Regret Based Biased Random Sampling

RBRS schedules operations based on probabilities that are dependent of the priority of an operation. We use the due date, processing time, and the setup time as possible attributes for the priority.

Before we can start with scheduling, the number of iterations should be determined. The number of iterations determines how many times there is a start made with creating a schedule. A high number of iterations means a larger computation time, but also a higher probability to find a better schedule than with a low number of iterations.

Also the alpha scheme should be determined before we can start with scheduling. We start with a high value of alpha that gives a lot of probability to an operation with high priority and we end with a value close to zero for alpha that schedules the operations in an almost random way. We use a decreasing factor to determine the values of alpha between these values. Therefore, we have to determine a start value, an end value, and a corresponding decreasing factor. The start value could be a random value that is high enough to give a large difference in probability for a small difference in priority. We use a start value for alpha of 50 and this value gives operations with a higher priority much more priority as showed in the following example:

$$v(1) = 10$$

$$v(2) = 100$$

$$v(3) = 1000$$

$$v'(1) = 1000 - 10 = 990$$

$$v'(2) = 1000 - 100 = 900$$

$$v'(3) = 1000 - 1000 = 0$$

$$v''(1) = (990 + 1)^{50} = 6.36 * 10^{149}$$

$$v''(2) = (900 + 1)^{50} = 5.45 * 10^{147}$$

$$v''(3) = (0 + 1)^{50} = 1$$

$$p(1) = \frac{6.36 * 10^{149}}{6.36 * 10^{149} + 5.45 * 10^{147} + 1} = 0{,}992$$

$$p(2) = \frac{5.45 * 10^{147}}{6.36 * 10^{149} + 5.45 * 10^{147} + 1} = 0{,}008$$

$$p(3) = \frac{1}{6.36 * 10^{149} + 5.45 * 10^{147} + 1} = 1.56 * 10^{-150}$$

We use an end value of alpha of 0.005 and this value schedules the operations almost randomly as showed with the following example.

$$v''(1) = (990 + 1)^{0.005} = 1.0351$$

$$v''(2) = (900 + 1)^{0.005} = 1.0346$$

$$v''(3) = (0 + 1)^{0.005} = 1$$

$$p(1) = \frac{1.0351}{1.0351 + 1.0346 + 1} = 0.337$$

$$p(2) = \frac{1.0346}{1.0351 + 1.0346 + 1} = 0.337$$

$$p(3) = \frac{1}{1.0351 + 1.0346 + 1} = 0.326$$

The decreasing factor should be that high that the first iteration has an alpha of 50 and the last iteration an alpha of 0,005. This means that the decreasing factor gets a value of $0.0001^{\frac{1}{n-1}}$ where n is the number of iterations.

After that, the first iteration can start and the first start dates are determined for all operations. Then should be determined which operations are schedulable which is dependent of the generation scheme. With a parallel generation scheme, all unscheduled operations with a first start date that is equal to the lowest first start date are schedulable, while with a serial generation scheme, all unscheduled operations are schedulable.

All schedulable operations get a certain probability to be the next operation that is scheduled and based on these probabilities the next operation that must be scheduled is selected. The operation is scheduled and the tardiness is calculated. The iteration is stopped when tardiness of the orders is larger than tardiness of the best solution, because the schedule cannot be better than the best one in that case. When all operations are scheduled, the objectives are calculated. After that, the next iteration is started when it is not the last one, otherwise the scheduling algorithm ends.

4.2.4. Simulated Annealing

Figure 4.4 presents the flow chart of the way of working of Simulated Annealing. SA is an improvement algorithm that needs an initial schedule to improve. We use one of the Priority Rules from Section 4.2.2 to create this schedule. The schedule and corresponding values for utilization, total job tardiness, setup time, and tardiness of the operations are saved as best solution.

We should determine the value of the start temperature. The start temperature should be high enough to accept the most changes at the begin of the algorithm. Therefore, the acceptance ratio should be close to 1, which means that the most changes are accepted. This means that most changes should have an acceptance value that is close to 1. Therefore, the formula $P_{bd} = e^{\left(\frac{b-d}{c}\right)}$ should be close to 1. The values b and d are the objective values of respectively the current schedule and the neighbor schedule and c is a positive control parameter which is decreased during the execution of the algorithm. We use a start temperature of 1000, because these value gives the most changes a high acceptance value which is showed with the following examples: $e^{\left(\frac{100-101}{1000}\right)} = 0.999$, $e^{\left(\frac{100-110}{1000}\right)} = 0.990$, $e^{\left(\frac{100-200}{1000}\right)} = 0.905$, and $e^{\left(\frac{100-1100}{1000}\right)} = 0.368$.

After that, we determine a neighbor schedule which is a schedule where two consecutive operations on a resource are changed. In cases that operations could be performed on multiple resources, an operation is sometimes placed on another resource to leave a local solution structure and to create a better schedule at the end.

The objective value of the neighbor structure is calculated based on the new start dates of all operations which are calculated as the maximum of the completion dates of the direct predecessors and the previous operation on the resource. The objective value is calculated and when this is at least as good as the current solution, it is accepted as the new current solution. Otherwise the schedule is accepted as current schedule with a certain probability. When the objective value is also better than the best solution, it is accepted as the new best solution.

This is repeated as long as the number of iterations per temperature step is smaller than the length of the Markov Chain which determines the number of neighborhood solutions that are compared per temperature step. The length of the Markov Chain should be large enough to have a significant number of iterations per temperature step compared with the number of neighbor solutions. We use a length of 100 for the Markov Chain.

When the number of iterations is equal to the length of the Markov Chain, the temperature is decreased with the decreasing factor. The decreasing factor determines the decrease of the temperature which means the number of iterations that are needed from the start to the stop temperature. We use different decreasing factors to find a balance between computation time and results.

This is repeated as long as the stop temperature is not reached. The stop temperature should be low enough to accept almost only improvements at the end of the algorithm, so the acceptance ratio should be close to 0 for deteriorations. Therefore, the acceptance value should be close to 0, which means that the formula $P_{bd} = e^{\left(\frac{b-d}{c}\right)}$ should be close to 0, so $\frac{b-d}{c}$ must be large enough. We take a stop temperature of 0.1, because this value does also not accept solutions that have an objective value that is a bit worse as showed with the following example: $e^{\left(\frac{100-101}{0.1}\right)} = 4.54 * 10^{-5}$.

When the stop temperature is reached, the results of the best schedule are given as results of this scheduling algorithm and the scheduling algorithm ends.

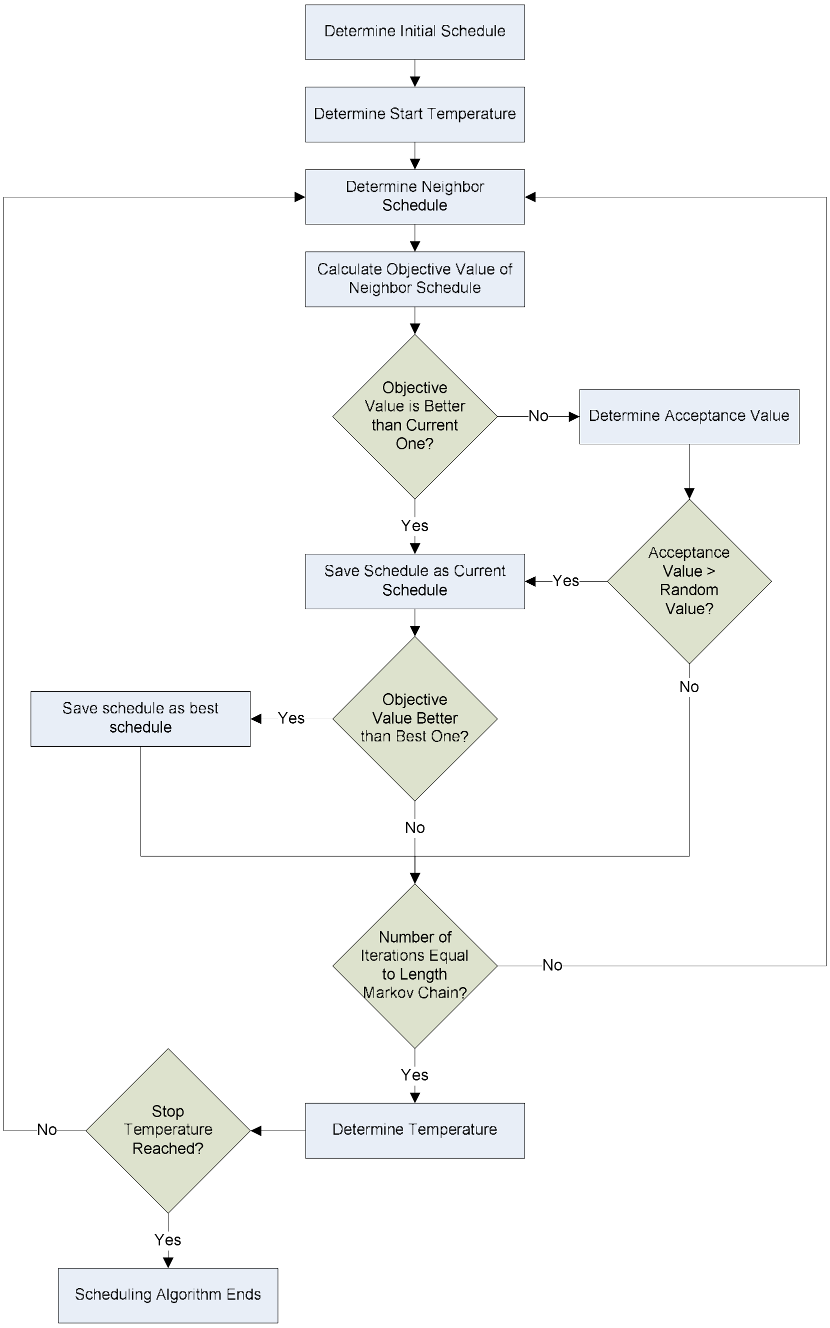


Figure 4.4 Flow Chart of Simulated Annealing

4.3. Results

We create two cases to test which scheduling algorithm is the best one for Limis Planner. The test cases contain different characteristics of the customers of Limis. We compare the scheduling algorithms for two cases with different objectives, because also the customers of Limis have different objectives. A lot of customers have minimization of tardiness as main objective, so Case I has the minimization of the tardiness of jobs as main objective and the other objectives are only relevant when the tardiness is the same. There are also customers with large setup times and in this case the sum of tardiness en setup time is the main objective. Therefore, Case II contains large setup times that are only needed between two sequenced operations with a different material code. The objective in this case is to minimize the total setup time and the tardiness of jobs. Section 4.3.1 presents the results of Case I and Section 4.3.2 shows the results of Case II.

4.3.1. Case I

We create five different situations of Case I to decrease coincidence. These situations have the same characteristics that are based on averages of customers of Limis. Case I consists of 1000 operations that must be scheduled on 20 resources and by 20 employees. Each employee operates a certain resource and each operation is processed on one resource. The operations are collected in jobs of different length with an average length of 20. The length of the job is generated by giving each operation a number between 1 and 20 and the operations with 1 are the first operations of a job. This results in lengths that vary from 1 to 38. The processing times of the operations could be divided in three categories: operations that need a few minutes which is about 50% of the operations, operations with a processing time of a few minutes to a hour which is about 30% of the operations, and operations that need more than a hour which is about 20% of the operations. Therefore, we give 50% of the operations a processing time between 1 and 5 minutes, 30% a processing time between 5 minutes and 1 hours, and 20% of the operations a processing time between 1 hour and 1 day. These processing times are generated with the help of a random number from 1 and 10 to generate random lengths of the processing times. There is setup time needed between operations if the operations does not have the same material code. Each operation has a random generated material code and there are 20 different material codes. The set up times varies from a few minutes to 1 hour which is dependent of the difference in material code. The objective in this case is to minimize the tardiness of the jobs.

The average utilization of the resources, job tardiness, total tardiness, and total setup time are presented as results of the schedules. We also give the computation time which is relevant, because this indicates how much time is needed to get these results. Table 4.2 shows the average results of the five created situations for Case I and Appendix D shows the results per situation for the different scheduling algorithms.

Algorithm	Job Tardiness	Utilization	Operation Tardiness	Setup Time	Computation Time
RBRS with 1000	49.40	49.84	235.95	585.7	850.2
RBRS with 100	59.90	50.12	310.45	585.35	89.6
SA with 0.99	60.80	47.75	364.95	572.8	1162.8
SA with 0.95	64.95	48.73	380.05	582.25	228.6
EDD	69.45	49.75	365.40	586.95	1
SPT	1134.95	52.09	6190.90	584.7	1
SST	1639.85	50.32	8658.30	499.9	1

Table 4.2 Results for Case I

Priority Rules need a small computation time, because there is only one schedule created. Priority Rules only score well on the objective that is related to the corresponding Priority Rule. The EDD rule gives a good result for tardiness, because the operations are sequenced on their due date. The SPT rule gives the highest value for the utilization which is applicable, because operations with a long processing time are processed at the end, so resources that does not have long processing time operations are ready earlier than with the other rules which give a high utilization. The schedule based on the SST rule contains the least setup time which is logic, because operations with the same material code are scheduled in sequence when this is possible and operations with the same material code does not need setup time. The prevention of setup time is not high enough to have a good result for the tardiness.

Regret Based Biased Random Sampling needs more computation time, because there are multiple schedules created. Minimizing the job tardiness is the objective in this case, so the schedule with the least job tardiness is saved as best solution. The tardiness is lower than the schedule with the EDD rule, which can be explained by the number of iterations in combination with scheduling based on probabilities. It is probable that scheduling an operation with a slightly higher due date sometimes gives better results for the tardiness of operations later in the schedule.

Also Simulated Annealing needs a lot of computation time compared to the Priority Rules, because there are a lot of changes required to get a good schedule. The improvements are reached by changes that prevent setup time and changes that decrease the tardiness. Sometimes it is better to let a resource idle for a moment, because an operation with the same material code is available a few moments later which means no setup time. Sometimes it is better to let a resource idle for an operation that is available a few moments later to decrease the tardiness or to change operations that gives less tardiness later in the schedule.

4.3.2. Case II

We also create five different situations of Case II to decrease coincidence. It also consists of 1000 operations that must be scheduled on 20 resources and by 20 employees. Each employee operates a certain resource and each operation can only produced on one resource. The operations are collected in jobs of different length with an average length of 18. The length of the job is generated in the same way as the previous case. This results in lengths that vary from 2 to 38 operations. The processing times of the operations could be divided in operations that need less than a hour and operations that have a processing time of more than a hour. Therefore we give 80% of the operations a processing time between 1 and 60 minutes and 20% of the operations a processing time between 1 hour and 1 day. The set up times varies from 1 hour to 2 hour which is dependent of the difference in material code. The objective in this case is to minimize the sum of the tardiness and the total setup time.

The same results are presented as in the previous case extended with the objective value which is the sum of the job tardiness and the setup time. Table 4.3 shows the average results of the five created situations for Case II and Appendix E shows the results per situation for the different scheduling algorithms.

Algorithm	Objective	Job Tardiness	Utilization	Operation Tardiness	Setup Time	Computation Time
SA with 0.99	1243.7	59.85	39.05	549.8	1183.85	1390.8
SA with 0.95	1523.2	59.7	38.15	715	1463.5	277.2
RBRS with 1000	1911.45	31.45	30.60	361.55	1880	1297.8
RBRS with 100	1944.1	29.1	29.82	291.2	1915	131
EDD	2013.8	38.3	29.62	357.7	1975.5	1
SST	2438.75	1177.75	37.61	6966.65	1261	1
SPT	4067.1	2127.6	31.38	11284.05	1939.5	1

Table 4.3 Results for Case II

The Priority Rules create a schedule in a short computation time, but do not give good results for the objective function. The EDD rule and the SST rule give a good score for respectively the tardiness and the total setup time. Also the utilization is high with the SST, because there is less time used for setup time, which automatically means a better ratio between processing time and total time. The SPT rule does not give good results, also the utilization is not that high which is caused by the high setup time.

Minimizing the average of the sum of the job tardiness and the setup time is the objective in this case, so the schedule with the lowest value is saved as best solution in Regret Based Biased Random Sampling. The due dates of the operations are used as priorities, so the result for tardiness is good and the setup time is not much improved, because the priority is based on the due date.

The schedule created with Simulated Annealing gives the best schedule based on the value for the objective function. This is applicable, because Simulated Annealing evaluates the schedules based on this score and can evaluate the schedule on both objectives at the same time. The setup time is lower than with the SST rule, because SA allows that a resource stays idle when there is an operation a few moments later available with the same material code which means no setup time. Also the utilization is high, because there is less time used for setups.

4.4. Conclusions

Priority Rules give good schedules in a short computation time if the Priority Rule corresponds with the objective, so the EDD rule gives good schedules when the objective is minimization of tardiness and the SMC rule gives good schedules in case of minimization of setup time.

Regret Based Biased Random Sampling needs approximately the same computation time to create a schedule, but needs more computation time for multiple iterations. Multiple iterations give better results, but a schedule that is better than the schedule created with the Priority Rules can be created in a short computation time. A disadvantage of RBRS is the way of which priorities should be determined in the case of multiple objectives that are unrelated which is showed in Case II. The priority is in this case based on one objective which gives worse results than with Simulated Annealing.

In the case of one main objective, Simulated Annealing only gives a better schedule than the initial schedule after a lot of computation time and the results are worse than RBRS. Therefore, SA is not a suitable scheduling algorithm for cases with one objective, but the results found in the case of multiple objectives are much better than found with the other algorithms, because the objectives can be easily combined in the objective value. This means that SA might be a suitable scheduling algorithm in the case of multiple unrelated objectives, because then the schedules that are created in a short computation time are also better than the schedules found with the other algorithms.

Chapter 5 Conclusions and recommendations

This chapter presents the conclusions and recommendations. Section 5.1 presents the conclusions about the different scheduling algorithms and Section 5.2 gives recommendations concerning the implementation and further research.

5.1. Conclusions

Based on this research, we conclude that Regret Based Biased Random Sampling with a parallel generation scheme is a suitable algorithm for Limis Planner in the most cases. Finding an algorithm that schedules the operations in a good sequence in an acceptable computation time is the objective of this research and RBRS schedules the operations in a good sequence in an acceptable computation time.

However, based on the requirements that are mentioned in Section 3.4.3 and the results from testing, RBRS does not completely meet the requirement to reach multiple objectives at the same time. RBRS can deal with multiple objectives, but it is difficult to create a priority in situations in which there are unrelated objectives. The priority should be based on one of these objectives which gives not the best results. Simulated Annealing could be a solution for this problem, because SA only evaluates the score of the objective value, so it does not matter when these objectives are unrelated. The disadvantage of SA is that it is difficult to use it in the right way, so it could be used, but only when the knowledge of the algorithm is available. Otherwise, the generated schedules are not better than RBRS in that situations.

RBRS with a parallel generation scheme does also not completely meet the requirement that the algorithm should prevent that operations with high priority and short processing time are processed after an operation with low priority and long processing time that is a few moments earlier available. RBRS gives the schedule with the best result for the objective function and situations in which operations with low priority are scheduled before operations with high priority have a negative effect on the objective function. Therefore, the best schedule will not contain many of these situations, but RBRS does not allow an idle resource when there are operations available. RBRS could be extended with a serial generation scheme to solve this problem, because with this way of scheduling, the high priority operations are scheduled first.

Summarized the conclusions are the following ones:

- Regret Based Biased Random Sampling with a parallel generation scheme is the most suitable scheduling algorithm for Limis Planner in the most situations.
- A serial generation scheme is most suitable for cases with a large difference in importance of jobs, so when it is important to process a few jobs on time and it does not matter when other jobs are ready.

5.2. Recommendations

Based on this research, we recommend to add Regret Based Biased Random Sampling to Limis Planner to give the best way of scheduling. In the most cases, RBRS should be used with a parallel generation scheme, because this gives the best schedules, in the most situations. Only when there is a large difference in importance of jobs, the serial generation scheme should be used.

This research could be used as theoretical basis for the development of a new version of Limis Planner, but the current Limis Planner contains a lot more functionality than used in this research. We take into account the basic principles of scheduling as predecessors, set up time and due dates, but not the more complex aspects like changing availabilities of resources and employees, for example. Therefore the proposed way of scheduling should be extended with more complex functionalities.

Each operation gets a priority that is based on one of the attributes. There must be provided on which of these attributes the priority is determined. This could be made as complicated as desired. It is possible to use more attributes that serve as input for the priority, but this can also give useless complexity.

The schedules are compared by means of one or multiple objective values, so these objective values should be derived from certain parameters. The parameters that are used as input for the objective values should be determined. We use the lateness of orders, the number of orders on time, the total lateness of the jobs, and the utilization of the resources as objectives. In theory, all objectives could be compared as long as the customer has the data available.

It is important to use the same aspects for the objective and the priority, so when the objective is to minimize the setup time, the priority of an operation should be based on something that is related to set up times and the priority should be based on due dates when the objective is to minimize the lateness, for example.

Possible attributes that could be used for the priorities and objectives are setup time, setup costs, lateness, transportation time, transportation costs, and makespan. Theoretically, all attributes could be used for the objective function and priorities, as long as the data could be made available. It is important to realize that some things are difficult to combine in the priority.

In Limis Planner, scheduling is not only used as input for the to-do-list on the work floor. It is also used to have an overview of the bottleneck resources and the further problems in the planning as on time delivery and delivery dates. Scheduling is also used as tool for the determination of the right capacities, employability of employees and outsourcing. There is not a clear distinction between planning and scheduling. Section 2.2.2 gives an overview in which way the planning should be organized in a production company. This section gives a clear distinction between planning and scheduling and this distinction should also be made within Limis Planner.

Finally, implementing Regret Based Biased Random Sampling can improve the functionality of Limis Planner. It gives Limis Planner more opportunities to create schedules in a different way for different situations.

References

Adams, J., Balas, E. & Zawack, D., 1988. Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science,* Volume 34, pp. 391-401.

Baker, K.R., 1974. *Introduction to Sequencing and Scheduling.* 1st ed. New York: Wiley.

Barták, R., 1999. *On the Boundary of Planning and Scheduling: A Study.* Manchester.

Barták, R., Salido, M.A. & Rossi, F., 2004. New Trends in Constraint Satisfaction, Planning, and Scheduling: A Survey. *The Knowledge Engineering Review,* Volume 0, pp. 1-24,

Bedworth, D.D. & Bailey, J.E., 1982. *Integrated Production Control Systems.* 1st ed. New York: John Wiley.

Blum, C. & Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys CSUR,* Volume 35, pp. 268-308.

Boothroyd, G. & Dewhurst, P., 1990. *Product Design for Assembly.* 2nd ed. Wakefield: Boothroyd Dewhurst.

Brailsford, S.C., Potts, C.N. & Smith, B.M., 1999. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research,* Volume 119, pp. 557-581.

Buzacott, J.A., 1989. *Generalized Kanban/MRP systems,* University of Waterloo: Department of Management Sciences.

Buzacott, J.A. & Shanthikumar, J.G., 1993. *Stochastic models of manufacturing systems.* 1st ed. Englewood Cliffs: Prentice-Hall.

Cavalieri, S., Terzi, S. & Macchi, M., 2007. A Benchmarking Service for the evaluation and comparison of scheduling techniques. *Computers in Industry,* Volume 58, pp. 656-666.

Cooper, D.F., 1976. Heuristics for scheduling resource-constrained projects: An experimental investigation. *Management Science,* Volume 22, pp. 186-194.

Cooper, D.F., 1977. A note on serial and parallel heuristics for resource-constrained project scheduling. *Foundations of Control Engineering*, Volume 2, pp. 131-134.

Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y. & Semet, F., 2002. A guide to vehicle routing heuristics. *Journal of the Operational Research Society,* Volume 53, pp. 512-522.

Dominic, P.D.D., Kaliyamoorthy, S. & Kumar, M., 2004. Efficient dispatching rules for dynamic job shop scheduling. *International Journal of Advanced Manufacturing Technology,* Volume 24, pp. 70-75.

Fogarty, D.W., Blackstone, J.H. & Hoffmann, T.R., 1990. *Production and inventory management.* 2nd ed. Cincinnati: South-Western Publishing Company.

Garrido, A., Arangu, M. & Onaindia, E., 2008. A constraint programming formulation for planning: from plan scheduling to plan generation. *Springer Science+Business Media,* pp. 227-256.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Computer & Operations Research,* Volume 13, pp. 533-549.

Hutchison, J. & Khumavala, B., 1990. Scheduling random flexible manufacturing systems with dynamic environments. *Journal of Operations Management,* Volume 9, pp. 335-351.

Kamboj, M.S. & Sengupta, J., 2009. Comparative Analysis of Simulated Annealing and Tabu Search Channel Allocation Algorithms. *International Journal of Computer Theory and Engineering,* Volume 1, pp. 588-591.

Kelley, J.E., 1963. The critical-path method: Resources planning and scheduling. In: J. Muth & G. Thompson, eds. *Industrial scheduling.* New Jersey: Prentice-Hall, pp. 347-365.

Kempf, K., Uzsoy, R., Smith, S. & Gary, K., 2000. Evaluation and comparison of production schedules. *Computers in Industry,* Volume 42, pp. 203-220.

Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science,* Volume 220, pp. 671-680.

Kolisch, R., 1996. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research,* Volume 90, pp. 320-333.

Kusiak, A. & Heragu, S.S., 1987. The facility layout problem. *European Journal of Operational Research,* Volume 29, pp. 229-251.

Laarhoven, P.J.M. v., Aarts, E.H.L. & Lenstra, J., 1992. Job Shop Scheduling by Simulated Annealing. *Operations Research,* Volume 40, pp. 113-125.

Makridakis, S., Wheelwright, S.C. & Hyndman, R.J., 1998. *Forecasting: methods and applications.* 3rd ed. New York: Wiley.

Meester, G.J. & Zijm, W.H.M., 1993. Multi-resource scheduling for an FMC in discrete parts manufacturing. In: M. Ahmad & W. Sullivan, eds. *Flexible Automation and Integrated Manufacturing.* Atlanta: CRC Press, pp. 360-370.

Miyakawa, S., 1991. *Simultaneous engineering and the producibility evaluation method.* Detroit, International Conference on the Application of Manufacturing Technologies.

Miyakawa, S. & Ohashi, T., 1986. *The Hitachi Assemblability Evaluation Method.* Newport, International Conference on Product Design for Assembly.

Pinedo, M.L., 2005. *Planning and Scheduling in Manufacturing and Services.* 1st ed. New York: Springers Science + Business Media.

Pinedo, M.L., 2012. *Scheduling: Theory, Algorithms, and Systems.* 4rd ed. New York: Springer Science+Business Media.

Schirmer, A., 1997. *Advanced biased random sampling in serial and parallel scheduling.* Research Report, University of Kiel.

Schutten, J.M.J., 1996. *Shop Floor Scheduling with Setup Times. PhD Thesis, University of Twente,* Enschede.

Schutten, J.M.J., 1998. Practical job shop scheduling. *Annals of Operations Research,* Volume 83, pp. 161-177.

Senthiil, P.V. & Selladurai, V., 2011. Optimal Job shop Scheduling Performance Enhancement Trough Computer Based Simulated Annealing Technique. *Journal for Manufacturing Science and Production,* Volume 8, pp. 169-180.

Seyedi, I., Maleki-Daronkolaei, A. & Kalashi, F., 2012. Tabu search and simulated annealing for new three-stage assembly flowshop scheduling with blocking. *Interdisciplinary Journal Of Contemporary Research In Business,* Volume Volume 4, pp. 394-402.

Silver, E.A., Pyke, D.F. & Peterson, R., 1998. *Inventory management and production planning and scheduling.* 3rd ed. New York: Wiley.

Sprecher, A., Kolisch, R. & Drexl, A., 1995. Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research,* Volume 80, pp. 94-102.

Suri, R., Sanders, J.L. & Kamanth, M., 1993. Performance evaluation of production networks. In: S. Graves, A. Rinnooy Kan & P. Zipkin, eds. *Handbooks in operations research and management science.* New York: Elsevier Science Publishers B.V., pp. 199-284.

Takahaski, K., Suzuki, I. & Suguro, T., 1989. *Producibility evalutation method.* Kanazawa, International Conference On Assembly Automation.

Wang, Z., Li, X. & Qi, X., 2008. The Design and Development of a Multi-agent Agile Scheduling System. *International Journal of Computational Intelligence Research,* Volume 4, pp. 45-52.

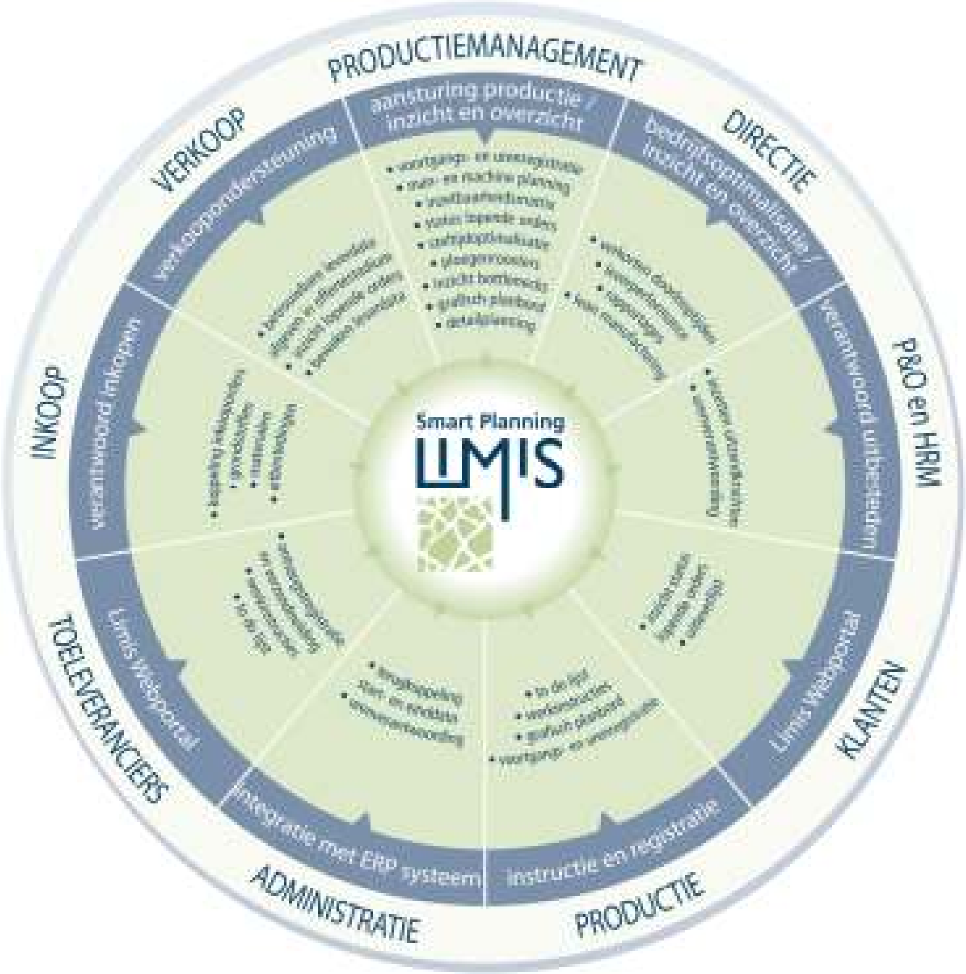
Wang, Z. & Liu, Y., 2006. A Multi-Agent agile scheduling system for job-shop problem. ISDA'06. Published by *IEEE Computer Society,* pp. 679.

Wemmerlov, U. & Hyer, N.L., 1989. Cellular manufacturing in the US industry: a survey of users. *International Journal of Production Research,* Volume 27, pp. 1511-1530.

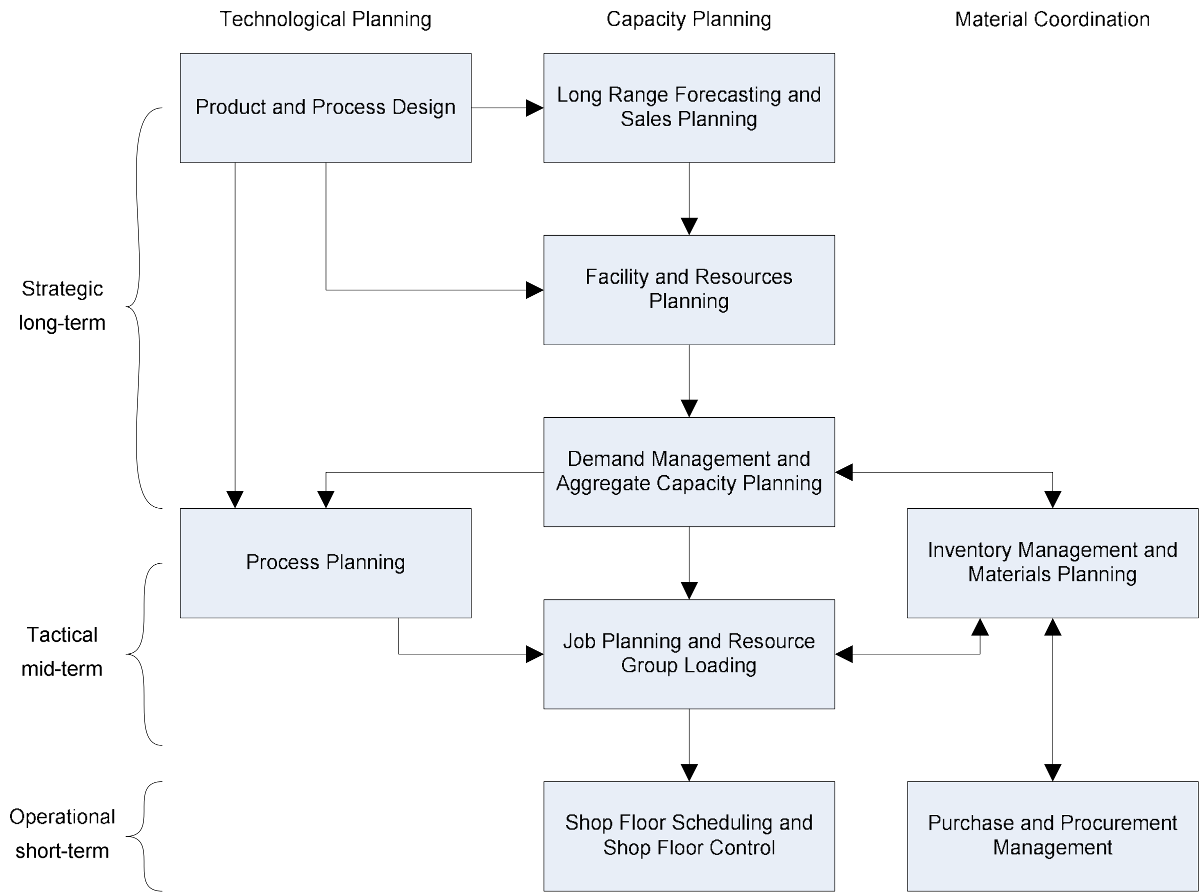
Yamada, T. & Nakano, R., 1997. Job-shop scheduling. In: A. Zalzala & P. Fleming, eds. *Genetic algorithms in engineering systems.* London: The Institution of Electric Engineers, pp. 134-160.

Zijm, W.H.M., 2000. Towards intelligent manufacturing planning and control systems. *OR Spektrum,* Volume 22, pp. 313-345.

Appendix A: Concept Smart Planning



Appendix B: Manufacturing planning and control architecture



Appendix C: Overview of planning in a production company

This appendix gives an overview of all relevant aspects of planning in a production company. The links between these different aspects are presented in a mathematical way. There could be made distinction between categories objectives in the planning. Minimizing costs is the first objective category and minimizing time the second category.

Relevant costs are changeover costs $tc_{ii'}$ between job i and job i', inventory costs ic_{kp} of material type k on day p, inventory costs ic_{mp} at machine m on day p, order costs ro_k of material type k, penalty costs pe_i of job i, penalty costs pe_o of order o, penalty costs pe_n of project n, set up costs cr_m of machine m, transportation costs $nc_{ii'}$ between job i and job i', using costs of machine um_i for job i , using costs of tool uq_i for job i, and using costs of employee uw_i for job i.

Relevant times are changeover times $tt_{ii'}$ between job i and job i', completion times ci_i of job i , completion times ci_o of order o, completion time ci_n of project n, set up times tr_m of machine m, transportation times $nt_{ii'}$ between job i and job i', time in inventory ti_{im} at machine m after job i, and time in inventory ti_k of material type k.

Equation (1) gives this objective function and Equation (2) and (3) show that all the priority factors β_r should be between zero and one and that these priorities sum up to one.

$$\begin{aligned} Min\Bigg(& \beta_1 \times \sum_{i=1}^{I} tt_{ii'} + \beta_2 \times \sum_{i=1}^{I} tc_{ii'} + \beta_3 \times \sum_{i=1}^{I} ci_i + \beta_4 \times \sum_{=1}^{O} ci_o + \beta_5 \\ & \times \sum_{i=n}^{N} ci_n + \beta_6 \times \sum_{i=1}^{I} nc_{ii'} + \beta_7 \times \sum_{i=1}^{I} nt_{ii'} + \beta_8 \times \sum_{i=1}^{I} um_i \\ & + \beta_9 \times \sum_{i=1}^{I} uq_i + \beta_{10} \times \sum_{i=1}^{I} uw_i + \beta_{11} \times \sum_{i=1}^{I} pe_i + \beta_{12} \\ & \times \sum_{=1}^{O} pe_o + \beta_{13} \times \sum_{i=n}^{N} pe_n + \beta_{14} \times \sum_{m=1}^{M} \sum_{p=1}^{P} ic_{mp} + \beta_{15} \\ & \times \sum_{k=1}^{K} \sum_{p=1}^{P} ic_{kp} + \beta_{16} \times \sum_{m=1}^{M} \sum_{i=1}^{I} ti_{im} + \beta_{17} \times \sum_{i=1}^{I} ti_k + \beta_{18} \\ & \times \sum_{k=1}^{K} ro_k + \beta_{19} \times \sum_{m=1}^{M} tr_m + \beta_{20} \times \sum_{m=1}^{M} cr_m \Bigg) \end{aligned} \tag{1}$$

$$0 \leq \beta_r \leq 1 \tag{2}$$

$$\sum_{r=1}^{R} \beta_r = 1 \tag{3}$$

There are a lot of different types of machines in a company. Machines that can process the same jobs, are collected in the same type of machines m. Equations (4), (6) and (7) show that at day p a job i could be processed by a type of machines m or not x_{imp}. The sum of planned jobs x_{imp} on type of machines m is equal to sum of jobs i that need that type of machines b_{im}. Equation (5) indicates that a job could be only planned at a type of machines when this job could be processed by this type of machines.

$$\sum_{p=1}^{P}\sum_{i=1}^{I} x_{imp} = \sum_{i=1}^{I} b_{im} \;\; (\forall m) \tag{4}$$

$$x_{imp} \leq b_{im} \tag{5}$$

$$x_{imp} = \{0,1\} \; (\forall i, \forall m, \forall p) \tag{6}$$

$$b_{im} = \{0,1\} \; (\forall i, \forall m) \tag{7}$$

A type of machines could process a limited number of jobs together, so each type of machines m has a limited processing capacity mc_{mp} on day p. Equation (8) gives the available capacity kc_{mp} of a type of machines m at day p, this capacity is calculated by reducing the maximum capacity mc_{mp} of a type of machines m at day p with the capacity c_{im} used by the jobs that are planned on that machine x_{imp}. The available capacity of a type of machines should be positive, but could be not higher than the maximum capacity for that day as showed in Equation(9). Equation (10) indicates, the required capacity could also be not more than the maximum capacity. However, the required capacity must be positive.

$$kc_{mp} = mc_{mp} - \; x_{imp} \times c_{im} \; (\forall m, \forall p) \tag{8}$$

$$0 \leq kc_{mp} \leq mc_{mp} \; (\forall m, \forall p) \tag{9}$$

$$0 \leq c_{im} \leq mc_{mp} \; (\forall i, \forall m) \tag{10}$$

Each type of machines has different costs, the costs of really using a type of machines um_i for processing job i are calculated by Equation (11). These costs are calculated by the sum of the using costs uc_m of the machines that are used for the job x_{imp}. The costs of using a type of machines should be positive as indicated at Equations (12) and (13).

$$um_i = \sum_{m=1}^{M} x_{imp} \times uc_m \; (\forall i) \tag{11}$$

$$uc_m \geq 0 \; (\forall m) \tag{12}$$

$$um_i \geq 0 \; (\forall i) \tag{13}$$

A lot of machines in production companies need different tools to process a job. There are a lot of tools in a company, tools that have the same possibilities are collected in a subset tools q. Each tool should be used by a machine that is suitable for that tool. Equations (14), (16), and (17) show that a job could be processed by a type of tools q or not x_{iqp}. All jobs should be scheduled, so the total number of planned jobs i at day p on that type of tools x_{iqp} is equal to the number of jobs that need that type of tools b_{iq}. Equation (15) indicates that a job i could be only planned x_{iqp} at a type of tools q when this job could be processed by this type of tools b_{iq}. Equation (18) shows that a job could be only planned on a machine x_{imp} and tool x_{iqp} when this type of tools could be installed on that type of machines b_{mq}. Equation (19) indicates whether a type of machines could use a certain type of tools or not.

$$\sum_{p=1}^{P}\sum_{i=1}^{I} x_{iqp} = \sum_{i=1}^{I} b_{iq}\ (\forall q) \tag{14}$$

$$x_{iqp} \leq b_{iq} \tag{15}$$

$$x_{iqp} = \{0,1\}\ (\forall i, \forall q, \forall p) \tag{16}$$

$$b_{iq} = \{0,1\}\ (\forall i, \forall q) \tag{17}$$

$$x_{iqp} \times x_{imp} \leq b_{mq} \tag{18}$$

$$b_{mq} = \{0,1\}\ (\forall m, \forall q) \tag{19}$$

Each type of tools could process a limited number of jobs together, so each type of tools has limited capacity. Each job needs capacity of the needed type of tools and a job could be only processed when a type of tools has enough capacity available. Equation (20) gives the available capacity kc_{qp} of a type of tools q at day p, this capacity is calculated by reducing the maximum capacity mc_{qp} of a type of tools q at day p with the capacity c_{iq} used by the planned jobs x_{iqp}. The available capacity should be positive, but could be not higher than the maximum capacity for that day as showed in Equation (21). Equation (22) indicates, the required capacity could be not more than the maximum capacity and must be positive.

$$kc_{qp} = mc_{qp} - \ x_{iqp} \times c_{iq}(\forall q, \forall p) \tag{20}$$

$$0 \leq kc_{qp} \leq mc_{qp}\ (\forall q, \forall p) \tag{21}$$

$$0 \leq c_{iq} \leq mc_{qp}\ (\forall i, \forall q) \tag{22}$$

Each type of tools has different costs, the costs of really using a type of tools uq_i for processing a job i are calculated by Equation (23). These costs are calculated with the sum of the using costs of a type of tools q of the jobs i that are processed by that tool x_{iqp}. The costs of using a type of tools should be positive as indicated at Equations (24) and (25).

$$uq_i = \sum_{m=q}^{Q} x_{iqp} \times uc_q \ (\forall i) \quad (23)$$

$$uc_q \geq 0 \ (\forall q) \quad (24)$$

$$uq_i \geq 0 \ (\forall i) \quad (25)$$

A lot of machines in production companies need an employee who operates the machine. Employees who could operate the same machines are collected in a type of employees w. Equations (26), (28), and (29) show that a job i could be processed on day p by a type of employees w or not x_{iwp}. All jobs should be scheduled, so the total number of processed jobs x_{iwp} with that type of employees is equal to the number of jobs that need that type of employees b_{iw}. Equation (27) indicates that a job could be only planned at a type of employees x_{iwp} when this job could be processed by this type of employees b_{iw}. Equation (30) shows that a job could be only planned on a machine x_{imp} and employee x_{iwp} when this type of employees could operate that type of machines b_{mw}. Sometimes when employees has more abilities, the employee have to work alternately, so therefore there is a limit to work at a type of machines for a type of employees. Equation (31) indicates that the number of jobs with the same type of machines and employees should be smaller than the maximum number of times mi_{wm} that an type of employees may operate a type of machines. Equation (32) shows if a type of machines could be operated by a type of employees or not. The maximum number of times that a type of employees may be operate a type of machines should be positive as indicated in Equation (33).

$$\sum_{p=1}^{P} \sum_{i=1}^{I} x_{iwp} = \sum_{i=1}^{I} b_{iw} \ (\forall w) \quad (26)$$

$$x_{iwp} \leq b_{iw} \quad (27)$$

$$x_{iwp} = \{0,1\} \ (\forall i, \forall w, \forall p) \quad (28)$$

$$b_{iw} = \{0,1\} \ (\forall i, \forall w) \quad (29)$$

$$x_{iwp} \times x_{imp} \leq b_{mw} \quad (30)$$

$$\sum_{i=1}^{I} \sum_{p=1}^{P} x_{imp} \times x_{iwp} \leq mi_{wm} \ (\forall m, \forall w) \quad (31)$$

$$b_{mw} = \{0,1\} \ (\forall m, \forall w) \quad (32)$$

$$mi_{wm} \geq 0 \ (\forall w, \forall m) \tag{33}$$

Some jobs need two employees for processing, but other jobs need only turning on the machine. Therefore each job needs a number of employees that are required c_{iw}. This could be an integer number, but also a part of an employee. Also each type of employees has limited capacity and a job could be only processed whether a type of employees has enough capacity available. Equation (34) gives the available capacity kc_{wp} of a type of employees w on day p. This capacity is calculated by reducing the maximum capacity mc_{wp} with the capacity c_{iw} used by the planned jobs x_{iwp}. The available capacity of a type of machines, tools, and employees should be positive, but could be not higher than the maximum capacity for that day as showed in Equation (35). Equation (36) indicate that the capacity that a job needs, could be not more than the maximum capacity, but must be positive.

$$kc_{wp} = mc_{wp} - \ x_{iwp} \times c_{iw} \ (\forall w, \forall p) \tag{34}$$

$$0 \leq kc_{wp} \leq mc_{wp} \ (\forall w, \forall p) \tag{35}$$

$$0 \leq c_{iw} \leq mc_{wp} \ (\forall i, \forall q) \tag{36}$$

Each type of employees has different costs, the costs of really using a type of employees uw_i for processing a job i is calculated by Equation (37). It is the sum of the using costs uc_w of the employees that are used for processing the job x_{iwp}. The using costs should be positive as indicated at Equations (38) and (39).

$$uw_i = \sum_{w=1}^{W} x_{iwp} \times uc_w \ \ (\forall i) \tag{37}$$

$$uc_w \geq 0 \ (\forall w) \tag{38}$$

$$uw_i \geq 0 \ (\forall i) \tag{39}$$

When a job needs another type of tools than installed on the machine, this type of tools should be installed. This costs time and money, so there should be changeover time and costs between type of tools. There could be also changeover times and costs between jobs. For example, when the same tool could be used, but the tools have to be cleaned. Also when a machine gets another employee, this costs money and time, so there are also changeover time and costs between types of employees. Equation (40) shows the total changeover time $tt_{ii'}$ for a job i when it is planned after job i'. This changeover time is a sum of the changeover time $ct_{ii'}$ between job i and job i', the changeover time $ct_{qq'}$ between tool q and tool q', and the changeover time $ct_{ww'}$ between employee w and employee w'. Equations (41), (42), (43), and (44) indicate that the changeover times should be positive. Equation (45) shows the total changeover costs $tc_{ii'}$ between job i and job i'. These changeover costs are a sum of the changeover costs $cc_{ii'}$ between job i and job i', the changeover costs $cc_{qq'}$ between tool q and tool q', and the changeover costs $cc_{ww'}$ between employee w and employee w'. Equations (46), (47), (48), and (49) indicates that the changeover costs should be positive.

$$tt_{ii'} = x_{ii'} \times \left(ct_{ii'} + \sum_{q=1}^{Q} \sum_{p=1}^{P} x_{iqp} \times x_{i'q'p} \times ct_{qq'} + \sum_{q=1}^{Q} \sum_{p=1}^{P} x_{iwp} \times x_{i'w'p} \times ct_{ww'} \right) (\forall i) \tag{40}$$

$$tt_{ii'} \geq 0 \ (\forall i) \tag{41}$$

$$ct_{ii'} \geq 0 \ (\forall i) \tag{42}$$

$$ct_{qq'} \geq 0 \ (\forall q) \tag{43}$$

$$ct_{ww'} \geq 0 \ (\forall w) \tag{44}$$

$$tc_{ii'} = x_{ii'} \times \left(cc_{ii'} + \sum_{q=1}^{Q} \sum_{p=1}^{P} x_{iqp} \times x_{i'q'p} \times cc_{qq'} + \sum_{q=1}^{Q} \sum_{p=1}^{P} x_{iwp} \times x_{i'w'p} \times cc_{ww'} \right) (\forall i) \tag{45}$$

$$tc_{ii'} \geq 0 \ (\forall i) \tag{46}$$

$$cc_{ii'} \geq 0 \ (\forall i) \tag{47}$$

$$cc_{qq'} \geq 0 \ (\forall q) \tag{48}$$

$$cc_{ww'} \geq 0 \ (\forall w) \tag{49}$$

Starting a machine also takes some time, so there should be assigned setup times and costs to type of machines. Equation (50) indicates, there is only set up time tr_{mp} when the machine is used x_{imp}. Equation (51) shows that the set up time of a machine is positive.

$$tr_{mp} = \max(x_{imp} \times st_m) \tag{50}$$

$$st_m \geq 0 \ (\forall m) \tag{51}$$

Whether an order needs multiple machines, the product should be transported. So there are transportation times and costs between machines. Equation (52) represents, transportation time $nt_{ii'}$ is only present between job i and job i' whether these are sequenced. Equation (53) shows that the transportation time between jobs should be positive. Also the transportation time between machines should be positive as showed in Equation (54).

$$nt_{ii'} = \sum_{m=1}^{M} \sum_{p=1}^{P} x_{imp} \times x_{i'm'p} \times tt_{mm'} \quad (52)$$

$$nt_{ii'} \geq 0 \; (\forall i) \quad (53)$$

$$tt_{mm'} \geq 0 \; (\forall m) \quad (54)$$

Equation (55) indicates that there are only set up costs cr_{mp} when the machine is used x_{imp}. Equation (56) indicates that the set up costs of a machine are positive.

$$cr_{mp} = \max(x_{imp} \times sc_m) \quad (55)$$

$$sc_m \geq 0 \; (\forall m) \quad (56)$$

Equation (57) represents the total transportation costs $nc_{ii'}$ between job i and job i' that are sequenced. Equations (58) show that the transportation costs between jobs should be positive and the transportation costs between machines should also be higher than zero as showed in Equation (59).

$$nc_{ii'} = \sum_{m=1}^{M} \sum_{p=1}^{P} x_{imp} \times x_{i'm'p} \times tc_{mm'} \quad (57)$$

$$nc_{ii'} \geq 0 \; (\forall i) \quad (58)$$

$$tc_{mm'} \geq 0 \; (\forall m) \quad (59)$$

A machine could be broken, so the time that the machine is available should be taken into account. Therefore the time to failure should be known from each type of machines and the time that is needed to repair each type. Equation (60) gives the definition of the availability a_m of a machine m. A machine is available all the time reduced with the time that is needed for repairing rt_m. The mean time to failure ft_m should be positive as showed in Equation (61). Also the repair time should be longer than zero as showed in Equation (62). Equation (63) shows that the availability is a percentage, so a number between zero and one.

$$a_m = \frac{ft_m}{ft_m + rt_m} \; (\forall m) \quad (60)$$

$$ft_m \geq 0\ (\forall m) \tag{61}$$

$$rt_m > 0\ (\forall m) \tag{62}$$

$$0 \leq a_m \leq 1\ (\forall i) \tag{63}$$

The utilization of a type of machines is relevant, because the utilization shows how busy a type of machines is. For example, whether a type of machines has a high utilization it is undesirable to have a high changeover time. Whether the type of machines has a low utilization the problem is smaller. This is the same, for the utilization of the tools and employees. Equations (64), (65), and (66) shows the utilization u_m of each type of machines m, the utilization u_q of each type of tools q and the utilization u_w of each type of employees w. These utilizations are calculated by how much of the maximum capacity is unused. Equations (67), (68), and (69) show that the utilization of respectively type of machines, tools and employees should be between zero and one.

$$u_m = \sum_{p=1}^{P} \left(\frac{mc_{mp} - kc_{mp}}{mc_{mp}} \right)\ (\forall m) \tag{64}$$

$$u_q = \sum_{p=1}^{P} \left(\frac{mc_{qp} - kc_{qp}}{mc_{qp}} \right)\ (\forall q) \tag{65}$$

$$u_w = \sum_{p=1}^{P} \left(\frac{mc_{wp} - kc_{wp}}{mc_{wp}} \right)\ (\forall w) \tag{66}$$

$$0 \leq u_m \leq 1\ (\forall m) \tag{67}$$

$$0 \leq u_q \leq 1\ (\forall q) \tag{68}$$

$$0 \leq u_w \leq 1\ (\forall w) \tag{69}$$

In a production company materials are needed to process a job. Equations (70) and (71) show that a job could be processed with a type of material or not x_{ikp} and that the total number of processed jobs with material type k is equal to the number of jobs that need that type of material b_{ik}. Equation (72) shows that a job needs a type of material or not. Material should be delivered by a supplier. When a supplier delivers more types of materials, it is handy to know which material is delivered by which supplier, because then orders could be combined. Equation (73) shows if a certain type of materials is delivered by a supplier or not b_{kl}.

$$\sum_{p=1}^{P} \sum_{i=1}^{I} x_{ikp} = \sum_{i=1}^{I} b_{ik}\ \ (\forall k) \tag{70}$$

$$x_{ikp} = \{0,1\} \; (\forall i, \forall k, \forall p) \tag{71}$$

$$b_{ik} = \{0,1\} \; (\forall i, \forall k) \tag{72}$$

$$b_{kl} = \{0,1\} \; (\forall k, \forall l) \tag{73}$$

Not only which material is relevant, but also how much material. Each supplier l has a quantity delivery percentage dq_l which is given by Equation (74). This is calculated with the number of right quantities gq_l in relation to the number of wrong quantities wq_l. Equation (75) indicates whether the right quantity is delivered, so the difference between the ordered quantity oq_{kh} of order h of material type k and the arriving quantity rq_{kh} is equal to zero. Equation (76) indicates whether the right quantity is not delivered, so the difference between the ordered and arriving quantity is not equal to zero. Equation (77) shows that the ordered volume v_k of a type of material k should be smaller than the space mc_k for that material type in inventory. Equation (78) shows that the maximum capacity should be positive. Equation (79) shows that the order quantity should be positive. The volume and the arriving quantity of a material type should be positive as showed in Equations (80) and (81).

$$dq_l = \frac{\sum_{h=1}^{H} \sum_{k=1}^{K} (b_{kl} \times gq_{kh})}{\sum_{h=1}^{H} \sum_{k=1}^{K} (b_{kl} \times gq_{kh} + b_{kl} \times wq_{kh})} \; (\forall l) \tag{74}$$

$$gq_{kh} = 1 \; if \; oq_{kh} - rq_{kh} = 0 \; (\forall k, \forall h) \tag{75}$$

$$wq_{kh} = 1 \; if \; oq_{kh} - rq_{kh} \neq 0 \; (\forall k, \forall h) \tag{76}$$

$$oq_{kh} \times v_k \leq mc_k (\forall k, \forall h) \tag{77}$$

$$mc_k \geq 0 \; (\forall k) \tag{78}$$

$$oq_{kh} \geq 0 \; (\forall k, \forall h) \tag{79}$$

$$v_k \geq 0 \; (\forall k) \tag{80}$$

$$0 < rq_{kh} \leq mc_k (\forall k, \forall h) \tag{81}$$

It needs time to deliver material, so each type of material has a delivery time and an order date. Each supplier has a percentage for on time delivery. Equation (82) gives the delivery percentage dq_l of the suppliers l, so how many times the delivery order is delivered on time. This is calculated as the on time deliveries ot_{kh} in relation to the too late deliveries tl_{kh} Equation (83) indicates if a delivery order is delivered on time, so if the difference between the order date and delivery date is equal to the delivery time. Equation (84) indicates if a delivery order is delivered too late or too early, so when the difference between the order date and arriving date is not equal to the delivery time. Equation (85) indicates that a delivery order could be not delivered on time and too late. Equation (86) shows that the arriving date of a delivery order should be further in time than the order date of this order. Equation (87) indicates that the delivery time must be positive.

$$dp_l = \frac{\sum_{h=1}^{H}\sum_{k=1}^{K}(b_{kl} \times ot_{kh})}{\sum_{h=1}^{H}\sum_{k=1}^{K}(b_{kl} \times tl_{kh} + b_{kl} \times ot_{kh})} \; (\forall l) \tag{82}$$

$$ot_{kh} = 1 \; if \; ra_{kh} - od_{kh} = dt_l \; (\forall k, \forall h) \tag{83}$$

$$tl_{kh} = 1 \; if \; ra_{kh} - od_{kh} \neq dt_l \; (\forall k, \forall h) \tag{84}$$

$$tl_{kh} + ot_{kh} \leq 1 \; (\forall k, \forall h) \tag{85}$$

$$ra_{kh} \geq od_{kh} \; (\forall k, \forall h) \tag{86}$$

$$dt_l \geq 0 \; (\forall l) \tag{87}$$

Ordering material costs money, so also order costs should be assigned to a type of material. Equation (88) shows that the real order costs ro_k are dependent of how often a material type is ordered od_{kh} in combination with the height of these costs per order oc_k. Equation (89) indicates that the order costs should be positive.

$$ro_k = \sum_{h=1}^{H} od_{kh} \times oc_k \tag{88}$$

$$oc_k \geq 0 \; (\forall k) \tag{89}$$

Equation (90) represents the inventory level gi_{kp} of material type k at the end of day p. This is calculated with the inventory of the day before gi_{kp-1} plus the quantity of material dq_{kh}, that arrives ad_{khp} minus the material that is used for production g_{ik} whether a job is processed x_{ikp}. The volume vi_{kp} of each material type k on day p is calculated by Equation (91) and the total volume in inventory vi_p on day p is calculated by Equation (92). It is not possible to have an infinite inventory, because there is a limited capacity for inventory. So there is a maximum inventory size which is dependent of the volume of the material. The volume in inventory should be smaller than the maximum capacity as specified in Equations (93) and (94). Equation (95) shows the quantity a job needs a type of material could be not negative and also the quantity of material in inventory must be positive as showed in Equation(96).

$$gi_{kp} = gi_{kp-1} + ad_{khp} \times dq_{kh} - x_{ikp} \times g_{ik} \; (\forall k, \forall p) \tag{90}$$

$$vi_{kp} = gi_{kp} \times v_k \ (\forall k, \forall p) \tag{91}$$

$$vi_p = \sum_{k=1}^{K} vi_{kp} \ (\forall p) \tag{92}$$

$$vi_{kp} \leq mc_k \ (\forall k, \forall p) \tag{93}$$

$$vi_p \leq mc \ (\forall p) \tag{94}$$

$$g_{ik} \geq 0 \ (\forall i, \forall k) \tag{95}$$

$$gi_{kp} \geq 0 \ (\forall k, \forall p) \tag{96}$$

Materials have to be stored which involves costs, so holding costs must be assigned to each type of material. Equation (97) gives the inventory costs ic_{kp} of a material type k at day p. These are dependent of the holding costs hc_k for the type of material and how much material is in inventory gi_{kp}. Also the intermediate and finished products must be stored. Each machine has restricted inventory space for waiting jobs. Equation (98) gives the inventory costs ic_{mp} at the machines m on day p which is calculated with the time in inventory ti_i, the holding costs hc_i and if the job is processed on the machine x_{imp}. Also the holding costs must be positive which is indicated with Equations (99) and (100).

$$ic_{kp} = \ hc_k \times \ \ (\forall k, \forall p) \tag{97}$$

$$ic_{mp} = \sum_{i=1}^{I} x_{imp} \times hc_i \times ti_i \ (\forall m, \forall p) \tag{98}$$

$$hc_i \geq 0 \ (\forall i) \tag{99}$$

$$hc_k \geq 0 \ (\forall k) \tag{100}$$

Equation (101) gives the time ti_{im} that a result of a job i is inventory at machine m, this is dependent of the difference between the completion time ci_i of a job i and the starting date $sd_{i'}$ of the successor. When there is no successor, this is dependent of the difference between the completion time and the shipping date ts_n of a project. Sometimes there is also an maximum time that a product could be in inventory. For example, this is relevant in the production of product with expiration date. The time in inventory ti_{im} should be shorter than the maximum time mt_i in inventory which is specified in Equation (102). Equation (103) indicates that the volume v_i in the inventory at a machine type should be smaller than the capacity of the inventory mc_m at machine type m. Equation (104) shows that the maximum capacity should be positive. Equation (105) shows that the time in inventory should be positive. The volume of a result of a job should be positive as showed in Equation (106).

$$ti_{im} = \min\left(b_{im} \times bs_{ii'} \times (ci_i - sd_{i'}), b_{im} \times (ts_n - ci_i)\right) \ (\forall i, \forall m) \tag{101}$$

$$ti_{im} \leq mt_i \ (\forall i, \forall m) \tag{102}$$

$$ti_{im} \times v_i \leq mc_m \ (\forall i, \forall m) \tag{103}$$

$$mc_m \geq 0 \ (\forall m) \tag{104}$$

$$ti_{im} > 0 \ (\forall i, \forall m) \tag{105}$$

$$v_i \geq 0 \ (\forall i) \tag{106}$$

Equation (107) gives the completion time ci_n of the project n which is the completion time ci_o of the latest completed order o of the project. The completion time ci_o of the order o is given by Equation (108), this is the completion time ci_i of the latest completed job i of the order. Equation (109) shows that a job i is part of an order o or not b_{io}. Equation (110) shows that an order o is part of a project n or not b_{on}.

$$ci_n = max(b_{on} \times ci_o) \ (\forall n) \tag{107}$$

$$ci_o = max(b_{io} \times ci_i) \ (\forall o) \tag{108}$$

$$b_{io} = \{0,1\} \ (\forall i, \forall o) \tag{109}$$

$$b_{on} = \{0,1\} \ (\forall o, \forall n) \tag{110}$$

The status of the project and the order should be showed for an overview of the progress. Equations (111) and (112) give the status z_n of the project n and the status z_o of the order o, so how much is completed. Each order consists of different jobs and has jobs which must be ready before a job could start. Equation (113) shows the number of predecessors pd_i. The job cannot be processed before all the predecessors are completed. The release date of a job should also be further in time than the completion dates of the predecessors. Equations (114) and (115) show which jobs are successors $bs_{ii'}$ and predecessors $bp_{ii'}$ from other jobs. The orders should be assigned to a customer, because when a customer has different orders, these orders should be ready around the same moment for joint transportation. Equation (116) indicates whether an order has a certain customer or not.

$$z_n = \frac{\sum_{o=1}^{O} b_{on} \times ci_o}{\sum_{o=1}^{O} b_{on}} \ (\forall n) \tag{111}$$

$$z_o = \frac{\sum_{i=1}^{I} b_{io} \times ci_i}{\sum_{i=1}^{I} b_{io}} \ (\forall o) \tag{112}$$

$$pd_i = \sum_{i'=1}^{I} bp_{ii'} \ (\forall i) \tag{113}$$

$$bp_{ii'} = \{0,1\} \ (\forall i) \tag{114}$$

$$bs_{ii'} = \{0,1\} \ (\forall i) \tag{115}$$

$$b_{oe} = \{0,1\} \ (\forall o, \forall e) \tag{116}$$

Equations (117), (118), and (119) shows the reliability of the company re, the reliability re_n of a project n, and the reliability re_o of an order o. These reliabilities are calculated by the percentage of the projects, orders, and jobs that are completed on time. Equations (120), (121), and (122) indicate that the reliabilities should be between zero and one. The project should be ready at a certain moment. Most of the times, this is dependent of appointments with the customer. This date is called the due date, also orders and jobs have due dates. There is also a date that the job is really finished, this is the completion date of a job. Equations (123), (124), and (125) indicate if jobs, orders, and projects are completed on time, so if the due date is later in time than the completion date. Equations (126) and (127) indicate if a job, an order or a project is completed too late, so when the completion date is further in time than the due date.

$$re = \frac{\sum_{n=1}^{N} ot_n}{\sum_{n=1}^{N} tl_n + ot_n} \tag{117}$$

$$re_n = \frac{\sum_{o=1}^{O} b_{on} \times ot_o}{\sum_{o=1}^{O} b_{on} \times tl_o + b_{on} \times ot_o} \ (\forall n) \tag{118}$$

$$re_o = \frac{\sum_{i=1}^{I} b_{io} \times ot_i}{\sum_{i=1}^{I} b_{io} \times tl_i + b_{io} \times ot_i} \ (\forall o) \tag{119}$$

$$0 \leq re \leq 1 \tag{120}$$

$$0 \leq re_n \leq 1 \ (\forall n) \tag{121}$$

$$0 \leq re_o \leq 1 \ (\forall o) \tag{122}$$

$$ot_i = 1 \ if \ ci_i \leq dd_i \ (\forall i) \tag{123}$$

$$ot_o = 1 \ if \ ci_o \leq dd_o \ (\forall o) \tag{124}$$

$$ot_n = 1\ if\ ci_n \leq dd_n\ (\forall n) \tag{125}$$

$$tl_i = 1\ if\ ci_i > dd_i\ (\forall i) \tag{126}$$

$$tl_n = 1\ if\ ci_n > dd_n\ (\forall n) \tag{127}$$

$$tl_o = 1\ if\ ci_o > dd_0\ (\forall o) \tag{128}$$

Equation (129) equates the earliest starting date es_i to the release date rd_i. The latest starting date ls_i is defined in Equation (130) as the latest moment that a job could start, taken into account the processing time pt_i and the due date dd_i. The starting date of a job should be further in time than the release date and the earliest starting date of a job as showed in Equations (131) and (132).

$$es_i = rd_i\ (\forall i) \tag{129}$$

$$ls_i = dd_i - pt_i(\forall i) \tag{130}$$

$$sd_i \geq rd_i\ (\forall i) \tag{131}$$

$$sd_i \geq es_i\ (\forall i) \tag{132}$$

The completion date could be later than the due date, but then lateness is created for the job and a penalty for the job is incurred. Lateness and a penalty are also incurred for orders and projects whether the completion dates are later than the due dates. The lateness la_i of job i, the lateness la_o of order o and the lateness la_n of project n are given by respectively equation (134), (138), and (142). Lateness is defined as the difference between the due date and the completion date. Equations (133), (137), and (141) give the penalty pe_i for a job i, the penalty pe_o for an order o, and the penalty pe_n for a project n. These penalties are dependent of the lateness and the standard penalty height ph. It should be possible that there is a limit to lateness, therefore is included a maximum lateness for jobs, orders and projects. Equations (135), (143), and (139) indicate that the lateness of a project, order and job should be smaller than the maximum lateness. The penalty heights should be zero or higher as showed in Equations (136), (144), and (140).

$$pe_i = la_i \times ph_i\ (\forall i) \tag{133}$$

$$la_i = max(0, ci_i - dd_i)\ \ (\forall i) \tag{134}$$

$$la_i \leq ml_i\ (\forall i) \tag{135}$$

$$ph_i \geq 0 \ (\forall i) \tag{136}$$

$$pe_o = la_o \times ph_o \ (\forall o) \tag{137}$$

$$la_o = max(0, ci_o - dd_o) \ (\forall o) \tag{138}$$

$$la_o \leq ml_o \ (\forall o) \tag{139}$$

$$ph_o \geq 0 \ (\forall o) \tag{140}$$

$$pe_n = la_n \times ph_n \ (\forall n) \tag{141}$$

$$la_n = max(0, ci_n - dd_n) \ (\forall n) \tag{142}$$

$$la_n \leq ml_n \ (\forall n) \tag{143}$$

$$ph_n \geq 0 \ (\forall n) \tag{144}$$

The processing time pt_i of a job i is represented by Equation (145) and the processing times pt for a job on the different type of machines, tools, and employees are given by Equations (146), (147), and (148). When jobs could be processed on different types of machines with different speeds, the speed is relevant. So each type of machines has a speed rate. Also each type of employees and type of tools have not the same speed, so also tools and employees have a speed rate. Equations (149), (150), and (151) show that the speed rates of a type of machines, tools and employees should be positive. The standard processing time of a job should be positive as indicated by Equation (152).

$$pt_i = \max\,(pt_{im}, pt_{iq}, pt_{iw}) \tag{145}$$

$$pt_{im} = \frac{sp_i}{s_m} \ (\forall i, \forall m) \tag{146}$$

$$pt_{iq} = \frac{sp_i}{s_q} \ (\forall i, \forall q) \tag{147}$$

$$pt_{iw} = \frac{sp_i}{s_w} \ (\forall i, \forall w) \tag{148}$$

$$s_m > 0 \ (\forall m) \tag{149}$$

$$s_q > 0 \ (\forall q) \tag{150}$$

$$s_w > 0 \ (\forall w) \tag{151}$$

$$sp_i \geq 0 \ (\forall i) \tag{152}$$

Equations (153) and (154) show that a job could be processed in sequence with another job or not $x_{ii'}$ and that all jobs have to be sequenced.

$$\sum_{i=1}^{I} x_{ii'} = 1 \ (\forall i) \tag{153}$$

$$x_{ii'} = \{0,1\} \ (\forall i) \tag{154}$$

Appendix D: Results of Case I

Job Tardiness in hrs	Situation 1	Situation 2	Situation 3	Situation 4	Situation 5
EDD	48	15	103.25	100.5	80.5
SST	1403.25	1161.75	1768.25	1751.25	2114.75
SPT	1132.75	968	1249.5	1011.75	1312.75
SA with 0.95	48	15	103.25	100.5	58
SA with 0.99	29.25	15	101.25	96.25	62.25
RBRS with 100	41.25	19.25	80.25	89.5	69.25
RBRS with 1000	28.25	14.5	74	77.5	52.75
Utilization in %					
EDD	45.95	45.62	52.97	49.95	54.27
SST	50.70	47.72	51.98	50.18	51.03
SPT	53.13	48.65	57.87	52.09	48.73
SA with 0.95	46.01	45.62	52.97	49.95	49.10
SA with 0.99	44.77	45.62	52.20	46.27	49.90
RBRS with 100	46.01	45.71	53.25	50.74	54.88
RBRS with 1000	45.84	44.85	53.57	50.57	54.36
Operation Tardiness in hrs					
EDD	274	73	476.25	675	328.75
SST	8488.25	6194.5	10364	8092	10152.75
SPT	5351	5450	6371	6436.75	7345.75
SA with 0.95	273.25	73	476.25	675	402.75
SA with 0.99	264.5	73	486.5	623.75	377
RBRS with 100	244	63.25	315.5	529.25	400.25
RBRS with 1000	205	57	262	517.25	138.5
Setup Time in hrs					
EDD	584.75	593.75	593.75	579.75	582.75
SST	497.5	505	504.25	499.5	493.25
SPT	580.75	598.75	588.25	578	577.75
SA with 0.95	586.5	593.75	593.75	579.75	557.5
SA with 0.99	556.5	593.75	596.5	558.5	558.75
RBRS with 100	583.5	592	591.5	583	576.75
RBRS with 1000	584	594.25	591.5	582.25	576.5
Computation Time in sec					
EDD	1	1	1	1	1
SST	1	1	1	1	1
SPT	1	1	1	1	1
SA with 0.95	231	231	225	229	227
SA with 0.99	1192	1178	1120	1168	1156
RBRS with 100	89	75	99	93	92
RBRS with 1000	828	742	927	920	834

Appendix E: Results of Case II

Objective in hrs	Situation 1	Situation 2	Situation 3	Situation 4	Situation 5
EDD	1022.625	1011.625	975.375	994.5	1030.375
SST	1341.375	1214.5	1207.625	1214.5	1118.875
SPT	2293.875	1923.5	1987.375	2187.625	1775.375
SA with 0.95	752	809.625	715	746.25	785.125
SA with 0.99	673.25	686.75	662.5	658.625	737.25
RBRS with 100	973.625	972.375	955.5	967.25	991.5
RBRS with 1000	965.875	957.375	935.875	962.625	956.875
Job Tardiness in hrs					
EDD	42.75	23.25	20.75	19	85.75
SST	1462.75	1176.5	1170.25	1176.5	902.75
SPT	2627.75	1902	1992.25	2485.25	1630.75
SA with 0.95	41.5	71.75	50	32.5	102.75
SA with 0.99	44	68.5	40	14.75	132
RBRS with 100	32.25	37.25	8.5	14.5	53
RBRS with 1000	14.25	39.75	11.75	25.25	66.25
Utilization in %					
EDD	30.08	31.28	30.09	27.15	29.51
SST	41.88	36.96	38.91	38.67	31.65
SPT	32.54	29.46	30.90	33.05	30.97
SA with 0.95	42.02	38.02	39.12	38.56	33.02
SA with 0.99	42.74	38.88	39.34	39.02	35.26
RBRS with 100	30.50	33.01	29.82	26.17	29.62
RBRS with 1000	31.64	33.64	30.40	27.32	30.01
Operation Tardiness in hrs					
EDD	182.25	101.25	111.5	118.75	1274.75
SST	7696.25	6421	6199	5723.25	8793.75
SPT	12067.25	11866.75	9746.5	9593.75	13146
SA with 0.95	217.75	712	266	309.25	2070
SA with 0.99	273.25	404.75	157.5	250.75	1662.75
RBRS with 100	131.25	254.75	10.50	118.75	940.75
RBRS with 1000	81.75	433.25	65.25	249	978.5
Setup Time in hrs					
EDD	2002.5	2000	1930	1970	1975
SST	1220	1252.5	1245	1252.5	1335
SPT	1960	1945	1982.5	1890	1920
SA with 0.95	1462.5	1547.5	1380	1460	1467.5
SA with 0.99	1302.5	686.75	1285	1302.5	1342.5
RBRS with 100	1915	1907.5	1902.5	1920	1930
RBRS with 1000	1917.5	1875	1860	1900	1847.5
Computation Time in sec					
EDD	1	1	1	1	1
SST	1	1	1	1	1
SPT	1	1	1	1	1
SA with 0.95	281	276	277	280	272
SA with 0.99	1408	1382	1388	1402	1374
RBRS with 100	131	127	128	136	133
RBRS with 1000	1302	1286	1294	1307	1300