Shutdown Scheduling

A practical approach to handle shutdowns at refineries

July 2004

Marco Bijvank

Master thesis in Business Mathematics and Informatics
Faculty of Sciences

Supervisors:
Prof. Dr. G.J. Franx, Vrije Universiteit
Prof Drs. E. Marchiori, Vrije Universiteit
Ir. G.C. Brandt, ORTEC
Drs. W.J. Lukje, ORTEC

Preface

For the final phase of my study Business Mathematics and Informatics (BMI) at the *vrije* Universiteit Amsterdam I had to do a six-month internship (February 2004 – July 2004). ORTEC bv in Gouda gave me the opportunity to do this at the Oil & Gas department. Appendix A provides some background information about the company and in more detail about the O&G department.

The subject of my thesis is shutdown scheduling (for refineries). This is a new business area for ORTEC. During six months I had to find out whether there was a need for a shutdown scheduling tool within the refineries. To gather information about the subject I talked to several people that are currently involved in some part of the shutdown scheduling process. Next, I had to summarize the possibilities and figure out whether it was possible to develop a tool, which makes a shutdown schedule. The last step was the development and implementation of the algorithm. It was a great challenge for me.

I would like to thank ORTEC for this opportunity they gave me. Furthermore, I would like to acknowledge the efforts of several people who helped me during my internship:

- Acknowledge the assistants, the constructive comments and the critical notes provided by Sandra Lukje (ORTEC), Gregor Brandt (ORTEC) and Geert-Jan Franx (VU)
- the provided information and cooperation from Shell; Erno Prosman, Henk Brobbel, Cees Lageweg, Albert Rink, Frans Derks, Jos Boode and Arjen Boelens.
- the collaboration and support from the PLANWISE team; Marco Schutten, John Nieuwenhuis, Daniël Dam and Niels Calis.
- moral support from my respective family and girlfriend. Their understanding is gratefully recognized.

Executive Summary

A refinery has a very complex manufacturing process in which different kinds of units are used to transform crude oil into refined products. All these processing units sooner or later malfunction or wear out, so refineries require periodic maintenance activities in order to keep the performance at a good level. That means that most units will at sometime have a shutdown from operation to be able to perform replacing, repairing, cleaning and modifying on various internal parts. Most of these maintenance activities cannot be performed during the operation of the unit, that is why the unit must be closed while the activities are occurring. These maintenance periods are known in the industry as turnarounds or shutdowns. Contractors, who are especially hired for this, perform the maintenance activities.

The objective of this thesis is to develop a method, which determines in which sequence the processing units should be shut down (for some predefined period of time) and what the flow of the materials should look like for the other units that are in process. It can be seen as a static deterministic scheduling problem, with a dynamic availability of the resources (units). The shutdown schedule should formulate the shutdown activities and the flow of materials in terms of quantities, for each day.

A mathematical model is formulated for the shutdown scheduling problem. This resulted in a MIP, which cannot be solved within an acceptable amount of time. Hence, the problem is simplified by neglecting most of the detailed production aspects at the refinery. Some of the aspects are captured in a relationship. This is the same way in which the refineries schedule the shutdowns in practice. This resulted in a number of different relationships:

- precedence relationships connect shutdowns between units that specify which shutdowns must precede other shutdowns and by how much of a delay or by how much allowed overlap they should take place
- simultaneous relationships connect shutdowns between units that should be shut down simultaneously (also called a parallel relationship)
- non-simultaneous relationships connect shutdowns between units that should be shut down sequentially (also called a serial relationships)

Besides relationships, the contractors have to be taken into account as well. There are three kinds of restrictions concerning the contractors:

- the required number of contractors for the shutdown of a unit should be available during the entire shutdown duration. It is very well possible that a shutdown requires several skilled technicians. For each skill, a restriction is formulated.
- there is a maximum amount of workforce the refinery can handle.
- there is a maximum amount of workforce available from the market perspective.

The shutdown scheduling problem is translated into the scheduling of activities (the shutdowns of the units). The shutdowns take some predetermined number of time periods (the shutdown duration) and have to take place under resource constraints and general precedence relationships. This is typically a resource constrained project scheduling problem with generalized precedence relationships.
An algorithm is developed to solve this problem. The algorithm is a heuristic and breaks the problem into several independent subproblems, which are solved sequentially. The algorithm is based on priority rules. By incorporating these priority rules, the algorithm is made problem specific. This was required since the problem is very complex.

The algorithm performs well, since the feasible schedules minimises the makespan and the total flow time. The procedure however does not guarantee to find the optimal solution. The computation times are very dependent on the restrictiveness of the problem. They could range from only a few seconds until unacceptable long. This later situation will happen in specific cases, which could be checked first.

One of the intentions of the research was to find out whether there was a need for a shutdown scheduling tool within the refineries. During several meetings with people from a refinery it was quite clear that the

use of a shutdown scheduling tool can be very beneficial. A number of reasons give rise to this conclusion:

1. more consistency
1. saving time to execute the scheduling process
1. improved schedules (shorter shutdown period and less time is required to perform the shutdowns)
1. no mistakes are made in the schedule anymore (except due to data errors). This will eventually save the refineries money, as the number of hired contractors is based on the shutdown schedule. If the schedule is incorrect, the hired number of contractors becomes too extensive or even insignificant.
1. the information concerning the shutdowns is captured within a tool.

Concluding can be said that shutdown scheduling definitely has potentials for ORTEC to focus on. The developed algorithm has to be developed further. So performance will increase and all user requirements are granted. The application also has to get a user-friendly graphical user interface, such that the refinery can use all functionalities of the application optimally.

Contents

1 Introduction

The refinery industry is a very capital-intensive sector. Maintenance has to take place at a refinery in order to assure its reliability. During a period of a couple of months a part of the refinery is shut down in order to perform maintenance activities, inspections etc. This period will cost the refinery a few million dollars, due to lost productions. So it is very important to prepare this shutdown period well. The way to determine which units have to be shut down at a particular timestamp is the subject of this thesis.

This thesis is organized in three parts: In chapter 2 till chapter 4 the first part the problem and its context are described. Chapter 2 contains an introduction of the refinery business and the relevant definitions for this thesis. This chapter will also introduce the concept of shutdown scheduling. In chapter 3 the context of a shutdown project is discussed in more detail. This chapter will include the problem description with all aspects that have to be taken into account. The problem is then formalized in a mathematical problem in chapter 4.
The second part of this thesis describes the approach, which is used, to solve the shutdown scheduling problem. Chapter 5 will give a literary overview of the relevant scheduling techniques. The developed algorithm, which will solve the problem, is discussed in chapter 6.
In the third and final part of this thesis the developed technique is applied and expanded to suit the shutdown scheduling context in chapter 7. Chapter 8 contains the results of a pilot study, which was used to test the algorithm. Finally, chapter 9 provides in conclusions and recommendations.

2 The Refinery

A refinery has a very complex manufacturing process in which different kinds of units are used to transform crude oil into refined products like LPG, motor gasoline, kerosene, diesel and luboil. In order to keep the performance at a good level, some kind of preventive maintenance should take place. To set out the context of this problem, an introduction is presented in this report. It starts with a description of the processes occurring in a refinery.

The complete supply chain of the petroleum industry, from drilling till the supply to the customers, is too much to consider in this research. Our interest is focused on the processing of crude oil up to the blending into final products, so distribution and the following parts of the supply chain are left outside the scope. For a complete overview of the supply chain see Figure 2.1.

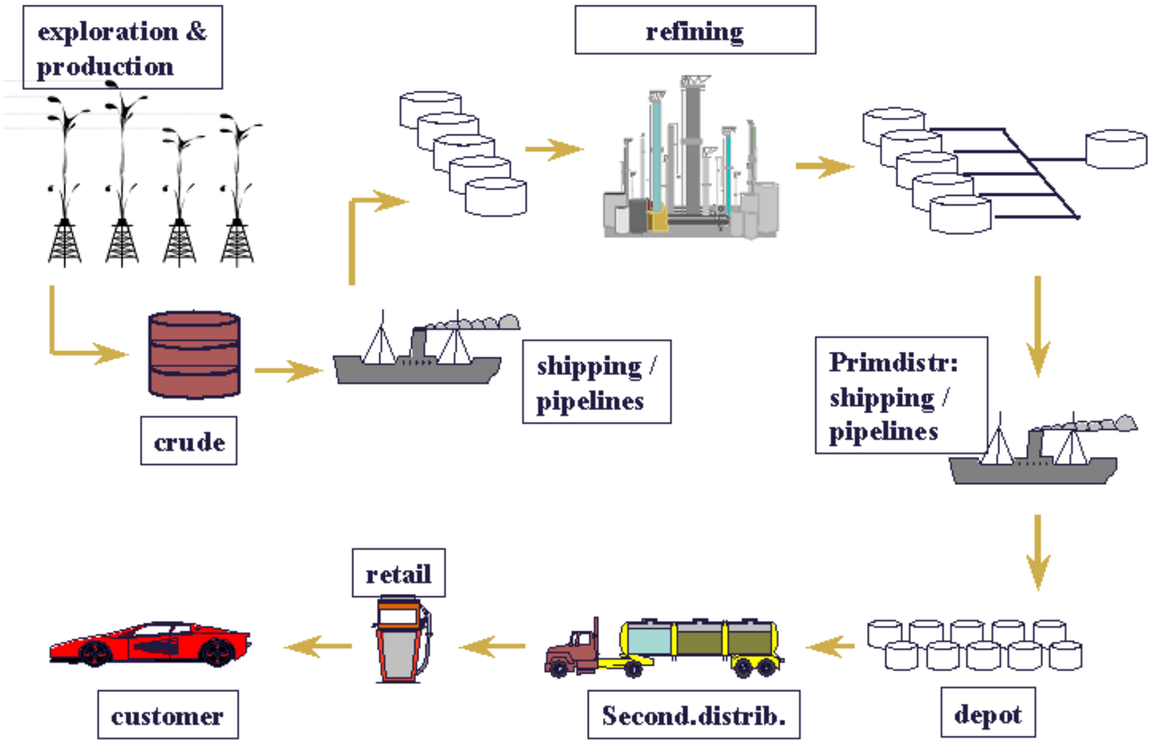


Figure 2.1: The oil supply chain.

Refining is the breaking down of crude oil into the products desired. These treatments take place on several factories, called processing units. They are possible because crude oil is a chemical compound consisting of many elements, each having their own boiling temperature. This separation process becomes more complicated since there are a lot of different crude oils, each having different properties dependent on the location where they were drilled.
The intention of processing is to physically separate the elements or change the structure of the molecules, since crude oils contain impurities such as sulfur, oxygen, nitrogen and certain metals that must be removed. Various separation techniques are applied to the crude oil and intermediate product processing steps. This is described in more detail in section 2.1. After these processing steps components are available which are ready to get mixed. In the next stage these components are blended into the final products, which will be discussed in section 2.2.

After the operational processes are described, refinery shutdowns will be introduced in section 2.3.

2.1 *Processing*

Since each refinery is different from the rest, this section will describe the main units. So, be aware that each refinery has a different topology and this is just an example.
In the refining process, various components of crude oil are separated by their boiling points. In general, the longer the hydrocarbon molecule, the higher it's boiling temperature. At the refinery, crude oil is first heated in the distillation tower, the crude distilling unit (CDU). The bottom of the tower will be heated and as the vapor passes up through the tower, heat is removed. This provides the gas flow upward and liquid flow downward. The tower contains trays where liquids, which have achieved their boiling points, can be drawn off. This is the most important process in a refinery. Figure 2.2 gives some idea how it works.

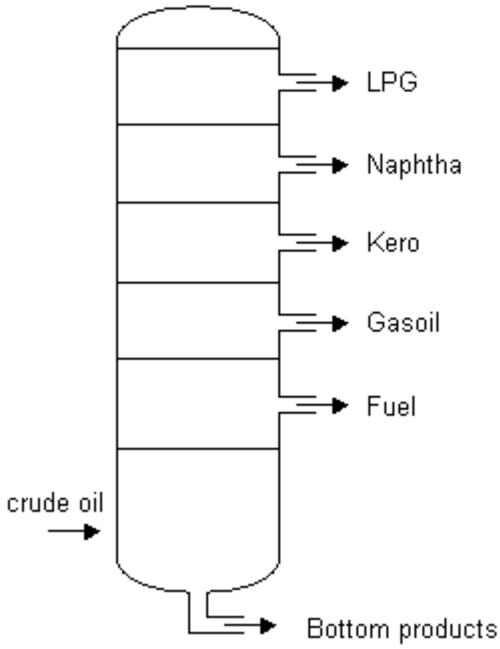


Figure 2.2: The crude distilling unit.

The liquid recovered from the very bottom of the crude tower is then subjected to a vacuum process, in which the pressure is lowered and the residue could be heated further at a lower temperature. This is possible because as pressure decreases, the boiling temperature for any given liquid also decreases. The unit, which performs this, is called the High Vacuum Unit (HVU). The process of high vacuum distilling is also known as vacuum flashing.

Another very important processing step is the cracking process. One of the results of this process is a reduction in the density; this means a volume gain. There are several kinds of cracking processes. In the catalytic-cracking process, oil is subjected to high temperatures and pressures in the presence of a catalyst. A catalyst is a substance that causes or enhances a reaction, but which itself is not changed in the reaction. Hydrocracking utilizes both a catalyst and hydrogen to process residuals or products in the middle-boiling range.

In order to remove sulfur from the materials, hydrogen is added (Hydrotreating and Hydro Desulphuriser Unit: HDU).
The catalytic reforming process (platformer) operates on naphtha from the crude tower. Unlike the cracking processes where large molecules are whittled down to smaller ones, the cat-reforming process merely rearranges the hydrocarbon molecules in the presence of catalysts, without actually breaking them down and altering their composition. The role of this process is to improve the octane number, which reforms the molecules from low octane naphtha in a high octane gasoline component.

Some compounds can be built from smaller molecules and other components can be made by breaking larger molecules. Some bonds of unsaturated hydrocarbons will open up under suitable conditions and join with other molecules of the same compound. This combining of molecules is known as polymerization. There are also some other processing steps like crystallization, adsorption, isomerization and alkylation.
To get an overview how complex the dependencies between the various units are, Figure 2.3 gives a flow diagram of the different flows in a refinery.

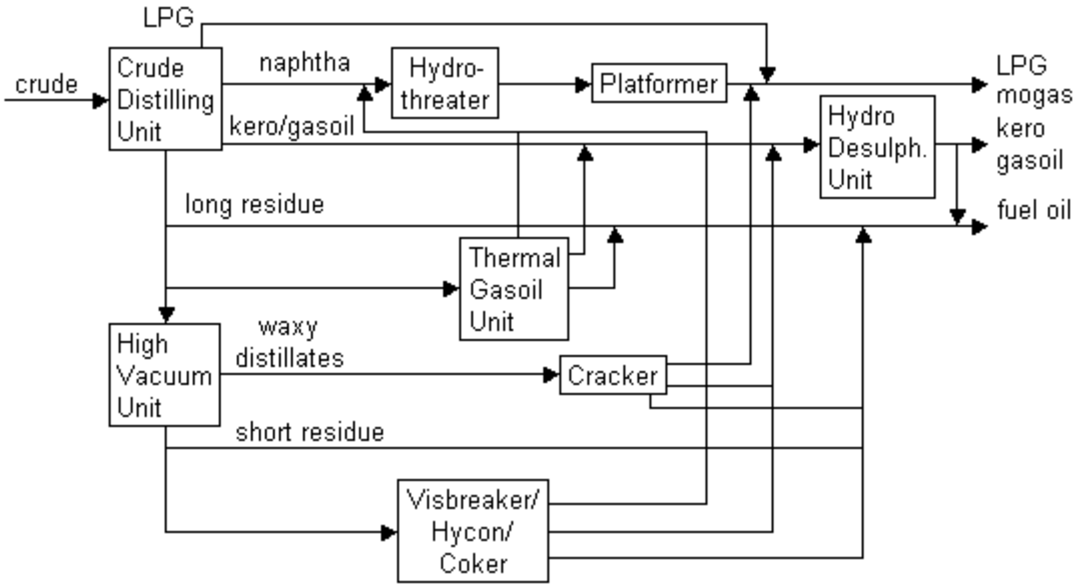


Figure 2.3: Flow diagram of a refinery.

There is not one distinct flow of components between the different processing units, but a wide variety of options makes the production processes quite dynamic. These dynamics are due to different crudes that a refinery can process. Along with different crudes come different properties for the crudes as well as for the produced components. Another reason why the process is dynamic, is a result of the different modes a unit can run in. Each mode has a characteristic yield table. In such a yield table is prescribed which components will be produced when a particular crude is processed in a particular mode on some processing unit. So, crude can be processed in different modes on the same unit. The produced quantities are expressed in a percentage. These percentages can be found in the yield table.

2.2 Blending

After the separation of the different compounds into components, they are ready to be mixed into final products. Each of these components has got specific properties (like octane, sulfur, etc.). Blending is the process of mixing the various components in the preparation of a product with required specifications.

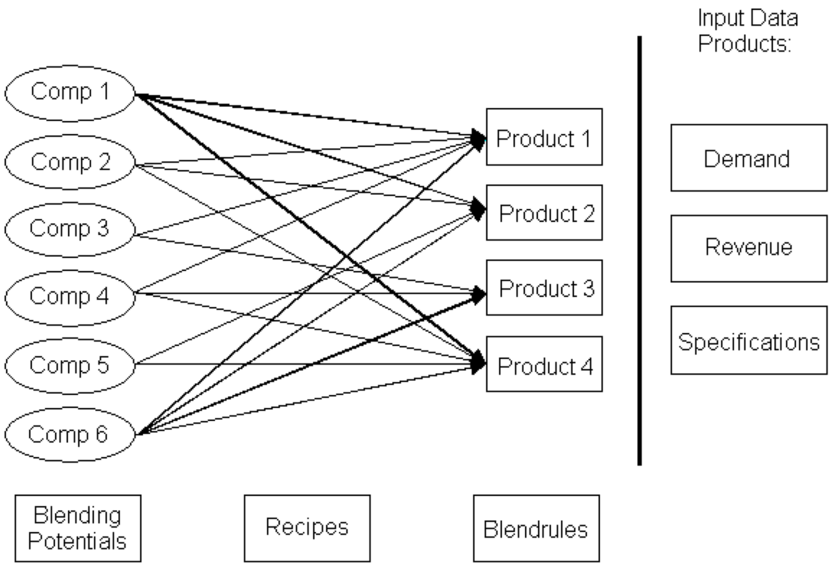


Figure 2.4: The blending process.

Each final product has properties, which should fall in a pre-specified interval (also called specs) like environmental restrictions and quality requirements. The product specifications determine which components and in which proportions the components should be blended. This process of blending should be performed as cheap as possible, so the cheapest components should be used and the final product should not have the best properties, only the specs should be met.
As can be seen in Figure 2.4, each component has blending potentials, which predefines for which products it can be used. Most of the time the properties of the final product are a weighted average of the properties of the different components. Due to chemical reactions that can occur, when mixing components, this is not valid for all properties. Non-linear blendrules or index rules are sometimes needed to calculate the product properties. This makes the process more complex.

Blending can be modeled by two essentially different methods. First there is blending by recipes. In this recipe is prescribed which components should be used and in which quantity in order to produce a final product. An experienced blender has for instance the knowledge that the mixture of components A, B and C in ratio 2:5:3 results in product D that fulfills product specifications. This kind of knowledge should be used to make the blendrecipes.
Then there is blending on specification. Software can be used to optimize the blending of components into products. Clearly blending on specification returns better solutions, since it give more freedom to combine components.

2.3 Refinery shutdowns

It is safe to say that all processing units sooner or later malfunction or wear out, so refineries require periodic maintenance activities. That means that most units will at sometime have a shutdown from operation to be able to perform replacing, repairing, cleaning and modifying on various internal parts. Most of these maintenance activities cannot be performed during the operation of the unit. That is why the unit must be closed while the activities are occurring. These maintenance periods are known in the industry as turnarounds or shutdowns. If there is a variable demand pattern, it would be advisable to conduct the shutdown during periods of reduced demand.
Turnarounds on major units in the refining industry are complex and expensive. If not well managed, they can result in excessive costs, business losses and so on. The probability a unit will breakdown generally increases in time from the last turnaround. Continuously postponing the maintenance activities will eventually result in a failure. Such unplanned shutdowns would take more time to repair and will be very expensive. To avoid this, shutdowns are usually scheduled in advance. That is why each unit is shut down once every four years.

2.3.1 Shutdown Cycle

During the shutdown cycle the focus is on three basic goals: quality, cost and time. A successfully planned and managed project is one that is completed at the specified level of quality, on or before the deadline and within budget. These specifications then form the basis for project control.
In order to manufacture a shutdown schedule, all kinds of activities have to be performed. The different phases are shown in Figure 2.5. This is a never-ending cycle. In this section the different phases in this cycle are discussed.

Planning

The first phase is the planning phase. Planning is the formulation of the duration and sequence of the activities to accomplish. The basic principle of this is to optimize the utilization of resources like people, material, time and so on. In a shutdown planning is determined what kind of preventive maintenance activities have to take place and what their duration will be. The workforce will be contracted as well.
This phase starts 12 till 18 months in advance. Hence, changes can occur because a planning is based on predictions. In reality more information becomes available, affecting the planning.

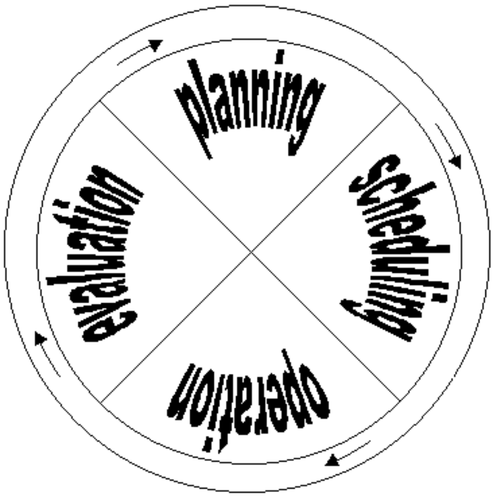


Figure 2.5: The shutdown cycle consists of four phases, after which a new cycle arises.

Scheduling

When this phase starts the initial planning is completed and is used as input for this phase. So when scheduling starts it is known in advance which units have to be shut down, what the shutdown duration for the units will be and how many workers are available and required to perform the turnaround. Since the planning is an iterative process, performed by different people, the information for this scheduling phase could change. Such information could concern unexpected businesses, new or changed legislation and so on. Consequently, once every month the schedule is revised. This makes the interaction between planning and scheduling iterative as well.

Scheduling the maintenance activities can be determined either before the scheduling of production or jointly with the scheduling of production. In the first case, maintenance periods are already known and fixed at the time when production has to be scheduled. The problem of scheduling production with this type of maintenance reduces to the problem and is often referred in the literature as *scheduling with machine availability constraints* because during maintenance periods the machines are not available for processing components.
In practice, the flow of components between the units of a refinery is scheduled separately from the shutdown. Two different departments take care of the two schedules. At first the shutdown schedule is produced and subsequently they look at the other processes in the refinery during a shutdown. This has the consequence that during the construction of the shutdown schedule, the possibilities to store components and to buy or sell them is not taken in consideration. Since a refinery consists of complex

processes, a careful coordination between maintenance and job processing will turn out a better overall schedule. Hence, simultaneously determining when to perform each maintenance activity and when to process each job results in a better efficiency during the shutdown period. There is no clue about the advantages that will be gained from integrating the two scheduling processes.

Previous sections outline the principles that have to be taken into consideration when the shutdown schedule is manufactured. The possibilities and limitations to change the current scheduling process are discussed in chapter 3.

Operation

During this phase the actual schedule is performed. Since the shutdown activities and the production are both performed at a refinery, although not at the same unit, this requires an accurate coordination. The heart of the control process is monitoring work in progress. It is the way of knowing what is going on and how actual reality compares to plan. Inspection is probably the most common way to monitor shutdown performance.

Evaluation

During the shutdown period some things can go wrong. This can have a lot of reasons. For instance when a unit is opened and it requires more maintenance than expected. In order to preventive these mistakes or surprises from happening in the future, an evaluation has to be performed. During this evaluation some feedback is given how well the previous three phases were performed. But more importantly, the output from this phase is very useful information for the next shutdown cycle. People from different refineries are gathered to exchange their knowledge and experience. This can be performed at the end (or beginning) of the shutdown cycle, but it can also be performed intermediately since shutdown scheduling is an iterative process anyway.

2.3.2 Business objectives

Shutting down one or several units in a refinery is a complex and expensive process. In order to perform this effectively a number of objectives can be identified. These business objectives are the ends that an organization sets out to achieve. Different objectives exist and will be discussed in this section.
First is the minimization of the total costs. Normally this means minimizing the duration of the shutdown, because the refining industry is highly capital-intensive. Cost is measured in two ways. One measurement of costs concerns lost income, since during the shutdown of a unit no materials are processed. A second measurement of costs deals with the actual expenses for the preventive maintenance, like labor costs and keeping extra inventory in the tanks for instance.
Besides this financial objective, there could as well be an operational objective. This means that the productivity should be maximized and the refinery should maintain its reliability. Reliability is the ability to adequately supply the contracts for refined products. The productivity can be maximized by keeping units in operation when it is needed most and scheduling it out of service when it is least needed or brings the least return.
The financial and operational objectives can lead to some contradiction, so all objectives may not be met at the same time. For example, when all units are shut down during the low load period might severely limit the reliability, while from a cost perspective this would seem a very wise thing to do since the revenues are not high. The solution to some, if not all, of these problems is found in coordination.

This thesis will look at the scheduling of preventive maintenance for a refinery to ensure its operations and reliability. Scheduling the shutdowns with some kind of smart procedure should result in an improved performance at a better level. All relevant aspects that have to be taken into consideration will be discussed in chapter 4, along with all aspects concerning a major shutdown at the refinery.

3 Problem description

Most of the shutdown scheduling is done manually, which has the advantage that the planner has got full confidence in the schedule he formulated. However, a lot of relevant factors are influencing the schedule process, which makes scheduling by hand not very efficient and difficult to reach the optimal solution. Automation would take away these disadvantages. Since most of the work on a turnaround is repetitive in nature, it is even logic to develop a standard turnaround-scheduling tool. Another main advantage of computerized network planning is its facility for rapid updating of the shutdown schedule in the light of alterations to planning constraints, either before or during the shutdowns. Besides the advantages of speed and accuracy, the increasingly complex systems make the scheduling process even more difficult. Hence, developing a scheduling tool is very advantageous for a refinery.

In this thesis an algorithm will be developed that will optimize a shutdown schedule, or find a nearly optimal schedule. There exist several objective functions to be optimized, as stated in section 2.3.1. At this moment the shutdown period is minimized. The idea behind this is the minimization of the total costs, since every day the refinery is running at normal productivity this will yield about half a million dollars. On the other hand, it should be of practical interest to directly minimize the overall costs during the shutdown period.

The possibilities and limitations during the shutdown scheduling process are discussed in this chapter. So all factors influencing the shutdown schedule are described, including the assumptions that will be made. But first, the current situation will be sketched.

3.1 Current Situation

The way a refinery determines the schedule for turnaround maintenance depends on the size of the refinery. When a refinery is small, the whole refinery can be shut down all at once for example. But when the refinery becomes larger and more complex, this is not possible anymore. One of the reasons is a limited amount of resources to perform the shutdowns. It would be advisable to shut down some units simultaneously while other units stay in operation in order to keep producing. Otherwise it would take too long before the refinery can produce anything and the reliability will decrease drastically. This would result in extensive costs.

…
CONFIDENTIAL
…

In order to find a feasible schedule, which can really be performed in practice, a lot of different aspects and information has to be taken into consideration. They are the subject in the remainder of this chapter. First the aspects concerning the actual shutdowns (like the activities in section 3.2, the time windows in section 3.3 and the workforce in section 3.4) will be discussed and afterwards the aspects and consequences for the production at the refinery are mentioned (in section 3.5 through section 3.7).

3.2 Shutdown activities

The activities during a shutdown are inspection, maintenance and projects. The public authorities dictate most of the inspection activities, since there is a stern legislation that prescribes when the units have to be checked. The wearing out and obsolescence of the units cause the maintenance activities. In order to prevent an unannounced breakdown of a unit, preventive maintenance takes place. These unannounced breakdowns cost a lot of money since nothing is arranged for such a situation. As a consequence most of the units are monitored and some of the units get time-based maintenance. This latter happens when a breakdown cannot be detected early. So this is more technical and operational. In the final determination of the activities for the turnaround it should be taken into account whether it is probable that a unit will break down or whether the activities could be performed outside the shutdown period (this is called a *scope challenge*). If a unit does not have to be shut down during the major shutdown period, it would be advisory not to take that unit under construction. Since it will cost a lot of valuable time and eventually money. These decisions are currently made based on experience. On an aggregated level, the activities that have to be performed on the units are known for the upcoming 10 years. The actual scope is eventually established for each unit, machine and item. It is important to perform the right activities; not too much but definitely not too few. This process is iteratively adapted since it is reviewed by different kinds of people.

For the problem in this thesis, we assume that the activities that have to be performed during the shutdown are known in advance. Determining whether they shall be performed (or even can be neglected) is out of the scope for this thesis. It is established that they just have to be done during the shutdown period.
Another assumption is that activities are non-preemptive. In practice the maintenance activities are not interrupted as well, since it will cost extra money.

3.3 Shutdown time

Let's first introduce two very important terms, which should not be mixed up. The activities performed during a shutdown of one unit need a certain number of days, which is called the shutdown duration of a unit (from the start of the first activity to the finish of the last activity. This time duration for the shutdown of each unit to perform the required activities is based on experience as well and can be split up in three different periods:

- pre-shutdown: make the unit empty,
- shutdown: cleaning, maintenance, inspection and
- post-shutdown and restarting the unit (sometimes these time periods are not equivalent, due to documentation etc.)

To give a clear overview how the actual shutdown fits into the complete shutdown project for one unit, Figure 3.1 gives an outline.

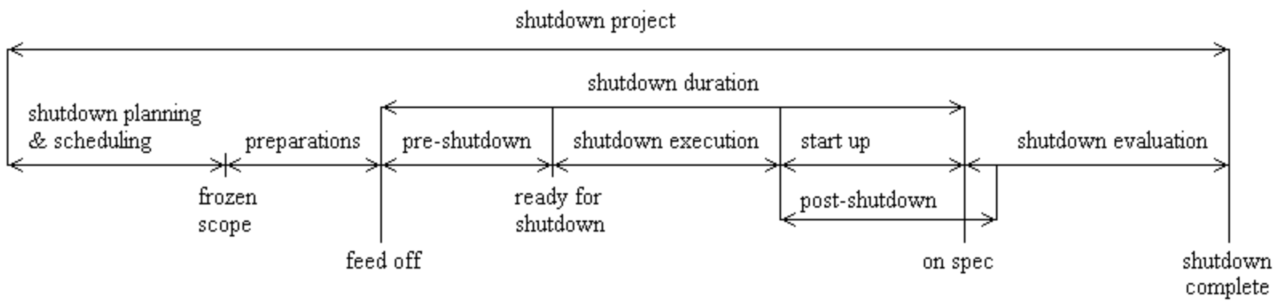


Figure 3.1: The shutdown durations over the entire shutdown project.

The problem assumes that the time in which the maintenance activities have to take place (shutdown duration) is predetermined and fixed when the schedule is made. Furthermore, the assumption is made that the shutdown times are never exceeded due to some interference. So all the equipment and machine components are present when they are needed and no delays occur. Otherwise some time should be reserved in the shutdown times.
The time to stop and to restart the units is also assumed to be included in the down time of a unit. Before all preventive maintenance can take place, the unit has to be empty. Looking at the way it is flushed or which chemicals are used to dissolve the dirt can accelerate this process. Restarting the units will take around 3 till 4 days. The required stop or restart times are always calculated in the downtime of a unit. This means that the shutdown times are independent of the last performed (or upcoming) activity at the processing unit before (or after) the shutdown.
The second important term, on the other hand, is the shutdown period. This is the time during which all units are shut down (from the start of the first shutdown to the finish of the last shutdown). How the time window for this major shutdown period is determined is discussed next.

On an annual basis is known which activities have to be done. For some of these activities a due date is defined, because legislations prescribe this or because the machine is on the edge of breaking down for instance. The refinery prefers to perform all turnaround activities during one major shutdown period. Even though this is not possible due to limitations on the amount of manpower, they are aiming for it. Consequently some activities have to be separated from this major shutdown. The units, which have to be separated. The first step for this process is found by looking at the dependencies between the units. This results in several units, which can be shut down independently from the rest. For instance the utility units are not maintained during a major shutdown, since they are not related to many other units. Subsequently a tool could be run in order to determine which of these clustered units are separated from the major shutdown. This tool gives them an impression what the effect is when one or more particular unit(s) is (are) shut down, what the alternatives in production are and how much this will cost the refinery.

The determination which moths this major shutdown takes place, depends on several factors:

- A communication with other producers or refineries should occur to tune whenever they can go down. They cannot shutdown simultaneously, because the supply in final products collapses otherwise. Whenever they shut down sequentially they could buy the final products from other refineries, so the customers do not notice the turnaround.
- Besides the required service, there will also be an insufficient amount of workforce when more than one refinery is performing a shutdown due to the extensive number of required personnel. The availability of contractors is also a major factor to decide when the turnaround can take place. Because the lack of specialized contractors can cause delays for a shutdown, enough contractors have to be available during the shutdown. This becomes a problem whenever a lot of activities have to take place simultaneously.
- The market situation for oil products is quite stable, so there is not really a period when the demand will be low. This could be the case for some products, but this is not a dominant factor for the major shutdown. The margin[1] on the final products on the other hand can change a lot during a year, so this factor is of more importance.
- The seasonality plays a role as well. The winter is a period that can cause a lot of trouble. When it freezes the shutdown activities are delayed. On the other hand, during the summer is a long holiday period. This period should be avoided whenever a shutdown is performed, because this would delay the turnaround immediately. But this period is a fixed period and it is known in advance. So the shutdown planning could anticipate on this (see the second issue).

The shutdown period is assumed to be known in advance. It specifies that the shutdowns of all units have to be scheduled within a specific shutdown period.

3.4 Workforce

One of the most important restrictions in shutdown scheduling is the availability of the amount of manpower. The amount of required personnel to perform the preventive maintenance is quite large. It is not unreasonable that a couple of thousand mechanics are working in the refinery to finish all maintenance activities on time. These activities require different skills. There are about 10 skills to be distinguished. The refineries hire contractors to do the work during the shutdowns.
Due to the large number of required workforce, some shutdowns cannot be done simultaneously and have to be moved to another moment during that year (as discussed in the previous section).

During the problem description, the assumption is made that the shutdown duration to perform the turnaround activities are the same for all workers of the same skill and it is known in advance how many men are required for a particular number of days to perform them. This means a deterministic scheduling problem. The number of required personnel for the shutdown activities is determined by experience. This number is specified for the different required skills.

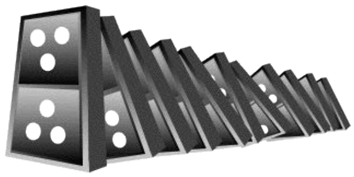
[1] The margin is calculated by looking at the cost price of crudes versus the revenue of the final product.

3.5 *Production*

The most important aspect in the refinery is production; making components and products and fulfilling customer satisfaction. So the scheduling process should be demand driven; meaning that there are (market) requests which products should be manufactured at the end of a period. In order to fulfil customer demands during a shutdown final products could be bought. When the refinery cannot produce or buy sufficient amounts of agreed on products, a fine will be laid on them.

3.6 *Domino effect*

When there is a shutdown, domino effects are triggered as other units (upstream and downstream) are forced to shutdown as well. For units downstream this is because there becomes a lack of supply of materials, which puts other units out of business. A first possibility to inhibit this consequence is the buffer storage between processes that was built up over time to prevent units from closing down immediately. The larger the available capacity in these storage tanks, the more time available to perform the maintenance activities, before more units need to be shut down. So larger storage tanks will provide a greater buffer, which will decrease the domino effect. But the size of the tanks is usually limited due to pressures on capital cost and the increased operating cost of carrying large inventories. Besides this domino effect on the units downstream, the shut down of a unit effects the units upstream as well. The components produced as feed for the closed unit cannot go anywhere, since the storage tanks could fill up and the feeds cannot be processed at alternative units. This comes down to a reduced production of components upstream.
There is another possibility to reduce the domino effect as well: the usage of alternative units to process the components that usually go to the closed unit. This is called re-routing. Doing a shutdown for the alternative units consecutively instead of at the same time, will keep the refinery in operation (perhaps at a lower productivity rate). Unfortunately, some processing units are unique.



These effects are formalized into dependencies between units. Different kinds of relationships exist and will be discussed in this section.
First, precedence constraints describe which shutdown activities should be performed before starting another activity. The activities can be subjected to different kinds of precedence relationships, which will be discussed in section 5.1.3.
A second kind of relationship is concerned with the fact that some of the shutdown activities on different units can not be performed simultaneously (see Figure 3.2).

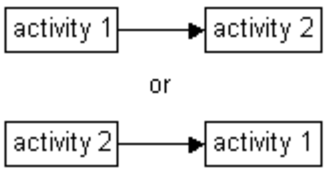


Figure 3.2: Some activities can not be performed simultaneously, the sequence in which they occur is of no importance.

With this kind of relationship, some freedom is involved in which sequence the units should be shut down. This can be restricted as well. One of the units could for instance have more impact on the manufacturing process when the activities exceed the planned shutdown time. This unit should be shut down first, to prevent unnecessary situations from happening.
The third and final important relationship is the performance of several shutdowns simultaneously. The graphical notation of the three different relationships that can occur between the shutdowns of units within the refinery is illustrated in Figure 3.3 (precedence, non-simultaneous (serial) and simultaneous (parallel) relationship respectively).

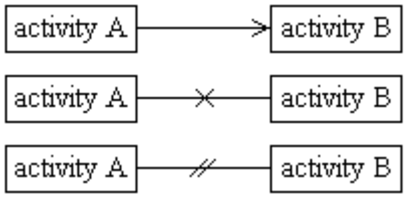


Figure 3.3: The graphical notation of the three most important relationships.

3.7 Extra production possibilities

Besides the processing and blending units, storage tanks are available for the different materials and components. Due to the high costs of components, the capacity of these tanks is limited. In about two days the full capacity of the tanks is completely used, which is a very short of a period compared to the shutdown durations (they could get till about 50 days). Since be a lot of investments are required in order to keep enough components on stock, this is not an option. But there are some possibilities to store some products at other locations, like keeping an emergency stock on a boat for example.

Import

Another possibility is to obtain intermediate blendstocks (components) or finished products in the open market to cover supply commitments during the period of the turnaround. Limitations occur on the imported components and involve in the first place the condition of the imported products. For instance, the intermediate components have to be at some particular temperature in order to get processed by the units. The temperature of the imported products is on the other hand much lower. The heat to warm up the components is taken away from other components through a heat exchange process. As a consequence, the temperature of all components is reduced. But the overall temperature has to be above some predefined number of degrees. In the second place, not all units are allowed to get imported components as feed due to quality constraints.

Export

The intermediate components can be sold at the market as well, although a limitation on the quantities to export exists. The components have to be sold for a relatively low price in accordance with the satisfied market situation, because the supply to export is much greater than the demand whenever such big quantities are involved.

Sometimes it could be more advantageous to export and to import components in stead of the storage of intermediate components, since the costs for storage are relatively high.

The same can be said about the (final) products, since they can be bought at other refineries in order to fulfill customer demands.

3.8 Objective

When taking this structure and all restrictions into mind, a method has to be developed which determines in which sequence the processing units should be shut down (for some predefined period of time) and what the flow of the materials should look like for the other units that are in process. The objective function to be optimized is the maximization of the profit. This is the most important aspect in real life, since a lot of lost money is involved when the refinery is (partially) shut down. Other aspects like reliability are dealt with in the shutdown scope.

3.9 Overview of the assumptions

The developed model has to be validated by comparing all restrictions with reality. In this section all assumptions are listed:

…

CONFIDENTIAL

…

So, the problem can be seen as a static deterministic scheduling problem, with a dynamic availability of the contractors. The shutdown schedule should formulate the shutdown activities for each day and the flow of materials in terms of quantities.

In the next chapter the aspects and assumptions, which are mentioned in this chapter, will be formulated into a mathematical model.

4 Mathematical Model

In order to find a schedule, which fulfills all restrictions from previous chapter, a mathematical model is formulated. The process of using mathematical models to help find good solutions to business problems is called mathematical programming. Its key feature is that the mathematical model is optimized. Mathematical programming has its roots in Operations Research back in the 1940s. It is one of the most successful and widely used methods to improve operations in real-world applications in a number of industries for scheduling, such as manufacturing or transportation. It is a very powerful technique to help solving certain types of management problems. The advantage of a mathematical model is that it is easy to understand.

A mathematical model represents the maximization or minimization of an objective function, subjected to constraints. An objective function is the function of one or more variables that one is interested in. The function can for example represent the cost or profit of some manufacturing process. Constraints are equalities or inequalities that describe restrictions involved with the minimization or maximization of the objective function and it defines the relationships between the decision variables.

This chapter will present a mathematical model for the shutdown scheduling problem, which is described in previous chapter.

4.1 Introduction

Before the actual mathematical model will be formulated it has to be determined which aspects have to be taken into account. Since this problem concerns a scheduling problem some information is known from the planning phase, which is done by Planstar and will be discussed in this section. So, the mathematical model will be simplified due to reasonable assumptions and due to the fact that some decisions are already made in the tactical planning process. In Appendix B the interaction and differences between planning and scheduling will be discussed.

In this thesis the problem is modeled as a deterministic problem. Since it concerns maintenance, it is very well possible that some stochastics can be used. Appendix C will give some more information about maintenance, and several approaches to model maintenance problems.

First the context of the problem will be summarized and next the information from Planstar will be clarified.

4.1.1 Processes at the refinery

In chapter 2 most of the important processes at the refinery are discussed. In this section they are recapitulated in order to find out which decisions have to be made in the mathematical model.

In a refinery the independent units have to process feeds into components. This can be done by running the unit in some mode. The possible modes dependent on the component that is used as feed. Each mode and feed combination has a different yield table. Hence some modes give components with a higher quality, but they produce at a lower productivity rate for instance. Besides the units, a refinery has several storage tanks. In a tank only one kind of component type can be stored one at a time. Another important aspect is the importation and exportation of components.

…
CONFIDENTIAL
…

4.2 Solving the mathematical model

There are no appropriate maintenance algorithms that can solve the shutdown scheduling problem. In Appendix C some well known maintenance models are discussed (Pintelon et al., 1997). They do not fit the mathematical model at all. Hence, it is not possible to make any adjustments to accomplish this. The next step in finding an appropriate solution technique is a literature study to more general scheduling algorithms.

In the literature most of the scheduling problems can roughly be divided into flow-shop, job-shop or open-shop problems. For refineries or chemical processing plants, this characterization is not sufficient since additional constraints and specific characteristics have to be taken into account, which result from features such as

- shared resources,
- continuous units and
- limited connectivity by piping networks.

The properties of the components cause additional constraints, such as limited storage time or limited storage capacity due to special requirements concerning how the material must be stored. The only possibility to solve scheduling problems is with the use of a mathematical programming technique.

More general mathematical programming techniques are represented in Appendix E. Such methods are used very often, because they do not use any specific knowledge about the problem. They are also applied in refineries, but mostly for a small part of the production process (like crude oil allocation, blending or cracking, Manne, 1956) instead of the entire production process (Joly et al., 2002).
The mathematical model is a mixed integer problem (MIP) and a number of solution techniques are developed for these kinds of problems. They are only not useful for the shutdown scheduling problem due to a number of limitations:

- the number of binary variables is too much and
- the required information concerning the stream of components at the refinery is very complex to put in a formal model. A lot of the required information is not available, like the maximum amount of import or export.

Hence, the problem has to be simplified. The first limitation could be dealt with in several different ways, as described in the last section of Appendix E. The second limitation however will be too restrictive, which cannot be tide over. Hence, production is not taken into account during the shutdown scheduling process.

When production is not taken into consideration at all, the shutdown schedule is worthless. In practice the shutdowns are related to production as described in section 3.1. To incorporate production into the shutdown scheduling process, a number of relationships have to be defined between the shutdowns of units. These relationships can be the result of a technical reason (as a result of a feed supplier between two units), but they can also be formulated with due observance of production (i.e. making a production schedule is easier when the relationship holds). These last relationships are not surveyed in the mathematical model, but they are the result of the simplification of the scheduling problem. On the other hand, these relationships are not so much detailed that the entire production process can be put in them.

In conclusion, this chapter will end with an overview of the aspects that will be taken into account in the shutdown schedule:

- the shutdown of the units
- the shutdown duration for these units
- the required number of contractors for the shutdowns
- the maximum number of available contractors
- the maximum number of contractors the refinery can handle
- relationships between the shutdown of the units

The next chapter will discuss the relationships and eventually the shutdown scheduling problem is translated into a project scheduling problem. Project scheduling has not been mentioned before. Hence, the next chapter will start with a short introduction to project scheduling. At the end of the following chapter the shutdown scheduling problem will be rewritten into a project scheduling problem.

5 Project scheduling

When the major shutdown period has to be scheduled without the actual production, the problem is reduced to scheduling the shutdowns of the units. The restrictions on the shutdown are related to relationships and to the contractors.

In this chapter the shutdown scheduling problem is reformulated to a project scheduling problem, where both aspects will be discussed. Since the theory about project management is broadly applicable, the context can be a project consisting of activities instead of a shutdown period with the shutdowns of different units. Section 5.1 will introduce project scheduling. Section 5.2 will introduce some standard problems in project scheduling, followed by the introduction of some solution techniques in section 5.3. At the end of this chapter the shutdown scheduling problem is discussed in relationship to project scheduling.

5.1 Introduction

Typically a project has the following characteristics:

- it can be broken down into separate activities, where each activity has an associated duration or completion time (i.e. the time from the start of the activity to its finish) and
- precedence relationships exist between the activities, which govern the order in which the activities may be performed.

The main objective in project scheduling is the minimization of the makespan, which is the time needed to complete all the activities. Other objective functions can be defined for the project scheduling problem of course. In this section the different relevant options for this thesis are discussed.

Network analysis is the general name given to certain specific techniques, which can be used for the planning, management and control of projects. It enables the user to take a systematic and quantitative approach for the problem of managing a project towards a successful completion. Moreover it can have a graphical representation, which means it can be understood and used by those with a less technical background and it provides a rapid communication and understanding.

5.1.1 Gantt chart

In the scheduling world it is customary to visualize a schedule by a Gantt chart. It is made popular by the industrial engineer Mr. Henry L. Gantt and Frederick W. Taylor in the early 1900s. These charts were the first means to graphically display project planning.
A Gantt chart consists of blocks representing the activities, with on the vertical axis the machines or resources required for an activity and on the horizontal axis the time at which the activities are performed. The length of a block denotes the duration of the activity.

When the Gantt chart is made, the user is able to see the total shutdown period and the sequence of the different shutdowns (the actual starting time for the shutdowns of each unit is presented).

The disadvantage of a Gantt chart is that it immediately gives a schedule, so the interdependencies between the activities are not clear. The Gantt chart is also limited in what it can do, since it does not show the restraints and control that the activities have on each other. Therefore Gantt charts can be implemented with additional features in software packages, such as the representation of the critical path in a different color, arrows indicating the relationships and so on.

5.1.2 Network model

In the previous section, it was mentioned that a Gantt chart has the disadvantage that the relationships between the activities are not represented. That is why the activities and the relationships between them are often depicted as a network model. Networks are made up of nodes and directed arcs. These diagrams provide a powerful visualization of the relationships among the various project activities. There are two kinds of networks:

- activity-on-arc (AOA): the arcs represent activities and the nodes represent events. The distinction between activities and events is subtle but important. Activities are processed and are associated with intervals of time over which they are performed. An event, in contrast with an activity, does not consume time or resources and is associated with a point in time. An event represents a stage of accomplishment; either the start or the completion of an activity (also called Arrow Diagramming Method).
- activity-on-node (AON): the nodes represent activities and the directed arcs represent precedence relations. This representation is easier to construct (also called Precedence Diagramming Method). The different kinds of relationships will be discussed in next section.

5.1.3 Relationships

The kind of relationship that is used the most is the restricted precedence relationship. This relationship prescribes that two activities have to be performed after one another. These precedence relationships can take the form of a chain, an intree or an outtree. An example of the different relationships is given in Figure 5.1.

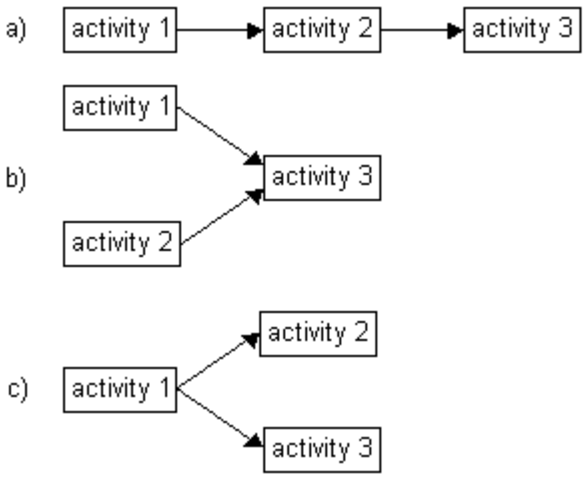


Figure 5.1: A chain (a), an intree (b) and an outtree (c) of successive activities.

However, real-life project scheduling applications often involve more complicated types of precedence relations, such as arbitrary minimal and maximal time lags between the starting and completion times of the activities and require more sophisticated objective functions. In that case we need generalized precedence relations (GPRs). GPRs enhance the capabilities of project scheduling models because they can be used to model a wide variety of real-life problem characteristics.

There are four different precedence relationships between activities, depending on the start and finish of the activities. The four distinguished relationships are:

1. finish-to-start (FS): the start of a shutdown for a particular unit depends on the finish of a shutdown of a preceding unit
2. finish-to-finish (FF): the finish of a shutdown for a particular unit depends on the finish of a shutdown of a preceding unit
3. start-to-start (SS): the start of a shutdown for a particular unit depends on the start of a shutdown of a preceding unit
4. start-to-finish (SF): the finish of shutdown for a particular unit depends on the start of a shutdown of preceding unit

The different relationships are illustrated in Figure 5.2.

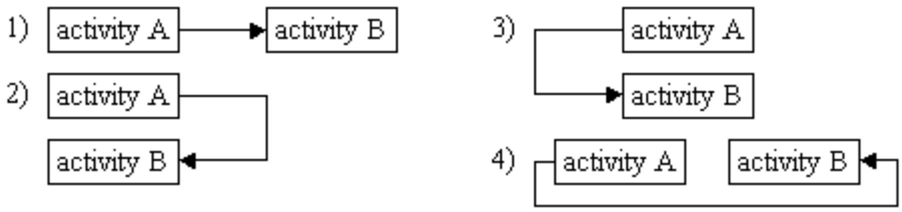


Figure 5.2: The different precedence relationships: finish-to-start, finish-to-finish, start-to-start and start-to-finish respectively.

Time lags

Besides the kind of relationship there can also be some time restriction between two shutdowns: they can be performed either immediately or after some lapse of time. Hence a minimum or a maximum time lag may be specified for these precedence relationships. A minimal time lag specifies that an activity can only start (finish) when the predecessor activity has already started (finished) for a certain time period. A maximum time lag specifies that an activity should be started (finished) at the latest a certain number of time periods beyond the start (finish) of another activity.
The minimal and maximal time lags between two activities *i* and *j* have the form:

$$s_i + SS_{ij}^{min} \leq s_j \leq s_i + SS_{ij}^{max}$$
$$s_i + SF_{ij}^{min} \leq f_j \leq s_i + SF_{ij}^{max}$$
$$f_i + FS_{ij}^{min} \leq s_j \leq f_i + FS_{ij}^{max}$$
$$f_i + FF_{ij}^{min} \leq f_j \leq f_i + FF_{ij}^{max}$$

The graphical representation for a maximum time lag is an interrupted line in stead of a straight line, which represents a minimum time lag.

5.2 Scheduling problems

In most scheduling problems a distinction is made between problems with and without renewable resources. Renewable resources (like manpower, machines, tools, space, etc.) are available on a period-by-period basis, that is, the available amount is renewed from period to period. So, resources may be available in an amount that varies over time in a (un)predictable manner. Resources may be shared among several jobs and a job may need several resources or resource types.

When the scheduling problem involves resources, this is called a resource-constrained project scheduling problem (RCPSP). Within these resource-constrained problems a distinction could also be made between single-mode or multi-mode[2]. In a sinlge-mode problem, the required resources are known in advance. Beside the single-mode RCPSPs, multi-mode resource constrained project scheduling problems (MRCPSP) exist as well. The activities in those kinds of scheduling problems poses different execution modes (reflecting different ways of performing the activity), each one possibly having a different impact on the duration of the activity, the cost associated with the activity and the required use of resources. Multiple activity modes give rise to several kinds of trafe-offs between

a) the duration of an activity and its use of resources (time/resource trade-off),
b) the duration of an activity and its cost (time/cost trade-off) or
c) the quantity and combination of resources employed by the activity (resource/resource trade-off or resource substitution).

[2] Be aware that a mode in the project scheduling context is something completely different than a mode in the refining topology.

Because different modes do not exist in the shutdown scheduling context, the remainder of this thesis will only look on single-mode project scheduling problems.

5.2.1 Resource constrained (RCPSP)

The resource-constrained project scheduling problem (RCPSP) involves the scheduling of project activities subject to precedence and resource constraints in order to minimize the total project duration. It can be seen as a combination for resource oriented and time oriented scheduling. The basic distinction between these two scheduling techniques is quite simple. For *resource oriented scheduling*, the focus is on using and scheduling particular resources in an effective fashion. Resource constrained scheduling should be applied whenever there are limited resources available for a project and the competition for these resources among the project activities is keen. For *time oriented scheduling*, the emphasis is on determining the completion time of the project given the necessary precedence relationships among activities.

5.2.2 Resource constrained with general precedence relations (RCPSP with GPR)

The RCPSP can get extended with other types of precedence relationships (finish-start, finish-finish, start-start and start-finish precedence relations). When there is just a minimum time lag involved in the relationships, this is called GRCPSP. But when arbitrary minimal and maximal time lags between the starting and completion times of activities are possible this is called RCPSP-GPR. Another notation for the problem is RCPSP/max.
The RCPSP-GPR is known to be strongly NP-hard and even the feasibility problem, i.e. the problem of testing whether a RCPSP/max instance has a feasible solution, is NP-complete (Bartusch et al., 1988).

A scheduling problem can also be denoted according to the classification scheme of Herroelen et al (1998), which is composed of three fields $\alpha|\beta|\gamma$. The first field α describes the resource characteristics of the problem. The second field β is used to describe the characteristics of the project activities. The third field γ denotes the objective function.

The RCPSP/max is denoted as $m,1|gpr|C_{max}$, in which

- m stands for the number of renewable resource types,
- generalized precedence relationships are defined,
- activities have arbitrary integer durations,
- activities must be performed in a single execution mode,
- activities require the resources in a constant discrete amount,
- all ready times are zero,
- no deadlines are assumed in the system,
- no preemption is allowed,
- no cash flows are specified in the project scheduling problem and
- the project makespan has to be minimized

The various time lags can be represented in a standardized form by transforming them to, for instance, minimal start-start precedence relations using the transformation rules given in Bartusch et al. (1988): The minimal and maximal time lags between two activities *i* and *j* have the form:

$$s_i + SS_{ij}^{min} \le s_j \quad \Rightarrow \quad s_j \ge s_i + d_{ij}^{min} \quad \text{with} \quad d_{ij}^{max} = SS_{ij}^{min}$$
$$s_j \le s_i + SS_{ij}^{max} \quad \Rightarrow \quad s_j \le s_i + d_{ij}^{max} \quad \text{with} \quad d_{ij}^{max} = SS_{ij}^{max}$$
$$s_i + SF_{ij}^{min} \le f_j \quad \Rightarrow \quad s_j \ge s_i + d_{ij}^{min} \quad \text{with} \quad d_{ij}^{max} = SF_{ij}^{min} - d_j$$
$$f_j \le s_i + SF_{ij}^{max} \quad \Rightarrow \quad s_j \le s_i + d_{ij}^{max} \quad \text{with} \quad d_{ij}^{max} = SF_{ij}^{max} - d_j$$
$$f_i + FS_{ij}^{min} \le s_j \quad \Rightarrow \quad s_j \ge s_i + d_{ij}^{min} \quad \text{with} \quad d_{ij}^{max} = FS_{ij}^{min} + d_i$$
$$s_j \le f_i + FS_{ij}^{max} \quad \Rightarrow \quad s_j \le s_i + d_{ij}^{max} \quad \text{with} \quad d_{ij}^{max} = FS_{ij}^{max} + d_i$$
$$f_i + FF_{ij}^{min} \le f_j \quad \Rightarrow \quad s_j \ge s_i + d_{ij}^{min} \quad \text{with} \quad d_{ij}^{max} = d_i - d_j + FF_{ij}^{min}$$
$$f_j \le f_i + FF_{ij}^{max} \quad \Rightarrow \quad s_j \le s_i + d_{ij}^{max} \quad \text{with} \quad d_{ij}^{max} = d_i - d_j + FF_{ij}^{max}$$

The scheduling problem can be formalized into a mathematical model. The bulk of the models and procedures designed for coping with PSP under resource constraints aim at scheduling project activities to minimize the project duration (minimize the makespan C_{max}). In the literature, other formulations are used for RCPSP-GPR. This other model will be discussed in the remainder of this subsection. The formulation is based on a refinery shutdown scheduling problem.

…
CONFIDENTIAL
…

5.3 Solution techniques

Different techniques for network analysis were developed independently in the late 1950's:
- Critical Path Method (CPM),
- Program Evaluation and Review Technique (PERT),
- Precedence Diagramming Method (PDM) and
- Metra Potential Method (MPM).

Table 5.1 (De Reyck and Herroelen, 1999) gives a classification of project scheduling problems and shows that the mentioned techniques are only applicable to project scheduling problems without resource constraints. These four methods are explained in Appendix F. Whenever resources do play an important role in the scheduling process, more specific techniques are used. These techniques will be discussed in this section.

	Single mode	
	No resource type	**Multiple renewable resource types**
Zero-lag FS	CPM/PERT	RCPSP
Min SS, SF, FS, FF	PDM	GRCPSP
Min + Max SS, SF, FS, FF	MPM	RCPSP-GPR

Table 5.1: The classification scheme for single-mode project scheduling problems.

CPM (Kelly and Walker, 1959) and PERT (Malcolm et al., 1959) were basically devoted to project scheduling under the assumption that required resources were available in sufficient amounts, and that the precedence relations implied strict precedence, meaning that an activity must be completed before another activity can be initiated. Numerous optimal and suboptimal procedures have been developed for project scheduling problems in which the assumption of sufficiently available resources is relaxed. These problems can be classified as time-cost tradeoff problems, resource-constrained project scheduling problems and resource leveling problems (Davis, 1973).

However, little research has been directed at extending the CPM to generalized precedence relationships.
Over the past few years, considerable progress has been made in the use of exact solution procedures for this problem type and its variants. But before these solution techniques are discussed, resource constraints are taken into consideration first.

5.3.1 Resource constraints

There are different ways to deal with resource constraints. The resource constraints could be ignored and an unconstrained network appears (Hackney, 1992). This relaxation can get solved with CPM, since it is extremely useful in giving insight and helping to produce heuristics for the full problems. The resource usage (or profile) over time can be analyzed to locate the difficult peak time period(s). Next, activities have to be moved earlier or later to spread the peak out over a longer time interval in a feasible manner, while delaying the project as little as possible. Even without fixed resource constraints, a scheduler tries to avoid extreme fluctuations in the demand for labor or other resources since these fluctuations typically incur high costs for training, hiring, transportation and management Such a peak in the resource requirements can be eliminated by hand or by adding precedence constraints to force that several activities do not occur simultaneously. All permutations of the different activities have to be checked. Another possibility is to use heuristic rules, but bottleneck dynamics (see section E.5.1) is by far the best performing heuristic to resource constrained scheduling.

Exact solution techniques exist as well. Demeulemeester and Herroelen (1992) presented an efficient depth-first branch-and-bound procedure for RCPSP. The nodes in the search tree represent the original project network extended with extra precedence relations, which resolve a resource conflict present in the project network of the parent node. Other branch-and-bound schemes exist as well (Fest et al., 1998).

5.3.2 Relationships

The RCPSP/max problem has been the subject of recent investigation within the OR community. The first treatment of generalized precedence relationships is due to Kerbosch and Schell (1975), based on the pioneering work of Roy (1962).
In most solution techniques, a distinction is made between time-feasible and resource-feasible schedules. A schedule is time-feasible if all relationships are satisfied and a schedule is resource-feasible if all resource constraints are satisfied. A schedule is feasible if both sets of constraints are satisfied.

It can be checked whether a time-feasible schedule exists. This is the case when the network model does not contain a cycle[3] of positive length (Bartusch et al., 1988). Therefore the distance matrix D = [d_{ij}] has to be calculated, where d_{ij} denotes the maximal distance (path length) from node *i* to node *j*. A positive path length from node *i* to itself indicates the existence of a cycle of positive length and, consequently, the non-existence of a time-feasible schedule.
The calculations of the distance matrix D can be done by standard graph algorithms for longest paths in networks, for instance by the Floyd-Warshall algorithm; The matrix $D = D^{(n+1)}$ is computed according to the updating formula $d_{ij}^{(v)} = \max\{ d_{ij}^{(v-1)}, d_{il}^{(v-1)} + d_{lj}^{(v-1)} \}$, where $D^{(1)} = [d_{ij}^{(1)}]$ with

$$d_{ij}^{(1)} = \begin{cases} 0 & \text{if } i = j \\ d_{ij}^{min} & \text{if } j \text{ succeeds } i \\ -\infty & \text{otherwise} \end{cases}$$

If $d_{ii} = 0$ for all *i* (the numbers in the diagonal of D), there exists a time-feasible schedule. The earliest start schedule is given by the numbers in the upper row of D ({$d_{1,1}, d_{1,2}, \ldots, d_{1,n}$}).

[3] a path <$i_s, i_k, i_l, \ldots, i_t$> in the network is called a cycle if s = t

To the best of our knowledge, the only exact solution procedure for the RCPSP/max presented in the literature are the branch-and-bound algorithms of Bartusch et al. (1988), Demeulemeester and Herroelen (1997) and DeReyck and Herroelen (1998). Because of the extreme complexity of problem m,1|gpr|Cmax, RCPSP-GPR instances of 30 activities becomes quite difficult to solve to optimality within acceptable computation times. Hence, approximate RCPSP/max procedures have been developed as well, though most work has focused on exact solution procedures. Quite a number of heuristics have been developed (Zhan, 1994; Neumann and Zhan, 1995; Brinkmann and Neumann, 1996; Franck and Neumann, 1996; Schwindt and Neumann, 1996; Hurink and Keuchel, 2001; Kämäräinen et al., 1999). Other techniques can also be used, such as constraint satisfaction problem solving (CSP).

Exact solution method

The systematic branch-and-bound procedure from Bartusch et al. (1988) conceptualizes the solution space as a network of temporal constraints of forbidden sets and reduced forbidden sets to designate resource conflicts and resource conflicts with a minimal number of activities respectively. The set of precedence relations will be extended by eliminating all reduced forbidden sets in an initial, time-feasible solution. More recent branch-and-bound approaches have retained the idea of extending a time-feasible solution by adding precedence relations (Dorndorf et al., 2000).

Demeulemeester and Herroelen (1997) based their solution technique on a depth-first solution strategy in which nodes in the search tree represent resource- and precedence-feasible partial schedules. Branches emanating from a parent node correspond to exhaustive and subset-minimal combinations of activities, the delay of which resolve a resource conflict at the parent node.
The algorithm of DeReyck and Herroelen (1998) is a hybrid depth-first/laser beam search branch-and-bound algorithm. The nodes in the search tree represent the initial project network, extended with extra zero-lag finish-start precedence relations to resolve a number of resource conflicts. Nodes that represent time-feasible but resource-unfeasible schedules lead to a new branching.

The branch-and-bound schemes can be accelerated with the use of lower bounds (Brucker and Knust, 2000)

Heuristics

Heuristic procedures for RCPSP/max represent either local search algorithms (like tabu search, simulated annealing and evolutionary computing), priority-rule methods or truncated branch-and-bound methods. Most heuristics are based on priority rules (Kolisch, 1996; Kolisch, 1995; Heilmann, 2000; Franck and Neumann, 1996). Such heuristics consist of two major components, a scheduling scheme and a priority rule. A scheduling scheme defines how a feasible schedule is constructed. A schedule is constructed by successively adding activities to a partial schedule, satisfying both the precedence and resource constraints, until a complete feasible schedule is obtained.
Heuristics for RCPSP/max can be classified into direct methods (Table 5.2) and contraction methods (Table 5.3). The contraction method schedules each cycle structure independently from other activities of the project.

Contraction method
Step 1: determine all cycle structures
Step 2: schedule each cycle structure separately
Step 3: contract the cyclic network
Step 4: schedule the acyclic contracted network by a priority-rule-based heuristic

Table 5.2: The framework for the contraction method.

In opposite to the contraction method the direct method schedules the entire cyclic network without a decomposition approach.

Direct method
Step 1: determine all cycle structures
Step 2: perform a temporary analysis for the cyclic network and the cycle structures
Step 3: schedule the cyclic network by a priority-rule-based heuristic

Table 5.3: The framework for the direct method.

In a serial scheduling scheme a feasible schedule is constructed in *n* iterations, in which the current partial schedule is extended by a single activity as follows. An available job *j* is selected by a priority rule and started at the smallest time point for which its execution does not violate the resource constraints. In a parallel scheduling scheme a feasible schedule is calculated in at most *n* iterations. At the beginning of each iteration, a decision point *t* is computed for the current partial schedule (PS) as the smallest time point at which at least one available job can be started without violating its release date as well as the precedence relations or resource constraints. For this decision point *t*, the set of eligible jobs E(PS,t) is computed containing all available jobs with $ES_j(PS) \leq t$. According to a priority rule, an activity *j* is selected and removed from E(PS,t). If it can start at *t* without violating the resource constraints it is scheduled at *t*. This is repeated until E(PS,t) is empty and the next iteration is started. The overall process stops when all jobs have been scheduled.

Franck and Neumann (1996) conclude that the direct method performs significantly better than the contract method, although the required computation time is higher due to the necessity of rescheduling steps caused by time-infeasibilities. No conclusion could be made whether the serial or parallel scheme was more efficient for the direct method. For the contract method, the most effective approach is to use the parallel scheme for scheduling the cycle structures and the serial scheme for scheduling the (acyclic) contracted project network.

While applying a scheduling scheme, priority rules are used in order to select the activity to be scheduled next. In the literature for RCPSP, a large variety of priority rules have been proposed. The most successful ones are presented in Table 5.4 (Klein, 2000).

Priority rule	Priority value p_j
Earliest starting time (EST)	min ES_j
Earliest finishing time (EFT)	min EF_j
Dynamic earliest starting time	min $ES_j(PS)$
Dynamic earliest finishing time	min $EF_j(PS)$
Latest starting time (LST)	min LS_j
Latest finishing time (LFT)	min LF_j
Minimum slack time (MST)	min LS_j-ES_j
Dynamic minimum slack time	min $LS_j(PS)$-$ES_j(PS,t)$
Greatest resource demand	max $d_j \sum_r u_{jr}$
Worst case latest starting time (WCLS)	min $\min\{LS_j(PS,i) \mid i \in A(PS)\}$
Worst case slack time rule (WCS)	min $LS_j(LB1)-\max\{ES_j(PS,i) \mid i \in E(PS,t)-\{j\}\}$
Random	max rand()

Table 5.4: Priority rules.

Except for the WCLS and the WCS rules, all rules are well known from other scheduling problems and are not explained. The WCLS rule has been introduced for RCPSP by Klein (2000) and extends the LST rule. The priority for activity *j* is computed as follows. Each available activity *i* (except for *j*) is scheduled temporarily. Subsequently, the resulting latest starting time of activity *j* is calculated, taking into account the adjusted residual resource availabilities. Afterwards, job *i* is released again. The smallest value obtained defines the priority value. The WCS rule extends the MSL rule by calculating a regret for not scheduling an eligible job $j \in E(PS,t)$ next (Kolisch, 1996). This regret corresponds to the smallest slack time of activity *j* in a partial schedule PS' resulting from selecting another eligible job instead of *j*.

The best performing priority rules are LST and WCS (Brucker et al., 1999; Neumann, 1997).

The way the serial and the parallel scheduling method have been described so far is termed as single-pass approach, i.e. one single pass and one priority rule are complied to derive one feasible solution. Multi-pass procedures, on the contrary, perform several single passes in order to generate a sample of several unique feasible solutions, where the best one is chosen. Basically, two different kinds of multi-pass methods can be distinguished: The multi-priority rule approach employs one scheduling scheme and different priority rules, while sampling makes use of one scheduling scheme and one priority rule. Different schedules are obtained by biasing the selection of the priority rule through a random device. Three different methods can be distinguished:

- random sampling assigns each activity in the decision set the same probability p_j
- biased random sampling biases the probabilities dependent on the priority values ν_j of the activities
- regret based biased random sampling assigns to each activity in the active set D_n a regret ρ_j, which compares the priority ν_j of activity j with the worst consequence in the decision set as follows

$$\rho_j = \begin{cases} \max\limits_{i \in D_n} \nu(i) - \nu(j), & \text{if objective function is minimized,} \\ \nu(j) - \min\limits_{i \in D_n} \nu(i), & \text{if objective function is maximized.} \end{cases}$$

Next a parameterized probability mapping arises

$$\psi(j) = \frac{(\rho_j + 1)^\alpha}{\sum\limits_{i \in D_n} (\rho_j + 1)^\alpha}.$$

By choice of the parameter α, the amount of bias can be controlled. Associated with an arbitrary large α will be no bias and thus deterministic activity selection on the basis of the employed priority rule while an α of 0 will give way for random activity selection.

5.4 Shutdown scheduling

The previous sections gave a literary overview of project scheduling. Since a lot of aspects concerning shutdown scheduling are in agreement with this, the shutdown scheduling problem is transformed in a project scheduling problem in this section. The two most relevant aspects are the relationships and the resources. Both will be discussed separately in section 5.4.1 and section 5.4.2 respectively.

5.4.1 Relationships

In shutdown scheduling three kinds of relationships are involved:

- precedence relationships connect shutdowns between units that specify which shutdowns must precede other shutdowns and by how much of a delay or by how much allowed overlap they should take place
- simultaneous relationships connect shutdowns between units that should be shut down simultaneously (also called a parallel relationship)
- non-simultaneous relationships connect shutdowns between units that should be shut down sequentially (also called a serial relationships)

As shown in section 5.1.3 a number of different precedence relationships exist, all with minimum or maximum time lags. That section gives a network representation for precedence relationships as well. A representation for the non-simultaneous and simultaneous relationships does not exist in the literature. In this thesis the three kinds of relationships are represented as in Figure 5.3.

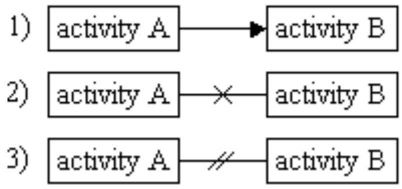


Figure 5.3: The precedence, non-simultaneous (serial) and simultaneous (parallel) relationships in a network representation.

Precedence relationships are already known in project scheduling. Parallel relationships however can be transformed into start-to-start relationships with a minimal and maximal time lag. Serial relationships on the other hand can be transformed into resource constraints. When two shutdowns have a serial relationship, a fictitious resource is introduced with an availability of 1 and a requirement for both shutdowns of 1 as well, such that they cannot both be scheduled at the same time. In the next subsection the resource requirements are elaborated in more detail.

5.4.2 Resources

Besides relationships, the contractors have to be taken into account as well. There are three kinds of restrictions concerning the contractors:

- the required number of contractors for the shutdown of a unit should be available during the entire shutdown duration. It is very well possible that a shutdown requires several skilled technicians. For each skill, a restriction is formulated.
- there is a maximum amount of workforce the refinery can handle.
- there is a maximum amount of workforce available from the market perspective.

Both restrictions can be rewritten in availability constraints. The first restriction is already written in an availability constraint and the second restriction can however be transformed. For this transformation, a new resource '*total*' has to be introduced. Each unit has a requirement for this resource equal to the sum of all requirements for the skilled technicians.

5.4.3 Limitations

The mathematical model of chapter 4 is very generally formulated. When production is removed and new relationships are introduced, there is a chance that not all shutdown situations can be modeled as a project scheduling problem. Take for instance the following situation

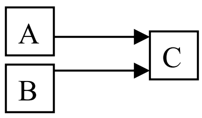


Figure 5.4: A situation, which could not be modeled as a project scheduling problem.

When unit C has to be shutdown after the shutdown of unit A ór after the shutdown of unit B, the problem can not be modeled as a project scheduling problem since the introduced relationships are not sufficient for this particular situation. This could be the case when unit A and unit B produce the same kind of components. Fortunately, these situations will hardly be present at a refinery because interchangeable units (like unit A and unit B) are not shut down simultaneously during the same shutdown period. If the relationships however do appear in a shutdown scheduling problem, they can be circumvented. The scheduler has to produce two schedules: one with a precedence relationship between the shutdown of unit A and unit C and one with a precedence relationship between the shutdown of unit B and unit C.

The shutdown scheduling problem can be translated into the scheduling of activities (the shutdowns of the units) which take some predetermined number of time periods (the shutdown duration) and have to take place under resource constraints and general precedence relationships. This is typically a RCPSP with GPR. The next chapter will give an algorithm to solve the problem. This algorithm will be based on the heuristics from the literature in section 5.3.2.

6 Algorithm

…
CONFIDENTIAL
…

7 Implementation

…
CONFIDENTIAL
…

8 Pilot Study

…
CONFIDENTIAL
…

9 Conclusions and Recommendations

The characteristics of the developed algorithm for resource-constrained project scheduling problems with generalized precedence relations are discussed and analyzed in section 9.1, as well as the generated shutdown schedules.

9.1 Conclusions

In this section a clear separation is made between conclusions that relate to the applicability of the algorithm for shutdown scheduling problems (9.1.1) and to the algorithm itself.

9.1.1 Shutdown Scheduling

One of the intentions of the research was to find out whether there was a need for a shutdown scheduling tool within the refineries. The use of a shutdown scheduling tool can be very beneficial. A number of reasons give rise to this conclusion:

1. It is useful to make the scheduling process more consistent.
2. Time is always saved when the scheduling process is performed automatically instead of manually. This is especially useful for refineries since the scope or other information for the shutdown period is updated regularly.
3. The schedules are improved in the sense that it will shorten the shutdown period and the total time until the shutdowns are completed.
4. The schedules do not contain any mistakes (except due to data errors) anymore, since all relations are being fulfilled. This will eventually save the refineries money, as the number of hired contractors is based on the shutdown schedule. If the schedule is incorrect, the hired number of contractors becomes too extensive or even insignificant.
5. The information concerning the shutdowns (like the relationships) is captured within a tool.

The following step is to find out what the role of ORTEC should be during the shutdown scheduling process. In this thesis an algorithm is developed that can schedule the units that have to be shut down during the shutdown period. This algorithm is implemented and a stand-alone application is made. Results from the pilot study show that the algorithm finds feasible schedules, which can even be optimal.

The final remark in this section is regarded to the use of the developed algorithm for shutdown scheduling. As explained in section 5.4.3 with an example, not all kinds of relationships could be translated in the desired relationships (a precedence, parallel or serial relationship). Sometimes it can be avoided dependent on the relationship that has to be modeled.

This application is however not ready to be used in practice. A number of developments have to take place in order to complete the research. First, some conclusions with regard to the developed algorithm are made.

…
CONFIDENTIAL
…

Concluding can be said that shutdown scheduling definitely has potentials for ORTEC to focus on. The developed algorithm has to be developed further. So performance will increase and all user requirements are granted. The application also has to get a user-friendly graphical user interface, such that the refinery can use all functionalities of the application optimally.

10 Bibliography

Anily, S., Glass, C.A., Hassin, R., The scheduling of maintenance service, Discrete Applied Mathematics, Volume 82 , Issue 1-3, pp. 27 - 42, 1998,

Ansoff, H.I., Corporate Strategy, McGraw Hill, New York, 1965

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., Vance, P., Branch-and-price: column generation for solving huge integer programs, Operations Research 46, 3, 1998

Bartusch, M. Möhring, R.H., and Radermacher, F.J., Scheduling project networks with resource constraints and time windows, Annals of operations research 16, pp. 201-240, 1988

Boer, De, R., Resource-constrained multi-project management -a hierarchical decision support system-, University Twente, 1998,

Brinkmann, K. and Neumann, K., Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: the minimum project-duration and resource-levelling problem, Journal of Decision Systems 5, pp. 129-156, 1996

Brucker, P., Knust, S., Lower bounds for resource-constrained project scheduling problems European Journal of Operational Research 149, pp. 302-313, 2000

Brucker, P., Drexl, A., Möhring, R., Neumann, K., Pesch, E., Resource-constrained project scheduling: Notation, classification, models, and methods, European Journal Of Operational Research 112, 1, pp. 3-41, 1999

Burke, E.K., Smith, A.J., Hybrid Evolutionary techniques for the maintenance scheduling problem, the IEEE Transactions on Power Systems, Volume 15 Number 1, pp. 122-128, 2000

Davis, E.W., Project scheduling under resource constraints – an historical review and categorization of procedures, AIIE Transactions 5 (4), pp. 297-313, 1973

Demeulemeester, E. and Herroelen, W., A branch-and-bound procedure for the multiple resource-constrained project scheduling problem, management science 38, pp. 1805-1818, 1992

Demeulemeester, E. and Herroelen, W., A branch-and-bound procedure for the generalized resource-constrained project scheduling problem, Operations Research, 45, pp. 201-212, 1997

DeReyck, B., Herroelen, W., The multi-mode resource-constrained project scheduling problem with generalized precedence relations, European Journal of Operational Research 119, pp. 538-556, 1999

DeReyck, B. and Herroelen, W., A branch-and-bound algorithm for the resource-constrained project scheduling problem with generalized precedence relations, European Journal of Operational Research, 1998

Dorndorf, U., Pesch, E. Phan-huy, T., A time-oriented branch-and-bound algorithm for resource-constrained project scheduling with generalised precedence constraints, Management of science, INFORMS Vol. 46 (10), pp. 1365-1384, 2000

Eglese, R.W., Simulated Annealing: A tool for operations research, European Journal of operational research 46, pp. 271-281, 1990

Fest, A., Möhring, R.H., Stork, F., Resource-Constrained Project Scheduling with Time Windows: A Branching Scheme Based on Dynamic Release Dates, TU Berlin Technical Report 596, 1998

Fowler, M., UML Distilled, Second Edition – A brief guide to the standard object modeling language, Addison-Wesley, 2000

Franck, B. and Neumann, K., Priority-rule methods for the resource-constrained project scheduling problem with minimal and maximal time-lags – an emperical analysis, Proceedings of the Fifth International Workshop on Project Management and Scheduling, 11-13 April, Poznan, pp. 88-91, 1996

Frost, D., Dechter, R., Constraint processing for optimal maintenance scheduling, An Electric Power Research Institute final report, February 1998

Gertsbakh, I.B., Models of preventive maintenance, North-Holland, 1977

Glover, F., Greenberg, H.J., New approaches for heuristic search: A bilateral linkage with artificial intelligence, European Journal of Operational Research 39, pp. 119-130, 1989

Grigoriev A., Klundert, van de, J., Spieksma, F., Modelling and solving the periodic maintenance problem, Working Paper of Faculty of General Science, Maastricht University, 2002

Hackney, J.W., Control and management of capital projects, second edition, McGray-Hill, Inc., 1992

Heilmann, R.: A priority-rule method for the multi-mode project scheduling problem MRCPSP/max; 7th International Workshop on Project Management and Scheduling, Osnabrück, 17. - 19. April 2000

Herroelen, W., Demeulemeester, F. and DeReyck, B., A classification scheme for project scheduling problems, Weglarz J. (Ed.), Handbook on recent Advances in Project Scheduling, Kluwer Academic Puclishers, 1998

Hurink, J., Keuchel, J., Local search algorithms for a single-machine scheduling problem with positive and negative time-lags. Discrete Applied Mathematics 112(1-3): pp. 179-197, 2001

Jacquet-Lagreze, E., Lebbar, M., A column generation model for a scheduling problem with maintenance constraints, Electronic Notes in Discrete Mathematics, vol. 1, Elsevier-Science, 1998

Joly, M., Moro, L.F.L., Pinto, J.M., Planning and scheduling for petroleum refineries using mathematical programming, Brazilian journal of chemical engineering 19 (2), pp. 207-228, 2002

Kämäräinen, O., Ek, V., Nieminen, K., Ruuth, S., A generalized resource constrained production scheduling problem: An evolutionary algorithm approach, Proceedings of EUROGEN99 Short Course on Evolutionary Algorithms in Engineering and Computer Science, Finland, pp 87-94, 1999

Karp, R.M., Combinatoriek, Complexiteit en stochastische algoritmen, Informatie Jaargang 28, nr. 29, pp. 722-733, 1986

Kelley, J.E., Jr. and Walker, M.R., Critical path planning and scheduling, Proceedings Eastern Joint Computing Conference 16, pp. 160-172, 1959

Kelly, A., Maintenance planning and control, Butterworths, 1984

Kerbosh, J.A.G.M. and Schell, H.J., Network planning by the extended METRA potential method, Report KS-1.1, University of Technology Eindhoven, Department of Industrial Engineering, 1975

Klein, R., Bidirectional planning: improving priority rule-based heuristics for scheduling resource-constrained projects, European Journal of Operational research 127 pp. 619-638, 2000

Klein, R., Project scheduling with time-varying resource constraints, International Journal of Production Research 38 (16), pp. 3937-3952, 2000

Kolisch, R., Efficient priority rules for the resource-constrained project scheduling problem, Journal of Operations Manangement 14, pp. 179-192, 1996

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

Koole, G., Lecture notes for course Optimization of Business Processes: Applications and theory of mathematical modeling, 2003

Lee, C.Y. and Chen, Z.L., Scheduling of Jobs and Maintenance Activities on Parallel Machines, Naval Research Logistics 47, pp. 145-165, 2000

Lübbecke, M.E., Desrosiers, J., Selected Topics in Column Generation, Technical report G-2002-64, GERAD, 2002

Malcolm, D.G., Roseboom, J.A., Clarck, C.E. and Fazar, W., Applications of a technique for R&D program evaluation (PERT), Operations Research 7 (5), pp. 646-669, 1959

Manne, A.S., Scheduling of Petroleum Refinery Operations, Cambridge, Massachusetts, Harvard University Press, 1956

Marwali, M.K.C., Shahidehpour, S.M., A probabilistic approach to generation maintenance scheduler with network constraints, Electrical Power and Energy Systems 21, pp. 533-545, 1999

Miller, R.W., Schedule, Cost and Profit Control with PERT – A comprehensive Guide for Program Management, McGraw-Hill, 1963

Milton, A., Mathematics in science and Engineering, Volume 26, Optimal Shutdown control of nuclear reactors, New York, Academic Press, 1966

Mobley, R.K., Plant Engineer's handbook, JKC, 2001

Morto, T.E., Pentico, D.W., Heuristic Scheduling Systems, John Wiley and Sons, Inc. 1994

Neumann, K. and Zhan, J., Heuristics for the minimum project-duration problem with minimal and maximal time lags under fixed resource constraints, Journal of Intelligent Manufacturing 6, pp. 145-154, 1995

Neumann, K., Heuristics and applications for resource constrained project scheduling with minimal and maximal time lags. 1997 IEEE International Conference on Intelligent Engineering Systems Proceedings, 1997

Niebel, B.W., Engineering maintenance management, Industrial Engineering/18, Second edition, Marcel Dekker, Inc., 1994

Oddi, C.A., Smith, S.F., A constrained-based method for project scheduling with time windows, Journal of Heuristics, 8, pp. 109-136, 2002

Pintelon, L., Gelders, L., and Van Puyvelde, F., Maintenance Management, Acco, Leuven/Amersfoort, 1997

Pistikopoulus, E. N., Vassiliadis, C.G., Arvela, J., Papageorgiou, L. G., Maintenance Optimisation for Multipurpose Production Plants - A System Effectiveness Approach. Industrial Engineering Chemical Research Vol. 40, pp. 3195-3207, 2002

Roy, B., Graphes et ordonnancement, Revue Francaise de Recherche Opérationelle, pp. 323-333, 1962

Roy, B., Algèbre moderne et théorie des graphes, Tome II, Dunod, 1970

Schwindt, C. and Neumann, K., A new branch-and-bound-based heuristic for resource-constrained project scheduling with minimal and maximal time lags, Proceedings of the Fifth International workshop on project management and scheduling, 11-13 April, Poznan, pp. 212-215, 1996

Strouvalis, A., Heckl, I., Friedler, F. and Kokossis, A.C., An Accelerated Branch and Bound Algorithm for Assignment Problems of Utility Systems, Computers Chemical Engineering 26, pp. 617-630, 2002

Shahidehpour, M., Marwali, M., Maintenance Scheduling in restructured power systems, Kluwer Academic Publishers, Norwell, Massachusetts, 2000

Simons, R., MIMI Brings OR Tools Together, MP in Action – The newsletter of Mathematical Programming in Industry and Commerce, December 1997

Tijms, H.C., Kalvelagen, E.M.F., Modelbouw in de operations research, Academic Service, Schoonhoven, 1994

Tijms, H.C., Ridder, A.A.N., Mathematische Programmering, 2001

Tsai, Y.T., Wang, K.S., Teng, H.Y., Optimizing preventive maintenance for mechanical components using genetic algorithms, Reliability Engineering and System Safety 74, pp. 89-97, 2001

Vanneste, S.G., Maintenance policies for complex systems, proefschrift, Katholieke Universiteit Brabant, Wibro dissertatiedrukkerij, Helmond, 1992

Wiest, J.D., Levy, F.K., A management guide to PERT/CPM, Prentice-Hall, Inc., New Jersey, 1969

Yao, X., Fu, M., Marcu, S.I., Gaucherand, E.F., Optimization of preventive maintenance scheduling for semiconductor manufacturing systems: models and implementation, Proceedings of the 2001 IEEE International Conference of Control Applications, 5 – 7 September, Mexico City, Mexico, 2001

Zhan, J., Heuristics for scheduling resource-constrained projects in MPM networks, European Journal of Operational Research, pp. 192-205, 1994

Appendix A ORTEC Consultants

This thesis project was conducted at ORTEC bv. In this section a short overview is given about ORTEC.

ORTEC is one of the largest independent consulting and software development companies operating in the field of operations research and management science. For many of their customers, ORTEC is synonymous with Professionals in Planning. Their mission is to support private companies and public institutions in their strategic and operational decision making by the development of Advanced Planning Systems using mathematical modeling and optimization techniques.

Five econometrical students of the Erasmus University in Rotterdam founded the company on the 1ste April in 1981. In the meantime ORTEC employs about 350 people and has 12 offices in 4 continents. The headquarter is in Gouda, but you will find them in Rotterdam, Amsterdam and Groningen as well. They are even expanding their horizon and settle in foreign countries, like US (Atlanta) and Germany (Siegen).

ORTEC provides its customers with enhanced decision-making processes and improved business operations. They accomplish this by their unique skills in combining Operations Research (modeling, optimization) and IT that leads to Advanced Planning Systems (APS) that are fit for purpose. Their efforts have lead to a diversity of software and off the shelf products used successfully around the world. The skills are exploited by selling:

- Standardized APS and components
- Product related services
- Tailor made APS and services.

The ORTEC International company consists of several wholly and partially owned companies of which ORTEC bv is the largest. Figure A.1 shows the organization structure of ORTEC International. ORTEC bv is divided in two main business units: logistics and financial.

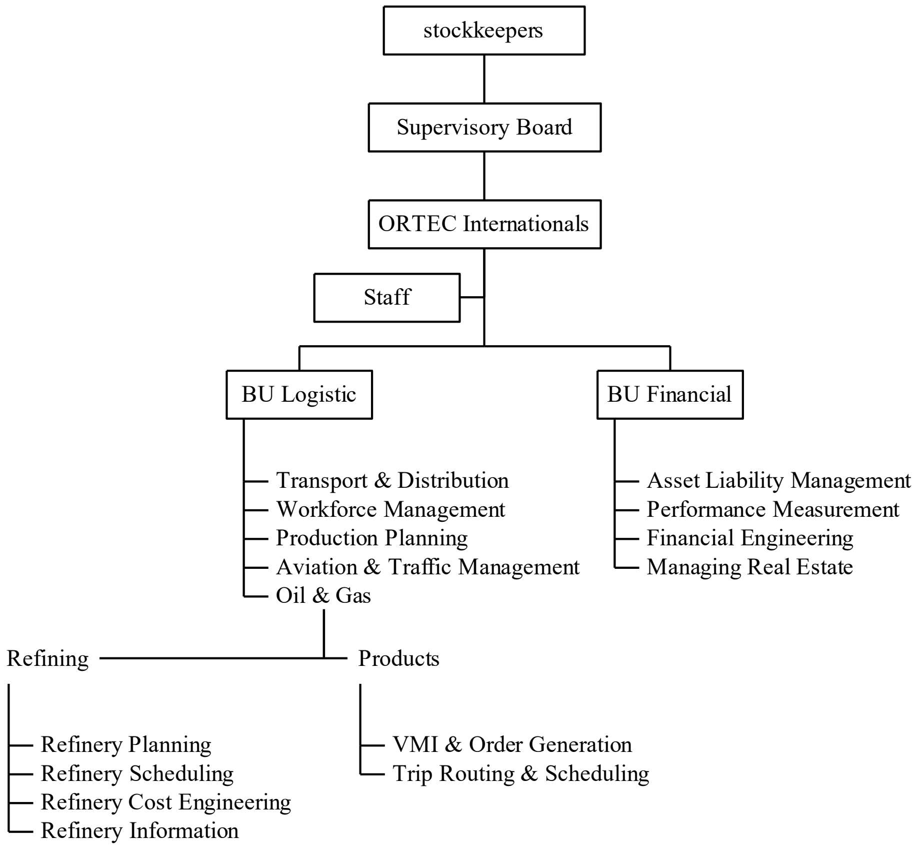


Figure A.1: The organisation chart of ORTEC and the position of the department Oil & Gas.

This thesis subject is of interest for the Oil and Gas (O&G) department. In Figure A.1 the siting of this department within the ORTEC company is shown. The department has about 30 employees working on problems varying from the winning of gas till the processing of oil and till the deliveries at the filling stations. The support of refinery planning and scheduling is an important part of ORTEC's activities in the field of production planning. A number of dedicated systems were developed for refineries all over the world. ORTEC also provides worldwide consultancy. The most important customers are Shell, BP and the NAM. This thesis deals with the refineries of Shell.

Appendix B Modeling

A model is a (simplified) description of a real-world process or phenomenon. The construction of the mathematical model is just the first step in the modeling process. In this phase, decisions concerning which details to model and which to leave out are of importance. Next, the resulting model is solved using one of the mathematical techniques. However, finding the right solution technique and executing it is not all that happens in the model-solving phase. To execute the model data has to be available. The necessary data collection and analysis is an important and time-consuming part of this phase. Next, the solution of the model has to be translated back to a system solution, as the model is a simplified representation of the system. Convincing the problem owners of the correctness and feasibility of the proposed solution is taken care of in the reporting phase. The complete process of modeling is stated in Figure B.1.

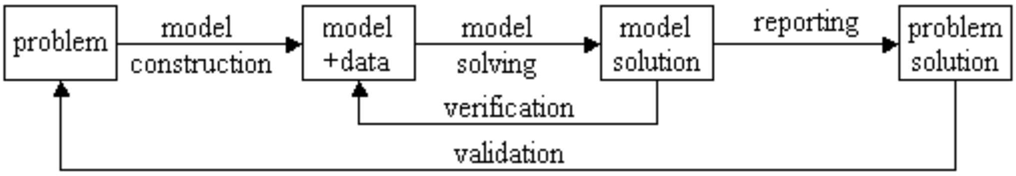


Figure B.1: The different modeling phases.

This is not a linear process by definition. In any stage of the modeling process there is feedback possible. An example of feedback is from the model-solving phase to the modeling phase; if it is discovered that the model is too hard to solve, simplifications to the model are required.

Simplifications are necessary to keep mathematical models manageable. As stated in chapter 3 a number of assumptions are made. The validation of these assumptions is of great importance to find a solution that can be implemented in real life. If the model does not represent reality, the solution is worthless. The model should describe a part of a system or a process and its interaction with its environment. An over-simplification of the problem for example would result in a model, which will not provide a good solution since much detail will be lost. However, if all aspects of the problem are modeled, the resulting mathematical model becomes too large and can not be solved in a timely way or the model will not have a solution at all, due to the complexity. So the assumptions should represent reality and which assumptions to take into account should be validated. Making the right assumptions is of great importance.

Besides the validation of the model, verification is crucial as well. The difference is quite simple; Verification means verifying that the implementation of the model is correct; validation means that it is checked that the model outcomes correspond with reality (up to a certain extent).

B.1 Planning versus scheduling

Since most scheduling problems are very complex (see section E.1), the mathematical model should be kept as simple as possible. This is done by eliminating some decisions, which are made in an earlier phase. Ansoff (1965) classifies the different levels of management decisions as follows:

- strategic decisions, dealing with the determination of long-range goals and the means to achieve these,
- tactical decisions, concerning the realization of the long-term goals and the management of the resources and
- operational decisions, dealing with the short-term planning and scheduling.

The lower the decision level, the shorter the horizon over which the decision takes effect.

Planning and scheduling are closely related as the decisions made at the planning level have a strong influence on scheduling. Scheduling is the process of organizing, choosing, and timing resource usage to carry out all the activities necessary to produce the desired outputs at the desired times, while satisfying a large number of time and relationship constraints among the activities and the resources. Planning, on the contrary, doesn't involve time issues. The differences are shown in Table B.1.

planning	scheduling
- tactical	- operational
- aggregate data	- detailed data
- describe large segments of production environment	- describe smaller segments
- planning horizon: 1 month to 1 year	- planning horizon of few hours to several days/weeks

Table B.1: Planning and scheduling characteristics.

Planning and scheduling take place over a hierarchy of time horizons. At the top level there is enterprise planning (strategic level): this is concerned with a company's market position worldwide and allocating capital investment over a period of several years. Below this, planning takes place at the tactical level for over a time horizon between 1 month and 1 year; this is for instance concerned with deciding which crudes to buy, how to process them, which products to sell and so on. At the bottom (operational) level detailed scheduling takes place within the refinery, which answers the question "What am I going to do next?". So planning is concerned with longer-term control issues compared to scheduling and is often classified at the tactical level. Scheduling deals with the operational short-term control. The difference can be seen in Figure B.2 (Simons 1997), which shows that planning takes place for a longer time horizon and has more degree of freedom. In scheduling on the other hand more detail is worked out with the information provided by planning.

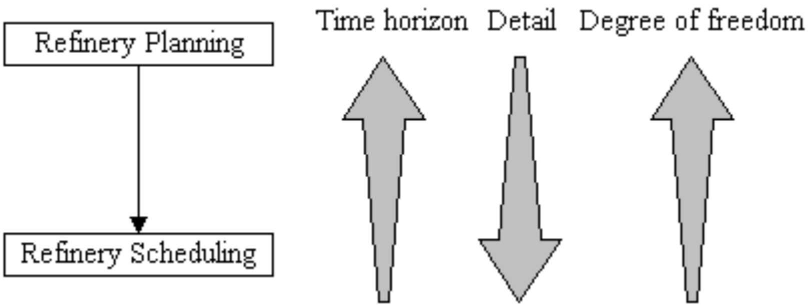


Figure B.2: The different levels involved in planning and scheduling.

Appendix C Maintenance

The maintenance of the means of production can be defined as –

> *"the set of activities to keep these means of production in the desired operating condition, or to restore them to this condition"* (Pintelon et al., 1997).

Maintenance usually involves repair in the event of a failure (a corrective action) or a preventive action. The costs incurred on this are normally a major portion of the total operating costs in most plants. Delays, product rejects, scheduled maintenance downtime and traditional maintenance costs are generally the major contributors to abnormal maintenance costs within a plant. Because of the excessive nature of maintenance costs, scheduling them will have a great potential for short-term improvement.

Industrial and process plants typically utilize two types of maintenance management: run-to-failure or preventive maintenance (Mobley, 2001).

Run-to-failure

The logic of run-to-failure management is simple and straightforward. When a machine breaks down then fix it. This is a reactive management technique that waits for machine or equipment failure before any maintenance action is taken. This is the most expensive method of maintenance management, since the maintenance costs are higher and the availability of process machinery is lower.

Preventive maintenance

Preventive maintenance is the total of all service functions aimed at maintaining and improving reliability performance characteristics and concerns itself with such activities as the replacement and renewals of elements, inspections, testing and checking of working parts during their operation. Preventive maintenance tasks are based on elapsed time of operation. Figure C.1 illustrates an example of the statistical life of a machine.

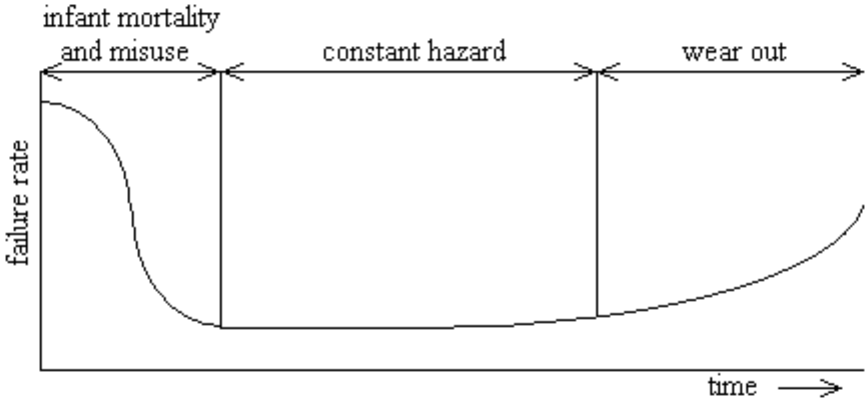


Figure C.1: The failure rate of equipment will change over time.

The mean-time-between-failure (MTBF) or bathtub curve indicates that a new machine has a high probability of failure during the beginning of operations, due to installation problems and misuse. Following this initial period, the probability of failure is relatively low for an extended period of time. Following this normal machine life period, the probability of failure increases sharply with elapsed time. In preventive maintenance management, machine repairs or rebuilds can be scheduled based on the MTBF statistic. This results in either an unnecessary repair or a catastrophic failure. Hence, a good preventive maintenance schedule can increase the availability of tools by trading-off between the planned unproductive down time versus the risk of much costlier unscheduled down time due to tool failures.

Predictive maintenance

Predictive maintenance is a condition-driven preventive maintenance program. Instead of relying on industrial average-life statistics (like MTBF) to schedule maintenance activities, predictive maintenance uses direct monitoring of the mechanical condition, system efficiency and other indicators to detect problems and prevent catastrophic failure.

Maintenance planning and scheduling will be the subject in this appendix. Starting with a short introduction in section C.1 why maintenance planning and scheduling are important, followed by planning and scheduling in section C.3 and C.4.

C.1 Why maintenance planning and scheduling is relevant

Analysis of historical costs have shown that unplanned or reactive repairs will cost about three times more than identical repairs that are well planned and scheduled. Therefore, creating an environment were all repairs and other maintenance activities are well planned and scheduled will substantially reduce overall maintenance costs. In addition proper planning will increase the number of hours that are available for production, which increases the production capacity and reduces product quality problems. Hence, planning and scheduling are vital ingredients for a successful maintenance program.

C.2 Solution techniques

Many papers on optimal preventive maintenance have been published (Vanneste, 1992). The first research on this subject appeared at the beginning of the fifties. During the search for relevant literature about shutdown scheduling or (on a broader context) about maintenance planning and scheduling, a lot was found but not applicable for the context in which the problem of this thesis plays a role. The literature can be subdivided into two categories:

- maintenance planning: this concerns the determination which activities have to be performed in order to assure the availability of a (manufacturing) system
- maintenance scheduling: this concerns the actual scheduling of the maintenance activities

Both categories will be discussed separately. Since the planning of preventive maintenance takes place before the scheduling process, this subject has less relevance and will be discussed shortly to give an overview of which aspects have to be taken into account in order to give a clear separation of the two subjects.

C.3 Maintenance planning

Many maintenance models have been proposed since the earlier works on systems reliability (Niebel, 1994). The majority of models have been focused on age-dependent maintenance policies or dependent on the equipment's technical states (like tool degradation for instance). That is why availability is one of the main characteristics from a manufacturing process.

Availability

Availability is widely recognized as one of the most important operability characteristics in process plants, which is mainly associated with equipment reliability, failure characteristics and maintenance policies. Hence, the principal objective for the planning of the maintenance activities is to maximize the availability of all equipment utilized by an industry or business. Because manufacturing systems rely on the units for revenue generation, it is highly desirable for the units to function above a minimum performance level for the duration of its operation life. Degradation comes from various factors, such as dirt, erosion and damage. Regular maintenance of a part may increase overall efficiency and prevent component failure but may also incur costly system downtime. An immediate question that arises is how often each maintenance job should occur. The answer determines how the machine can run at minimum expected maintenance cost while incurring minimum profit loss, which is caused by the efficiency degradation.
The reliability is usually expressed as the mean time between failures and of course, has a significant impact on the availability. As can be seen in Figure C.1, the reliability will decline in time since the failure rate will increase. There should be some optimum in availability, which can be expressed as

$$\text{Availibility} = \frac{T_{up}}{T_{up} + T_{down}} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTD}}$$

where

MTBF = mean time between failure
MTD = mean downtime

In maintenance planning a careful planning and a good coordination is perceived as essential to achieving an optimal trade-off between the cost of maintenance and the service reliability. Unfortunately, only a fraction of equipment conforms to this traditional bathtub curve. Sufficient historical data should be collected so that the appropriate failure-rate-versus-time curve can be made, then a preventive maintenance plan can be developed. The Weibull distribution is a useful probability density function f(t) for capturing the probability the unit will fail at running time t, because it provides meaningful parameters which can be used to represent the describe certain situations (Kelly, 1984):

$$f(t) = \frac{\beta(t - t_0)^{\beta-1}}{\eta^{\beta}} e^{-\left(\frac{t-t_0}{\eta}\right)^{\beta}}$$

where

t_0 = guaranteed life or threshold time-to-failure
η = characteristic life
β = shape factor: pattern of time-to-failure

The determination of the types and number of maintenance a manufacturing system should pursue plays a central role in planning.

C.4 Maintenance Scheduling

Maintenance scheduling is an important tool for increasing both the generating equipment availability and system reliability, while at the same time minimizing both operational and maintenance costs. There is extensive literature on preventive maintenance (Bruke and Smith, 2000; Tsai et al., 2001; Strouvalis et al., 2002; Pistikopoulos et al., 2002; Lee and Chen, 2000; Yao et al., 2001; Lübbecke and Desrosiers, 2002; Jacquet-Lagréze and Lebbar, 1998). However, approaches in literature usually are of a stochastic nature where a probability is used to describe the failure properties of some part.

Most of the literature on maintenance scheduling uses heuristics of one sort or another along with the optimization techniques. During years, five mathematical techniques have been used in searching for maintenance schedules: heuristics, integer programming, mixed integer programming, dynamic programming and evolutionary algorithms. Since these scheduling techniques are not applicable to maintenance only, they will be discussed in more detail in Appendix E.

Classifications of the different approaches to maintenance scheduling can be made on different bases, like Markov processes and lifetime distribution functions for instance (Gertsbakh, 1977).

Another problem with finding the right literature was the context in which it is stated. In the electric power industry the optimal scheduling of preventive maintenance of power generating units on a power plant, is a well-studied problem (Frost and Dechter, 1998). A typical power plant consists of several dozen generating units, which can be individually scheduled for preventive maintenance since they all produce the same only at different capacities. The general purpose is to determine the duration and sequence of outages of power generating units over a given time period, while minimizing operating and maintenance costs over the planning period subject to various constraints. This context is not comparable to the refinery context, it is not even close.

In the next subsections two approaches are brought up, which clarify that a lot of different contexts exist in which preventive maintenance scheduling plays a role.

C.4.1 Periodic Maintenance Problem

Whenever the operating costs associated with any given type of activity increases linearly with the number of periods elapsed since the last maintenance, the method wants to perform as much maintenance activities as possible (Grigoriev et al., 2002; Anily et al., 1998). Hence, most of the time the assumption is made that at most a single activity can be scheduled for maintenance during one period. So it is expected that a schedule is of a cyclic nature, since it consists of repetitions of a finite sequence. This problem is referred to as the Periodic Maintenance Problem (PMP).

Example Let a set of machines {1,2,3} operate over 7 time periods, maintenance will take one time period for all machines and the costs when not maintained are 10 for machine 1 and 2 and are 1 for machine 3. Consider the solution (1,2,1,2,1,2,3). This sequence of maintenance activities is to be read as follows: in the first period machine 1 is subjected to maintenance, in the second period machine 2 etc., until machine 3 is maintained in the seventh period. Then this sequence of maintenance activities is repeated. The cost of this solution can be computed as follows: machine 1 has operating costs of 0+10+0+10+0+10+20=50. The total cost for this solution is 121 and is in fact optimal.

Whenever the cycle length is not given, the optimal cycle length has to be determined. This is called Free Periodic Maintenance Problem (FPMP).

C.4.2 A Markov maintenance model

In a markovian maintenance model a manufacturing system is represented by different kinds of states. The transitions between the states are determined by the failure characteristics of the components and the operational and maintenance procedures in the event of a failure, in which the duration of a repair is a random variable following some kind of distribution function. According to the maintenance policy transitions between states may represent a failure, repair, preventive maintenance, etc. Assuming a continuous time representation, a continuous Markov chain[4] can be constructed, corresponding to a set of ordinary differential equations to describe the probability of each system state as a function of time and maintenance policy. So, the determination of the probabilities of each state and the performance of the system in each state, are taken into consideration for the evaluation of system effectiveness under different maintenance policies.

---

[4] A Markov chain is a discrete-state random process in which the evolution of the state of the process beginning at a time t (continuous-time chain) or n (discrete-time chain) depends only on the current state X(t) or X_n, and not how the chain reached its current state or how long it has been in that state. To be more precise, let's begin with discrete-time chains.

In a discrete-time Markov chain, X_{n+1} depends only on X_n , and not on any X_i, $1 \le i < n$

$$Pr[X_{n+1} = s_i \mid X_n = s_j, X_{n-1} = s_k, \ldots, X_1 = s_1] = Pr[X_{n+1} = s_i \mid X_n = s_j]$$

This equation is referred to as the Markov property.

Now consider a continuous-time random process in which the random variables X(t) change value (the process changes state) at times $\{t_1, t_2, t_3, \ldots\}$. If we ignore how long the random process remains in a given state, we can view the sequence $\{X_{t1}, X_{t2}, X_{t3}, \ldots\}$ as a discrete-time process embedded in the continuous-time process.

A continuous-time Markov chain is a continuous-time, discrete-state random process such that

1. the embedded discrete-time process is a discrete-time Markov chain and
2. the time between state changes is a random variable with a memory-less distribution. This means that the distribution of the time until the next state change is not a function of the time since the last state change.

Appendix D Mathematical Programming

This appendix contains some general concepts from mathematical programming (Tijms and Kalvelagen, 1994; Tijms and Ridder, 2001):

1. duality in linear programming
2. column generation
3. cutting plane

D.1 Duality in linear programming

Corresponding to any linear program, called the primal problem, there is another one called its dual.

$$\text{minimize} \quad \sum_{n=1}^{N} c_n x_n$$

$$\text{subjected to} \quad \sum_{n=1}^{N} a_{mn} x_n \geq b_m \qquad m = 1, \ldots, M$$

$$x_n \geq 0 \qquad n = 1, \ldots N$$

$$\text{maximize} \quad \sum_{m=1}^{M} b_m \pi_m$$

$$\text{subjected to} \quad \sum_{m=1}^{M} a_{mn} \pi_m \leq c_n \qquad n = 1, \ldots N$$

$$\pi_m \geq 0 \qquad m = 1, \ldots, M$$

If x_n is given as the primal variable, then π_m is the dual variable. In LP, a dual variable is also called a simplex multiplier. A one-to-one correspondence between the m^{th} dual variable π_m and the m^{th} primal constraint and between the n^{th} primal variable and the n^{th} dual constraint. π_m is a shadow price for item m and represents the increment in the objective value due to a small increase in item m.

D.2 Column generation

A column generation model is very effective for solving many parallel-machine scheduling problems. Each column represents a partial schedule on a single machine in which a subset of activities is scheduled. Subsequently a mathematical model is formulated in which m columns have to be selected, where m is the number of available machines. It has to be ensured that each activity is covered exactly once. The objective can be to minimize the total weighted completion time of all the selected partial schedules. This selection process can be restricted. This formulation is called a set partitioning (or set covering) problem (with side constraints).

How these columns are generated can be very different. This can be done iteratively for instance, whenever the number of columns becomes too large (this is typically the case). In order to generate new interesting columns it is efficient to generate them dynamically taking into account the dual values of the current solution.

D.3 Cutting plane

This technique cuts of some part of the solution space. The cut is such that it does not eliminate any solution to the original problem. For example, suppose we want to solve

$$\max\{(x,y) \in \{0,1\}^2 \mid 2x + 2y <= 1\}$$

The only feasible point is (0, 0). The LP relaxation is

$$\max\{(x,y) \in [0,1]^2 \mid 2x + 2y <= 1\}$$

and an optimal solution is at (1/2, 0). One cutting plane is $\{(x,y) \mid x + y <= 1/3\}$. A deeper one is $\{(x,y) \mid x + y <= 0\}$. When adding a cutting plane to the LP, a new relaxation is obtained and the solution is closer to the IP solution.

Appendix E Scheduling

Production scheduling is the process of finding a policy where for each moment is decided which order should be handled next. Different research communities made numerous contributions to these scheduling problems, mostly Operations Research and Artificial Intelligence. This appendix describes the basic background knowledge to tackle scheduling problems.

Practical implementations of scheduling techniques, which were developed at academic institutions, are rare. There is a gap between planning and scheduling. Currently, most scheduling tasks are still done by hand, while planning is performed with the assistance of a computer program. A scheduling system, on the one hand, could support the human scheduler. Replacing the scheduler, on the other hand, is not an option, because there is no guarantee that a system is able to deal with any unforeseen event as well as there is no guarantee that sufficient and accurate data are available. So a scheduler should always be present with the required knowledge and experience. A tool however is almost indispensable since it accelerates the scheduling process, no mistakes are made and the produced schedules are better.

Before any scheduling technique will be discussed, the computational complexity of scheduling problems is mentioned in section E.1. The different scheduling techniques will be classified in section E.2 and explained in more detail in the following sections E.3 till section E.6.

E.1 Complexity-theory

Models can be complex in different ways: their size can be large, the time to build an algorithm can be long and the time to execute an algorithm can vary a lot. Computational complexity concerns the running times. Different classifications are introduced to indicate the complexity (as shown in Figure E.1).

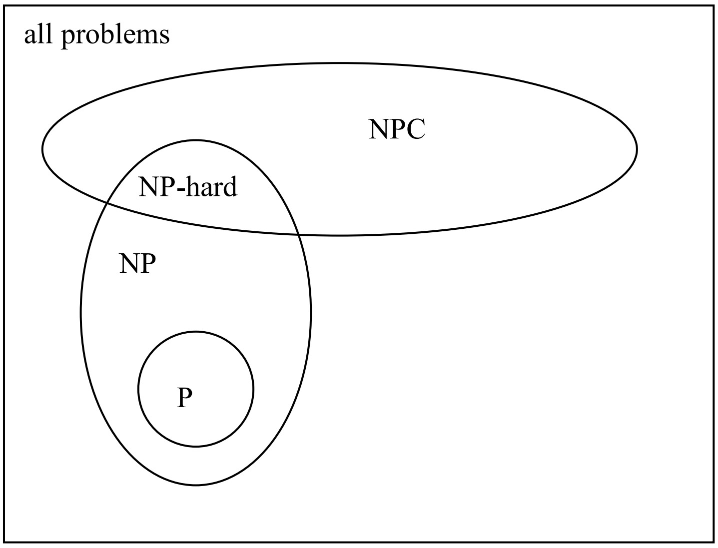


Figure E.1: The different classifications indicate the computation complexity of a problem.

A problem is said to be polynomial (or in the class P) if it can be solved in polynomial time in its size, otherwise it is classified as NP. A problem is said to be non-deterministically polynomial (or in the class NP) if an answer can be verified in polynomial time. It is obvious that $P \subseteq NP$, but there is no proof that $P \neq NP$ or $P = NP$. A problem X is NP-hard if any problem in NP can be transformed to X in polynomial time. A problem is called NP-complete (or NPC) if it is NP-hard and NP as well. Because NP is such a large class of problems, most NP-hard problems will in fact be complete. NPC is the intersection of NP and NP-hard.

E.2 Scheduling techniques

The most straightforward algorithm to find the optimal solution is complete enumeration (i.e. check all solutions). Since most scheduling problems belong to the class of NP-complete problems, it was often argued that only the use of problem specific algorithms can lead to efficient solution procedures.
Besides exact algorithms (like mathematical programming (in section E.3) and constrained programming (in section E.4)), a second class of algorithms is defined for optimization problems: heuristics and approximation algorithms (in section E.5 and section E.6). Exact algorithms produce optimal solutions and are reliable, but for most realistic problems their running time grows exponentially when it becomes larger in size. Approximation algorithms generally produce solutions in relatively little computation time, but the solution might not be optimal. In the earlier times much emphasis was given to heuristic methods because of their flexibility and simplicity. Artificial Intelligence (AI) is also operative on the development of approximation techniques for scheduling. It is difficult to generalize most AI techniques, since they are designed according to the particular characteristics of a designated problem. The literature indicates that, of all the possible intelligence techniques, evolutionary algorithms are the most suitable AI technique for scheduling.

E.2.1 Production and maintenance simultaneously

Most literature on scheduling assumes that machines are available all the time. In practice, a machine may not always be available in the scheduling horizon, for example, due to machine breaks (stochastic) or preventive maintenance (deterministic). During a maintenance period, a machine is not available for processing any jobs. Therefore, a more realistic scheduling model should take the associated machine maintenance activities into account as well, next to the production. Scheduling problems with limited machine availability have been studied to a less extent, as discussed in Appendix C.

E.3 Mathematical Programming

For scheduling problems, mixed integer formulations are widely used either with linear constraints and a linear objective function only (MILP: mixed integer linear program) or with non-linear constraints and/or a non-linear objective function (MINLP: mixed integer nonlinear program). In very rare cases, pure linear problems with real valued variables only (LP: linear program) arise and sometimes all variables are required to be integer-valued (IP: integer program). Table E.1 gives an indication of the different models.

LP	IP	MILP
minimize cx subjected to $Ax \le b$ $x \ge 0$	minimize cx subjected to $Ax \le b$ $x \in \{0, 1\}$	minimize $c(x + y)$ subjected to $Ax \le b$ $Dy < e$ $x \ge 0$ $y \in \{0, 1\}$

Table E.1: An overview of the differences between a linear program (LP), an integer program (IP) and a mixed integer linear program (MILP).

In this section a brief explanation will be given for most of the used techniques to solve the programs stated in Table E.1 (Tijms and Kalvelagen, 1994; Tijms and Ridder, 2001; Shahidehpour et al., 2000). The techniques that will be mentioned are linear programming (E.3.1), branch-and-bound (E.3.2), decomposition (E.3.4) and dynamic programming (E.3.5).

E.3.1 Linear Programming

It is unusual to formulate a scheduling problem in a pure linear model. Another option is to reformulate (or relax) a MILP problem to an LP problem. The most widely applicable method of linear programming is the simplex method.

The simplex method

The simplex method is invented by George Dantzig (1947) and described by Charnes and Cooper in their *Introduction to Linear Programming*, 1953. This method is widely used because it generally converges to an optimal solution in an acceptable amount of computer time.
The idea is to start at a corner from the solution space (a basic feasible solution *x*) and find another basic feasible solution whose cost is less. This is done by *pivoting*. The method moves the decision variables *x* along an edge from one corner of the solution space to an adjacent one, each time either lowering the cost or keeping it constant. It is guaranteed to converge to an optimal value of the cost if an optimum exists since the cost never increases and there are a finite number of corners.

Linear relaxation

A mathematical model with a large number of binary variables is hard to solve. To circumvent this problem, the restriction can be changed. Take for instance the following restriction

$$x_{pst} \in \{0,1\}$$

This restriction can get relaxed to

$$0 \leq x_{pst} \leq 1$$

Next the linear relaxation can be solved. The optimal solution of the linear relaxation is then used to obtain a sub-optimal solution of the original mixed integer problem. To that purpose the fractional values should be rounded. Several rounding strategies exist to perform this.

E.3.2 Branch-and-bound

Solutions for MILP's are mainly addressed through applications of the branch-and-bound algorithm. The algorithmic efficiency relies on the selection criteria for candidate problems, bounding, pruning and branching.

Generally the branch-and-bound algorithm works as follows: First, the linear relaxation of the problem is solved to obtain a partial solution. Based on this partial solution, branching only occurs on variables that are required to be integers while they get some real value in the partial solution. Every new branching instruction creates two new sub problems involving this integer variable. Each sub problem will generate a solution or leads to further branching. The branching process is illustrated with the use of an example.

Example Suppose that we obtain $x_j^* = 4.3$ by the simplex method as optimal value for the integer variable x_j. Since the variable is supposed to be an integer, two mutually exclusive and exhaustive sets are established: satisfying $L_j \leq x_j \leq 4.0$ and satisfying $5.0 \leq x_j \leq U_j$. Using this observation, two new sub-problems are created from the original problem.

During the branch-and-bound process, one attempts to eliminate a sub-problem by determining a lower bound on the objective value (for a minimization problem) for all sub-problems. If one could obtain a reasonably good schedule through some clever heuristic before starting the branch-and-bound algorithm, it may be possible to eliminate many sub-problems. By repeating the same process, this method will generate the optimal solution, since all interesting solutions are analyzed. The branching process can be represented in a search tree (as shown in Figure E.2). The search tree of the branch-and-bound method consists of a simplification of the original problem at the root node (top level) and along the way more restrictions are added (top-down).

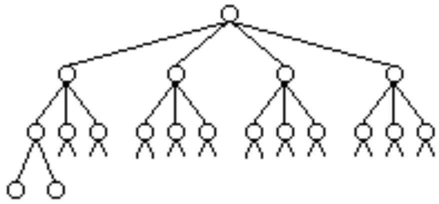


Figure E.2: The search tree for the branch-and-bound method.

For a scheduling problem the branching process may be based on the fact that schedules are developed starting from the beginning of the schedule: There is a single node at level 0, which is at the top of the tree. At this node none of the jobs have been put into any position in the sequence. With n jobs there are n branches going down to n nodes at level 1. Each node at this level corresponds to a partial solution with a specific job in the first position of the schedule. At each node at level 2, the jobs in the first two positions are specified, etc.

Branching rules

Branching rules define how the search tree is walked through and they can range from breadth-first to depth-first. In which the first strategy is to expand all possible branches from a node in the search tree before going deeper into the search tree and the later is to expand the deepest node first (and backtrack, as necessary). For example, suppose the binary variable x_j is branched. The breadth-first rule would solve the LP's for each of the two cases ($x_j = 0$ and $x_j = 1$). The depth-first rule would solve the case with $x_j = 0$, postpone the complementary case ($x_j = 1$) and goes deeper into the search tree.

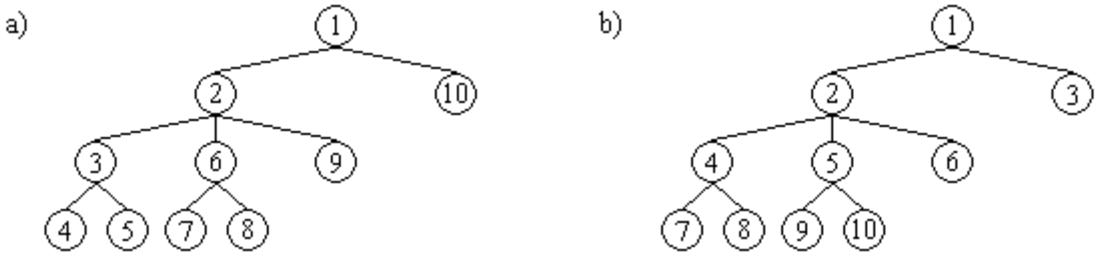


Figure E.3: The different search strategies: a) depth-first and b) breadth-first.

A best-first search is to select the sub-problem that seems most promising.

E.3.3 Enhancements branch-and-bound methods

There are a few different enhancements possible for the branch-and-bound method. These enhancements make use of the techniques from mathematical programming described in Appendix D.

Branch and price

Branch-and-price (Barnhart et al., 1998) is a branch-and-bound algorithm, where LP relaxations at every node are solved by delayed column generation. In a delayed column generation only a subset S of columns is considered, at every iteration. The problem with only a subset of all columns is called the restricted master problem. In every iteration of the algorithm, first the restricted master problem is solved. Next the so-called sub-problem is solved (called the pricing problem). In solving the sub-problem, a set S of columns with the lowest reduced cost with respect to the optimal dual vector π of the master problem is identified. If no column with negative reduced cost can be found, then the algorithm stops since π is an optimal dual solution to the original problem. Otherwise, we append columns in S to the restricted master problem and the entire procedure is iterated.

Branch and cut

Branch-and-cut (Barnhart et al., 1998) is a hybrid of branch-and-bound and the cutting plane method. The basic idea of branch-and-cut is simple. Classes of valid inequalities are left out of the LP relaxation because there are too many constraints to handle efficiently and most of them will not be binding in an optimal solution anyway. Then, if an optimal solution to a relaxation is infeasible for the original problem, a sub-problem (called the separation problem) is solved to try to identify violated inequalities in a class. If one or more violated inequalities are found, some are added to the relaxation to cut off the infeasible solution. Then the relaxation is re-optimized. Branching occurs when no violated inequalities are found to cut off an infeasible solution. So, branch-and-cut allows separation and cutting to be applied throughout the branch-and-bound tree.

Beam search

Beam search is like a type of branch-and-bound procedure, but instead of waiting to throw away a part of the tree that is guaranteed useless, parts of the tree that are likely to be useless are thrown away as well. On a certain level in the search tree, it keeps only a maximum number of search nodes for expansion. This number is referred to as the beam width and isdenoted with *w*. The search tree for beam search (Figure E.4) is reduced in comparison to the original search tree (see Figure E.2), since only a maximum number of nodes (*w*) can branch into new nodes.

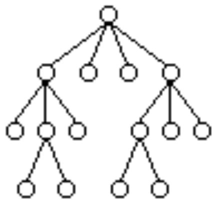


Figure E.4: The search tree for the beam search method.

E.3.4 Decomposition

The idea of decomposing a mathematical program into two or more sets of variables and associated constraints, is to separate some part with a special structure from the rest of the mathematical program. So the scheduling problem is divided in smaller (easier to solve) sub-problems which operate independently. The master problem is a relaxation of the original problem in that it contains only a subset of constraints and is solved to generate a trial solution for some decision variables. Its optimal value is a lower bound on the optimal value of the original problem. Once several variables are fixed, the sub-problem will solve a set of independent sub-problems. These dual multipliers are used to form one or more constraints (known as cuts) which are added to the master problem for the next iteration. The process continues until a feasible solution is found whose cost is sufficiently close to the lower bound. This process is shown in Figure E.5 (Marwali, 1999), in which the problem is decomposed into a master problem and a series of subproblems.

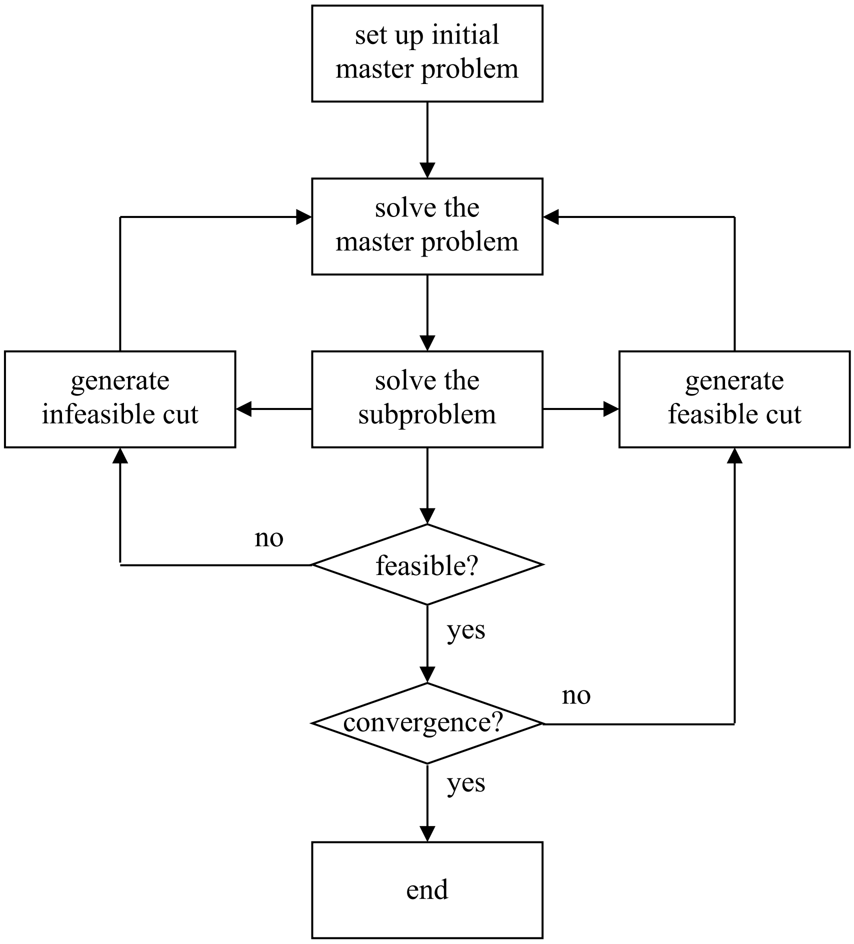


Figure E.5: The decomposition method in a flow diagram.

Earlier in the 1960-1970, many of the decomposition techniques were motivated by the inability to solve large-scale problems.

Lagrangian decomposition (Lagrangian relaxation)

Suppose we can partition constraints into "easy" and "hard" constraints. The concept behind a Lagrangian decomposition is that the difficult constraints are replaced with a linear penalty term in the objective function, by use of Lagrange multipliers, representing the violation of constraints. The resulting problem is easily solvable. Take for instance the following problem.

$$\begin{array}{lll} \text{minimize} & cx & \\ \text{subjected to} & Ax \le b & \text{(hard constraints)} \\ & Dx \le e & \text{(easy constraints)} \\ & x \ge 0 & \end{array}$$

To create the Lagrangian problem, we define the non-negative Lagrange multipliers λ. The Lagrangian relaxation function is defined by relaxing the hard constraints into the objective function. Each constraint that is relaxed into the objective function will get a separate Lagrange multiplier as a penalty when the constraint exceeds its restricted value. The Lagrangian function is then defined as follows:

$$\Gamma(x,\lambda) = cx - \lambda(b - Ax)$$

The mathematical optimization problem can be written in terms of the Lagrangian function as follows:

$$\begin{aligned} \text{minimize} \quad & cx - \lambda(b - Ax) \\ \text{subjected to} \quad & Dx \le e \\ & x \ge 0 \end{aligned}$$

The Lagrangian problem provides a lower bound on the optimal value of the original minimization problem (and an upper bound for a maximization problem). Now the best lower bound has to be found, i.e. the objective function of the Lagrangian relaxation has to be maximized over the set of all λ. There are different techniques such as Kuhn-Tucker or the dual optimization technique to solve the Lagrangian relaxation optimization problem. Another method is the subgradient algorithm. This algorithm is very appealing since it is easy to implement and it handles complex side constraints. One drawback of this approach is that there is no guarantee to find feasible solutions.

Benders' decomposition.

This technique decomposes the original problem into a master problem, which is a relaxation of the original problem, and several independent sub-problems. The process of solving the master problem begins with only a few (or no) constraints. Resulting into some fixed values for some of the variables from the original problem. Next the sub-problem is used as a test to see if this solution satisfies the remaining constraints. If so, the solution is optimal. Otherwise a constraint, which is most unsatisfied, will be added to the restricted master problem and the master problem will be resolved with more constraints. This extra restriction is called the Benders cut. Determining what the Benders cut will be, is done with information from the optimal dual vector of the sub-problem.

Dantzig-Wolfe decomposition

Dantzig-Wolfe decomposition is an efficient procedure when applied to linear problems whose coefficient matrices have an angular structure, i.e., independent blocks linked by coupling equations.

$$\begin{aligned} \text{minimize} \quad & c_1x_1 + c_2x_2 + \ldots + c_px_p \\ \text{subjected to} \quad & A_1x_1 + A_2x_2 + \ldots + A_px_p = b_0 \\ & B_1x_1 = b_1 \\ & \quad B_2x_2 = b_2 \\ & \qquad \ldots \\ & \qquad\quad B_px_p = b_p \\ & x_1 \ge 0, x_2 \ge 0, \ \ldots, x_p \ge 0 \end{aligned}$$

The first row identifies the coupling constraints and the remaining rows represent the independent blocks. The principle applied here is that the total problem consists of:

- master program that ties together the sub-programs and
- sub-programs corresponding to its almost independent parts.

Dantzig-Wolfe decomposition is used in combination with column generation, since the independent sub-problems can be captured into a set of columns. The sub-problems receive a set of parameter (simplex multipliers or prices) from the master program and then send their solutions to the master program, which combines these with previous solutions in an optimal way and computes new prices and send them back to the sub-problems. The iteration proceeds until an optimality test is passed.

E.3.5 Dynamic programming

The fundamental idea of dynamic programming was introduced in the 1950's by Bellman. Dynamic programming interprets any optimization as a multistage decision process, in which decisions have to be made over time. In order to calculate the optimal criterion value for any problem of size k, we first have to know the optimal value for each sub-problem of size k-1. So the results of dynamic programming can become known only after the total search for the best solution to the sub-problem is completed.

This method could look like:

$$F(t, s) = \underset{x \in X(t, s)}{\text{minimize}} \{r(t, s, x) + aF(t', s') \mid s' = T(t, s, x)\},$$

where

- F(t, s) = optimal total return on reaching time t in state s
- x = decision variable(s);
- s, s' = state variables;
- t, t' = time;
- X(t, s) = decision space (usually depends only on state);
- r(t, s, x) = immediate return;
- T(t, s, x) = state transform.

In words, the optimal return on reaching time *t* in state *s* equals the optimal return from an immediate decision plus the optimal return for the remainder of the process on the transition to state *s'*. This new state is a function of the current time, state, and decision. For discrete time, $t' = t+1$ for the forward equation and $t' = t-1$ for the backward equation. The multiplier *a* is generally a positive scalar, such as a discount factor to yield the present value of money.

E.4 Constraint Programming

The idea of constraint programming (CP) is to solve a problem by stating constraints about the problem area and finding a solution satisfying all the constraints. A constraint satisfaction problem (CSP) is defined as:

- a set of variables $X = \{x_1, \ldots, x_n\}$,
- a finite set D_i of possible values for each variable x_i (its domain) and
- a set of constraints restricting the values that the variables can simultaneously take.

A solution to a CSP is the assignment of values to each variable from its domain, in such a way that all constraints are satisfied at once. This is done by removing inconsistent values from the domain for each variable. Where an inconsistent value means a value for which it can be proven that it cannot be a part of the overall solution. So the domains are reduced. In many real-life applications, we do not want to find any solution but a good solution. The quality of a solution is usually measured by an application dependent function (the objective function). Such problems are referred to as Constraint Satisfaction Optimization Problems.

E.5 Heuristics

Priority or dispatching rules require little computational time and are easy to implement. Another attractive characteristic of a heuristics is that it can incorporate problem specific knowledge. This has the side effect as well, because the heuristic is not generally applicable anymore. A few examples of basic priority rules are:

- longest processing time first: this rule orders the jobs in decreasing order of their processing times
- largest number of successors: this rule may be used when the jobs are subjected to precedence constraints. It selects the next job by looking at the largest number of successive jobs

In this section most heuristics and search algorithms are presented. In general heuristics generate schedules of poor quality, so the trade-off between costs (time) and quality has to be made.

The first heuristics to be discussed are those which make use of bottlenecks (E.5.1). Next more recently developments like simulated annealing methods (E.5.3), taboo search (E.5.4) and genetic (or evolutionary) algorithms (E.5.5) are mentioned. Before these methods are discussed in detail, the more general model (local search algorithms or neighborhood search methods) will be discussed first in section E.5.2.

E.5.1 Bottleneck approaches

A few heuristics are based on the bottleneck approach. A bottleneck is a stage in a process that causes the entire process to slow down or stop.

OPT

In the late 1970s Eliyahu Goldratt developed OPT. OPT identifies the bottleneck resource and then schedules the activities around this bottleneck.

OPT procedure
Step 1. determine the bottleneck resource
Step 2. schedule the bottleneck to use this resource most effectively
Step 3. schedule the remainder up to the bottleneck
Step 4. schedule the remainder after the bottleneck

Shifting bottleneck

The shifting bottleneck method was first developed by Adams et al. (1988) and has recently been extended by Balas and Vazacopoulos. During each iteration of this heuristic, the new bottleneck is determined, scheduled and next the already scheduled activities are rescheduled to maintain feasibility.

Shifting bottleneck procedure
Step 1. initialization: $M_0 = \varnothing$ (scheduled machines)
Step 2. determine the bottleneck resource $M_k \in M - M_0$
Step 3. schedule the bottleneck, $M_0 = M_0 \cup M_k$
Step 4. resequence already scheduled machines ($Mi \in M_0 - \{k\}$)
Step 5. if $M = M_0$ stop, else go to Step. 2

Bottleneck dynamics

Basically, bottleneck dynamics estimates prices (delay costs) for delaying each possible activity and, similarly, estimates prices (delay costs) for delaying each resource. These calculations determine which activity to choose next (the highest benefit/cost ratio for being scheduled next) or which resource should be used next to perform an activity.

E.5.2 Local search algorithms

Local search algorithms usually start with a feasible solution and try to improve this solution by making small changes to it, like for instance reversing the sequence of two consecutive jobs or giving a single job a new position in the schedule. A solution that is obtained after making such a change to a solution is called a neighbor. The set of all neighbors is called the neighborhood. The simplest local search algorithm is Hill-climbing, in which the neighborhood of an initial (feasible) solution is searched to find a better solution. If a better solution is found, then the neighborhood of this solution is searched for again. This process continues iteratively until no improvements can be made anymore. Such a solution is a local optimum.

Neighborhood search procedure
Step 1. Obtain a seed solution
Step 2. If stopping criterion is met, stop; else continue
Step 3. Choose a neighborhood (adjacent pairwise interchange, general pairwise interchange)
Step 4. Choose a strategy to order the evaluation of solutions
Step 5. After evaluating some part of the solutions in the neighborhood, choose one or more new seeds.
Step 6. Return to step 1.

In order to find the global optimum, more solutions have to be analyzed. Several alternatives exist to do this. One option is to run the same algorithm more than once, each time with a different starting solution. But more sophisticated solutions exist and will be discussed next.

E.5.3 Simulated annealing

Scheduling heuristics employing simulated annealing are described by Eglese (1990). In simulated annealing a new solution is accepted based on a certain probability. If the neighbor ($f_{neighbor}$) is a better solution than the current one ($f_{current}$), then this neighbor becomes the current solution. Otherwise, it becomes the current solution with a certain probability. This probability depends on the difference between the value of the current solution and the value of the neighbor ($f_{neighbor} - f_{current} = \Delta$) and on a control parameter T, called the temperature.

$$p_{\text{current solution}} = \exp^{[-\Delta/T]}$$

The value of the control parameter T decreases during the execution of the algorithm, so the probabilities will decline and become unlikely that the algorithm will move to a neighbor with a worse solution value.

E.5.4 Taboo search

Scheduling heuristics employing taboo search have been reported by Glover and Greenberg (1989). In this search method a taboo list is keeping track of the most recent visited solutions. Taboo comes from the fact that these solutions may not be repeated while they are in the active list, when the best solution in the neighborhood is sought. The idea behind this is that solutions do not jump back and forth. If the number of entries in the taboo list is too small cycling may occur, if it is too large the search may be over constrained.

The primary difference between simulated annealing and taboo search is that simulated annealing diversifies by randomizing, while taboo search diversifies by forcing new solutions.

E.5.5 Evolutionary algorithms

Since the 1950's, some authors have been using concepts based on Darwin's evolution theory for the solution of optimization problems. Numerous algorithms based on the same concepts have been developed over the last 30 years. They are usually described by the term *evolutionary computation methods.*

Evolutionary algorithms (EAs) refer to any search process simulating the natural evolutionary process. This stochastic search technique explores different regions of the search space simultaneously, since they view several solutions as individuals of a population. Each individual is characterized by its fitness. The fitness of an individual is measured by the associated value of the objective function. The procedure works iteratively and each iteration is referred to as a generation. The population of one generation consists of individuals surviving from the previous generation plus new schedules (children). The children are generated through reproduction and mutation of individuals who were part of the previous generation (the parents). In each generation, the fittest individuals reproduce while the least fit die.

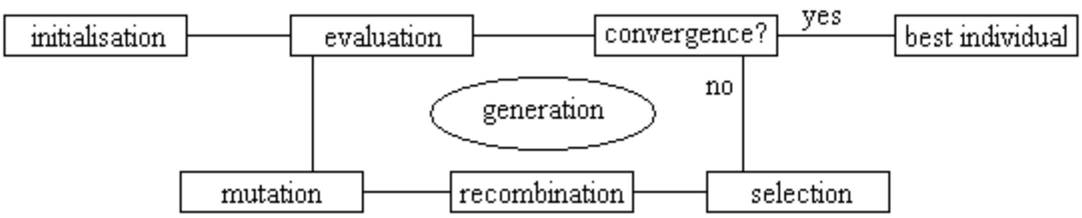


Figure E.6: The evolutionary process.

There are several approaches to map the problem into a representation, which is suitable for the genetic search; indirect and direct coding. The direct coding represents a complete schedule, while the indirect coding represents rules to some scheme builder. During the interpretation of an indirect representation some domain specific knowledge can be introduced. The actual scheduling is done mostly with local search techniques.

The strength of genetic algorithms is the exploration of different regions of the search space in relatively short times, since they explore a set of solutions simultaneously. Another advantage is that they can be applied to a problem without having to know much about the structural properties of the problem. Furthermore, they are very flexible and robust (e.g. multiple and complex objectives can easily be included). On the other hand, convergence to the local optimum usually is rather slow and cannot be predicted, in comparison with the more rigorous problem specific approaches.

E.6 Simulation

Simulation can represent quite realistic systems and high-level simulation languages make the programming relatively modest. Simulation also has the advantage of providing a more natural approach for interfacing with human expertise. However, finding a good solution is not likely with this technique. That is why simulation can be used to test candidate schedules that are proposed by the user. This technique can be combined with other techniques as well.

E.7 Applicability to shutdown scheduling

The general mathematical programming techniques that are mentioned in this appendix are not all always applicable. Sometimes the formulation has to be adapted in such a way that the model can be solved by a particular technique. Take for instance dynamic programming. This technique can be found useful for shutdown scheduling (Milton, 1966), since it solves a sequential decision process problem and in scheduling shutdowns the decision which unit should be shut down and at what time can be determined sequentially. But there are so many variables to take into account that this technique is severely limited by the curse of dimensionality (i.e. a problem that is complex enough to be interesting is generally too large to be solved within practical constraints of computer time and memory).
This is the problem with most mathematical programming approaches (or exact methods for that matter), since all choices are made simultaneously and the optimal solution can be obtained. For small refineries this can be possible with some solver, but the problem becomes soon too large. This can (till some extend) be suppressed by a relaxation of the problem. Several relaxations are possible:

- discard some of the restrictions and/or some decision variables (like the storage variables which require binary decisions in the model of chapter 4),
- a Lagrangian relaxation by putting some restrictions into the objective function or
- a linear relaxation of the binary variables.

Another possibility is to divide the problem into smaller subparts. This can be done in a decomposing way as described in this chapter. The original problem can be divided into production scheduling and shutdown scheduling. Both scheduling problems could be analyzed, using different techniques. But the determination of the optimal maintenance and production policies clearly depends on each other. If a production plan is fixed and used as input to the optimization formulation for the determination of the optimal maintenance policy, it is likely that a different production plan may facilitate a better

maintenance policy. On the other hand, if a maintenance schedule is fixed and used as input for the determination of the optimal production plan, it is likely that a different maintenance policy may facilitate a better production schedule. To overcome these concerns and quantify the interactions between production and maintenance planning models, proper linking mechanisms between the two models must be established in such a way that a simultaneous strategy is developed.
This interaction can be realized by solving the shutdown scheduling part (under the objective of minimizing the completion time for instance) and use the outcome as input for the production scheduling part. Once the maintenance variables are fixed, the operational problem will be treated as an independent problem. With the use of dual variables (see Appendix D) from the production scheduling part, it could be found out what the advantage would be to have some unit in production if it is shut down during a particular time period and vice versa. An additional constraint can be introduced that specifies that a particular unit should be shut down (or operating) at some predefined time period. Figure E.7 shows the interaction between shutdown and production scheduling, in which a master problem is solved at the first stage to determine a solution for the maintenance schedule decision variables. In the second stage, sub-problems are solved to minimize operating costs while satisfying network constraints. The dual variables give an extra constraint.

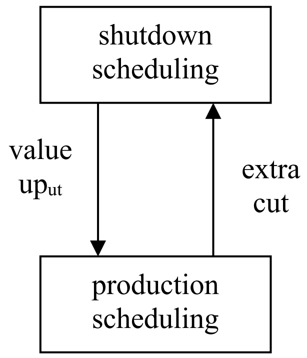


Figure E.7: The interaction between shutdown and production in scheduling.

These interactions will greatly increase the complexity of the problem, resulting in a large optimization problem.

Appendix F Solving simple network problems

This appendix gives a brief introduction to solution techniques for project scheduling problems without any resource constraints. The different methods that will be discussed are:

- CPM: Critial Path Method (Wiest and Levy, 1969; Tijms and Kalvelagen, 1994)
- PERT: Program Evaluation and Review Technique (Miller, 1963)
- PDM: Precedence Diagramming Method (Moder et al., 1983)
- MPM: Metra Potential Method (Kerbosh and Schell, 1975; Zhan, 1994)

F.1 Critical Path Method (CPM)

The critical path method (CPM) was developed by Du Pont in 1957. This network analysis technique is used to predict the project duration by analyzing which sequence of activities (which path) has the least amount of scheduling flexibility (the least amount of float). Besides the determination of the length of the project, this method also identifies the activities that are critical to the completion of the project. The chain of activities formed by the critical activities is called the critical path. Any delay in a critical activity will immediately cause changes to the start or finish dates of the project.
The activity duration can be treated as variable, depending on the amount of labor or capital for instance. Then there is a time/cost trade-off for each activity in the project. For certain activities it may be possible to decrease their completion times by spending more money. It becomes the question how this affects the overall completion time of the project. The CPM could emphasize this trade-off between the expenses of the project and its overall completion time. But mostly the CPM is used for identifying the critical path.

F.1.1 Calculations

Consider a project consisting of N activities, numbered such that activity *i* has only lower numbered predecessors. This implies that node 1 is the start node and node N the finish node. Let d_i indicate the duration of activity *i*. The next step is to determine the earliest and latest starting time for each activity. Early dates are calculated by means of a forward pass using a specified start date. For each node we calculate the earliest finish time (EF_i) with the formula

$$EF_i = d_i + \max\{EF_j \mid j < i, \text{ i has predecessor j}\}$$

Then the earliest start time becomes $ES_i = EF_i - d_i$. It is also interesting to determine the latest times at which activities should start (LS_i), while keeping to the earliest finish time. Late dates are calculated by means of a backward pass starting from a specified completion date (usually the forward pass's calculated project early finish date).

$$LS_i = \min\{LS_j \mid j > i, \text{ i has predecessor j}\} - d_i$$

Three separate categories of float are defined in critical path scheduling:

1. *Free float* is the amount of delay, which can be assigned to any one activity without delaying subsequent activities. The free float (FF_i) associated with activity *i* is:
   $$FF_i = \min(ES_j - EF_i)$$
2. *Independent float* is the amount of delay, which can be assigned to any one activity without delaying subsequent activities or restricting the scheduling of preceding activities. Independent float (IF_i) for activity *i* is calculated as:
   $$IF_i = \min ES_j - \max LF_h - d_I$$
   where activity *h* is a predecessor of activity *i* and activity *j* a successor.
3. *Total float* (also called slack) is the maximum amount of delay, which can be assigned to any activity without delaying the entire project. The total float (TF_i) for any activity *i* is calculated as:
   $$TF_i = LF_i - EF_i = LS_i - ES_i$$

Activities with $TF_i = 0$ are called critical. A set of k activities is called a critical path if all activities are critical and if the activity has to start immediately whenever the predecessor finished. For all critical activities the start times are known. For the non-critical they still have to be determined, the possibilities for activity *i* are in the interval $[ES_i, LS_i]$. This completes the description of the critical path method.

F.2 Program Evaluation and Review Technique (PERT)

In 1958 Project evaluation and review technique was developed by the US Navy for the planning and control of the Polaris missile program and the emphasis was on completing the program in the shortest possible time. This technique adds the complication that activity durations are not really exactly known but are assumed to be probabilistic (e.g. for a particular activity the most likely completion time is 4 weeks but it could be anywhere between 3 weeks and 8 weeks).

If the only delay will come from the critical path, then it becomes interesting to estimate the probability distribution of the project duration. This distribution is often assumed to be normal. This method of estimating the distribution of project durations, applies the critical path method to a weighted average duration estimate. The model requires three basic assumptions:

a) the critical path is considerably longer than any near-critical path,
b) the activities are probabilistically independent and
c) the critical path can be analyzed approximately by the central limit theorem.

The first assumption is required to keep it simple, because the critical path only could lengthen the project's completion time then. When the duration of each of the critical activities is seen as a stochastic variable, they can get accumulated and form a new stochastic variable. This is possible due to the other two assumptions. What this means will be discussed in the next section.

F.2.1 Calculations

Sometimes it is found more realistic to estimate shutdown duration as a range rather than as a precise amount. Another way to deal with the lack of precision in estimating time is to use a commonly accepted formula for a particular shutdown. The last option is the usage of a probability by calculating a standard deviation of the time estimate. Take for instance the following three time estimates

t_m = the most probable time
t_o = the optimistic (shortest) time within which only 1% of similar projects are completed
t_p = the pessimistic (longest) time within which 99% of similar projects are completed
t_e = the estimated time

These estimates can be really estimations or some historical averages. The most relevant pros and cons are summarized in Table F.1.

	positive	negative
historical data	- a fair degree of consistency - training is relatively simple	- historical values reflect what has happened rather than what should happen - past inadequacies become part of the system
estimating	- estimates are easy to develop - all shutdowns can be estimated	- inconsistency - varying accuracy between different estimators - training is difficult

Table F.1: Historical data offers several advantages disadvantages over estimating.

The estimates are used to determine the average completion time. The expected duration of an activity (t_e) is calculated using the following formula:

$$t_e = \frac{t_o + 4t_m + t_p}{6}$$

It is not necessary to go into the theory behind the formula. It is enough to know that the weights are based on an approximation of the Beta distribution. The average times are used to figure the critical path and the standard deviation of completion times for the entire project. The standard deviation for the duration is calculated by the rather simplified formula:

$$\sigma = \frac{t_p - t_o}{6}$$

The variance is the square of the standard deviation. The standard deviation is a good measure of the variability of each activity. Calculating with these numbers, it can be said that

- 68.26% of the time the shutdown will be completed within the range of $t_e \pm 1$ standard deviation,
- 95.44% of the time the shutdown will be completed within the range of $t_e \pm 2$ standard deviations and
- 99.73% of the time the shutdown will be completed within the range of $t_e \pm 3$ standard deviations.

Whenever the average times are calculated and used for the critical path method, the activities on the critical path q are known. Let p_i be the duration of activity i. This is a stochastic variable with average μ_i and standard deviation σ_i. Then the length of the critical path (L_q) has approximately a normal distribution with the following parameters:

$$\mu_q = \sum_{i \in q} \mu_i$$

$$\sigma_q^2 = \sum_{i \in q} \sigma_i^2$$

The probability of meeting a project deadline of d becomes:

$$P(L_q \leq d) = P((L_q - \mu_q)/\sigma_q \leq (d - \mu_q)/\sigma_q) = \Phi((d - \mu_q)/\sigma_q)$$

PERT has the ability to cope with uncertain activity completion times. This employs managers to estimate a probability distribution for a particular activity and uses this to obtain a simple estimate of the distribution of the overall project duration.

F.3 Precedence Diagramming Method (PDM)

The precedence diagramming method (PDM) was introduced in 1964 by H.B. Zachry in cooperation with IBM as an extension of the basic PERT/CPM network analysis. The precedence diagramming method (PDM) is a network diagramming technique used to show the relationship between activities by linking sequences with precedence relationships to show how activities dependent on each other. PDM permits mutually dependent activities to be performed partially in parallel instead of serially. It is very similar in appearance and use to the AON, but with the added feature that relationships need not only be finish-to-start in nature. The usual finish-to-start dependencies between activities are relaxed to allow activities to be overlapped.

F.4 Metra Potential Method (MPM)

Positive and negative time lags are general timing restrictions between the starting times of jobs, which have been introduced in connection with the metra potential method (Roy, 1970). Although very powerful, these relations have been considered only seldom in the literature.