Quantitative models for stochastic project planning

Quantitative Models for Stochastic Project Planning

This thesis is part of the PhD thesis series of the Beta Research School for Operations Management and Logistics (onderzoeksschool-beta.nl) in which the following universities cooperate: Eindhoven University of Technology, Maastricht University, University of Twente, VU Amsterdam, Wageningen University and Research, and KU Leuven.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-4813-2

Quantitative Models for Stochastic Project Planning

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. F.P.T. Baaijens, voor een
commissie aangewezen door het College voor
Promoties, in het openbaar te verdedigen
op 4 juli 2019 om 16:00 uur

door

Sjors Wilhelmus Franciscus Jansen

geboren te Angeren

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof. dr. T. Van Woensel
1e promotor:	prof. dr. A.G. de Kok
2e promotor:	prof. dr. ir. I.J.B.F. Adan
co–promotor:	dr. Z. Atan
leden:	prof. dr. E. Demeulenmeester (KU Leuven)
	prof. dr. D. Trietsch (American University of Armenia)
	prof. dr. F. Erhun (University of Cambridge)
	dr. ir. N.P. Dellaert

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Contents

1

Introduction

What is a project? Ask this question to 10 different people and you will probably get 10 different answers. A commonly used definitions is 'A set of activities aimed to achieve a specific objective and have a clear start, middle and end' (Goldratt and Cox, 2016). This is of course a very broad definition. For example, writing this thesis is a project but so is building a house or organizing a conference.

Projects consist of activities. Activities can be seen as parts of a project, that each have a specific objective and have a clear start and end. When all activities are completed, the project is completed. Typically, activities require resources, such as workforce, materials and equipment. The more resources available, the better and faster an activity is completed. However, resources are scarce, which makes planning activities difficult. Activities can also have precedence relations. Some activities can be executed in parallel, others can only be executed in a particular order.

Projects are typically a 'one off' task. For example, writing a dissertation is something a student does only once. One could argue that over the years many theses are written. However, these projects are executed by different PhD candidates on different topics. Therefore it is fair to say that each project is unique. The uniqueness of a project makes it difficult to predict the realization of a project in terms of duration, cost and quality. For the organization executing the project, it is

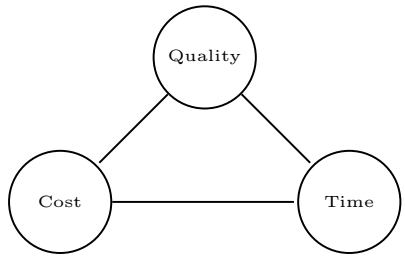


Figure 1.1: The iron triangle in project management

key that all projects it is carrying out are properly managed. Although all projects are unique, there can be similarities. For example when organizing an annual conference, the organizers gain experience and can better predict the outcome of a project. At this high level of abstraction, we can use a probabilistic view to improve on current project management practices.

In this thesis we study production planning problems in a high-tech environment. At first, approaching a production environment from a project planning perspective sounds somehow strange. Production planning or production control is a field of research existing for already many years. In the field of production planning, one generally thinks of concepts like customer demand, inventory and throughput times. While this is still a very valid approach for mass-produced products, this approach is not suitable for complicated products, for example aircrafts. With a sales price of 445.6 million USD and only 12 deliveries in 2018[1] the Airbus A380 aircraft is an example where producing a product could be seen as a project. Of course a company can work on multiple projects simultaneously. In that case the available resources are shared amongst projects and can put additional constraints on the individual project. Aircrafts are not the only example. This type of products, also known as capital goods, are very complicated, customer specific and capital intensive products. Manufacturing them is a long process, with many uncertainties.

The success of a project is determined by three factors: cost, time and quality. Together, these factors form the so-called iron triangle (Atkinson, 1999) as shown in Figure 1.1. Using these factors, one can define the success of a project. A project is successful when it is completed on-time, within budget and meets the quality requirements. The project manager decides on trade-offs between the three factors

[1] https://www.airbus.com/newsroom/press-releases/en/2018/01/airbus-2018-price-list-press-release.html

Factor	Chapter 2,3,4	Chapter 5
activity leadtime	uncertain	deterministic
customer demand	deterministic	uncertain
number of activities per project	deterministic	uncertain
optimization problem	unconstrained, optimize project	capacity constrained, optimize portfolio.
main application area	high-tech manufacturing systems	new drug development

Table 1.1: Properties of the two model types

(Meredith and Mantel Jr, 2011). A project's cost can be reduced, but this decreases quality and/or increases the duration.

In this thesis we focus on the cost and time aspects of project management and the trade-off between them. We do not explicitly consider the quality aspects of a project. The reason for this is that in production environments, the required quality of the final product is set by the customer. It is a requirement that should be met, instead of a variable that can be changed by the manufacturer. Furthermore we aim to develop general quantitative models that can be used in different application areas. It is difficult to come up with a general quantitative measure for quality that suits multiple application areas.

Predicting the result of a project in terms of time and cost is difficult due to uncertainties that arise during the project. Uncertainties can cause delays and can lead to high costs. Therefore we develop stochastic models that capture these uncertainties. In this thesis, we develop two types of models, which are motivated from two different application areas. In Table 1.1 we show the differences between the two model types.

In Chapters 2,3 and 4, the main uncertainty we consider is the duration of project activities, which together determine the total leadtime the production of a product. In each chapter we develop production plans that take into account the leadtime uncertainty. What differs between the chapters is the complexity of the production network and the way costs are incurred. We mainly derive structural results which are illustrated by numerical examples.

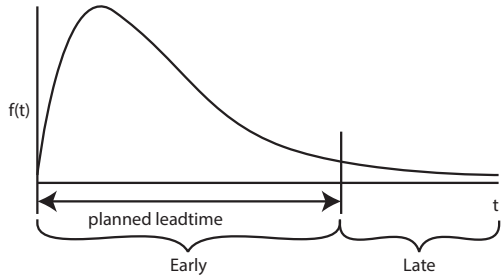


Figure 1.2: Distribution of stochastic leadtime (random variable) and planned leadtime (decision variable)

In Chapter 5 we develop a completely different model, which does not build on the models in previous chapters. Here, we focus on uncertainty in customer demand and the number of activities in the project. The main application area here is new drug development, where the outcome of a project is uncertain. This model is developed specifically for this application area. In the next sections we will further introduce both types of models.

1.1. Project Planning Under Leadtime Uncertainty

In the first three chapters of this thesis, we focus on project planning. This problem is motivated from the production of lithography machines. After the customer order has been confirmed, the production plan for this specific machine can be developed. A set of activities and precedence relations is developed and precedence relations are defined. Each activity has a planned duration, which we in a manufacturing environment often refer to as the leadtime.

However, manufacturing a new product is subject to all kinds of uncertainties (Mula et al., 2006). For example, resources might be unavailable, errors in production require repairs or ordered materials are delayed. Due to these causes the duration of an activity, or its leadtime is uncertain. For this reason, models with deterministic leadtimes are not suitable and stochastic models should be used (Herroelen and Leus, 2005). Therefore, in this part of the thesis we model the leadtime of an activity by a random variable. Due to this uncertainty it is difficult to allocate time to an activity in advance. This problem is shown in Figure 1.2 in this thesis.

This figure shows the probability density function of a stochastic leadtime and a so called planned leadtime, the time reserved for that activity. Depending on the realization an activity is completed either too early or too late. When increasing the planned leadtime, the probability that the activity is late reduces. On the other hand, increasing the leadtime also increases the probability that a system finishes too early. Both an early completion and a late completion are undesired, since deviations from the plan are costly.

Different streams of literature use different definitions for leadtime. For example, in supply chain management, leadtime refers to a norm, while the actual duration of an activity is referred to as flowtime or throughput time (Cox and Blackstone, 2002). To emphasize the difference between the planned leadtime (the norm) and the stochastic leadtime (the actual duration), we will only use planned leadtime and stochastic leadtime as described in Figure 1.2.

The figure shows the leadtime of one activity, but in a project with multiple activities it is even more important that deadlines are met. Otherwise delays propagate through the project. On the other hand, when every activity has a long planned leadtime, the total leadtime of the project becomes too long. Therefore it is key to develop a production plan that allocates the appropriate amount of time to the right activities such that a high service level can be achieved at minimal costs. This plan should describe exactly when an activity should start and when it should be completed.

When developing a project plan all kinds of factors should be taken into account. What is a good plan? What is a good plan for one stake-holder might be an infeasible plan for another. To develop a suitable plan, we consider the cost for deviating from the plan. As leadtimes are uncertain, there will always be deviations from the plan. Our objective is to minimize the expected cost for deviating from the plan, i.e. we average over all possible realizations of the leadtime random variables. This is different from other planning methods, such as the Project Evaluation and Review Technique (PERT) (Malcolm et al., 1959), which mainly focus on minimizing the total duration of a project. The cost for deviating from the plan are different throughout the network. A deviation at the start of a project is for example not very costly, while a deviation from the project due date is extremely expensive. By minimizing the cost, we find a plan that allocates the right amount of planned time to the right nodes.

1.1.1 Cost Structures

In this thesis, we consider project networks with two different cost structures. Both cost accounting schemes incur a linear holding and a linear penalty cost for earliness and tardiness for deadlines. This can be intermediate deadlines between activities or the final project deadline. This type of cost function was first introduced by Yano (1987a) and after that it became a commonly used cost function in the literature. For a network with only one activity, this problem is mathematically identical to the commonly known Newsvendor problem, first formulated by Edgeworth (1888). In a Newsvendor problem, one sets a target level for a stochastic variable. The optimal target level minimizes the expected cost for the stochastic variable realizing above or below this target level. The term Newsvendor stems from an application where a publisher needs to decide how many newspapers it prints to satisfy demand for that day (Arrow et al., 1951). In planning problems, we set a target (planned leadtime) for the stochastic leadtime. At optimality, the probability that the actual duration of this node exceeds the planned duration is a Newsvendor fractile $\frac{h}{h+p}$ where h is the cost for completing earlier than planned and p the cost for completing later than planned.

When analyzing networks with multiple activities, the problem becomes more difficult. However, in general it is still a Newsvendor type of problem, i.e. balancing costs of being early with costs of being late. Instead of 1 type of holding costs, we now assume that each activity in the network adds value to the final product. The more activities are completed, the higher the value of the project and the higher the holding cost per time unit. Under these increasing holding costs, waiting time early in the project is cheap, while waiting time right before the due date is the most expensive. From a cost perspective one could argue that safety time therefore should be allocated early in the project. On the other hand, safety time early in the project is not very effective. It can only be used to compensate for delays that occurred in the few activities that are scheduled earlier. Putting the same amount of safety time at the end of the project is more effective, as it can compensate for delays in all activities of the project. Following this argument, one should put all safety time at the end of the project. This shows the difficulty of allocating safety time: it is a trade-off between cost and effectiveness.

To explain the difference between the two cost functions, we consider an example of

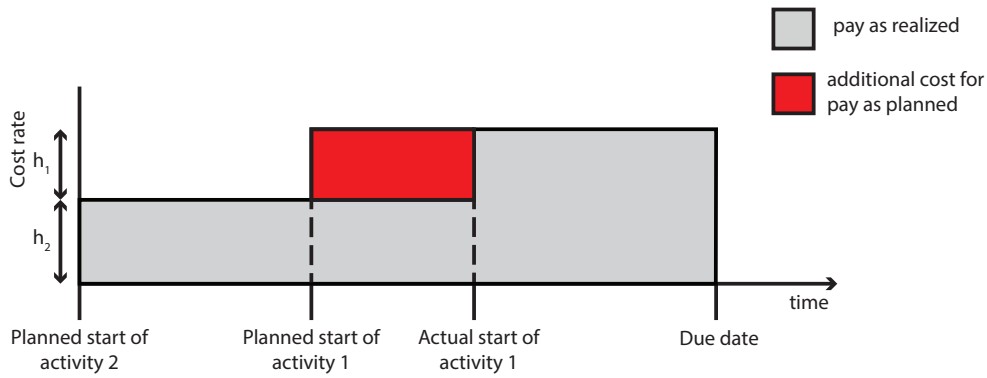


Figure 1.3: Difference between the two cost functions

a project consisting of two activities. Activity two is the first activity to start, activity 1 starts when 2 is completed. We assume the precedence relation to be fixed. In Figure 1.3 the cost rate per time unit is shown. At the moment activity two starts, we start incurring holding cost h_2 per time unit. We incur this cost until the product is delivered to the customer. After some time, we pass a time point where activity 1 is planned to start. However, activity 1 can only start at this time when activity 2 is already completed. In the figure, activity 2 is late and hence the start of activity 1 is delayed. Now there are two options to start incurring holding cost for node 1. Either incur the holding cost for node 1 from the actual start time of node 1 or from the planned start time of node 1. Both holding cost are incurred until the due date of the product. In this thesis, we define the 'pay as realized' costing scheme as the scheme that starts incurring cost from the actual start time and the 'pay as planned' costing scheme as the scheme that starts incurring holding cost for a node from its planned start time.

In the figure, the total amount of holding cost incurred for the product is the integral of the cost rate from the start of the project, until the final activity is completed. For the 'pay as realized' costing scheme this is the grey area in Figure 1.3. The 'pay as planned' costing scheme includes both the grey and the red area. At first sight, this is only a minor difference between the two cost functions. However, this small difference has significant implications for properties of the planning solution and algorithms for finding the optimal solution.

Besides a constant holding cost rate and the corresponding linear increasing total

	Assembly Systems	General Networks
Pay as planned	Chapter 2	Chapter 3
Pay as realized		Chapter 4

Table 1.2: Overview of Chapters 2, 3 and 4

holding cost, one could also look at other types of cost functions. In Ferguson et al. (2007) and San-José et al. (2015) nonlinear cost functions are considered for an inventory model. For a one activity project these results could be used. However, generalizing these results to a network of activities is difficult. Therefore, we focus on linear cost functions for project networks. The obtained results can then be used when analyzing nonlinear cost structures.

1.1.2 Research Objectives

We analyze both cost structures and apply them to a variety of network structures in Chapters 2, 3 and 4. What distinguishes the chapters are the cost structures considered and the network structures we analyse. In Table 1.2 an overview is given. Chapter 2 studies both cost functions simultaneously for an assembly system with N parallel subassembly activities and one final assembly activity. Chapters 3 and 4 both study more complicated networks, and each focus on a specific cost structure.

The assembly system in Chapter 2 consists of one final assembly activity, which can only start when N parallel sub-assembly activities are completed. For such a system the 'pay as realized' cost structure is already widely used in the literature on planned leadtimes. Although the 'pay as planned' cost function is known in the literature of project management (see Baker and Trietsch (2019)), it is rarely used in the field of production planned leadtimes. In this chapter, we introduce and analyze the 'pay as planned' cost function with the following objective:

Research Objective 1 *Analyze the 'pay as planned' cost structure and compare the structural results to results for the 'pay as realized' cost structure for an assembly system.*

Our main contribution in this chapter is the introduction of the 'pay as planned' cost structure. We show that for production environments where intermediate deadlines are relatively important compared to the final deadline, the 'pay as planned' cost

structure is suitable. For this cost function we derive optimality equations and properties of the optimal solution. We develop a so called 'blame policy' that we analytically compare the cost structure to the 'pay as realized cost structure'. In this comparison, we also unveil that the optimal solution of this cost structure can contain negative planned leadtimes, which do not make sense in practice. We prove that under the 'pay as planned' cost structure, planned leadtimes of the optimal solution are always positive. The chapter is a slightly modified version of Jansen et al. (2019).

Most production systems and also most projects have more complicated precedence relations than assembly system. Therefore, in Chapter 3 we extend the results for the 'pay as planned' cost structure to a general class of networks, namely directed acyclic networks with multiple end points. This allows us to model projects that have multiple converging points (activities that can only start when a set of predecessors is completed) and diverging points (a set of activities that all can start when a single activity is completed). We introduce a new problem formulation that instead of planned leadtimes (durations) uses planned start times (time points) to describe the project plan. Our objective is to extend the results from the assembly system to this type of networks.

Research Objective 2 *Develop a problem formulation suitable for general networks and extend structural results obtained from the assembly system for the 'pay as planned' cost structure.*

Our main contribution in this chapter is the derivation of optimality equations for this general network structure. Before the derivation, we extend the idea of the blame policy in Chapter 2 into the concept of tardy paths. In short, a tardy path can be seen as the stochastic version of a critical chain, a term coined by Goldratt (1997). After a project is completed i.e. when all stochastic durations have realized, we search for a path that determined the completion of the project. The tardy path cannot be determined beforehand, we can only, given a plan, determine the probability of a tardy path. For each node in the network, we can derive a Newsvendor Equation, that relates the probability of a tardy path starting in that node to the value that node adds to the final product. Furthermore, we show that the problem is convex and that simulation-based optimization is a valuable tool to determine the optimal solution for a specific problem. The chapter is a slighlty

modified version of Jansen et al. (2018).

The problem formulation developed in Chapter 3 is also applicable to the 'pay as realized' cost structure. In fact, using this formulation, we can extend existing analytical results for specific network structures, which is the main objective of Chapter 4.

Research Objective 3 *Derive optimality equations for a general network under the 'pay as realized' cost structure.*

While the 'pay as realized' cost structure is commonly used in the literature, the derivation of analytical results was limited to specific network structures, such as serial systems and two-stage assembly systems. Our main contribution in this chapter is that we derive optimality equations for any strictly converging network with 1 final node. To this end, we use the idea of tardy paths again, but slightly modify the mathematical definition of it. We show that also for this cost structure optimality equations have a Newsvendor shape. In the field of setting planned leadtimes, this is a long awaited result and it can be seen as a closure to the field of planned leadtimes. As determining the optimal solution is complicated, the result helps in validating solutions obtained via heuristic methods. Finally we discuss the possibilities and limitations of extending the results to networks with fork-join structures and networks with multiple end nodes.

1.2. Capacity Constrained Project Portfolio Planning

The first part of this thesis focused on optimal planning of individual projects. The chapters built on each other and are extensions and variations of the single-activity Newsvendor problem. The last chapter of this thesis addresses a completely different problem and does not built on the models developed in the previous chapters. More than the previous chapters, the research described in this chapter is ongoing work. Therefore the results are obtained for relatively small and specific problem instances. Future research is needed to generalize the results.

We consider an organization executing multiple projects simultaneously. For an organization executing multiple projects, optimizing each project individually most likely is not the best strategy. This is due to the fact that projects interact with

each other. Suppose we have an organization executing multiple projects. If these projects have similar activities it can be useful to execute these activities simultaneously. For example, when both projects need materials from a supplier, it can be wise to place a single order, saving transportation and ordering cost. On the other hand activities can require the same resources, which implies that activities of different projects need to be executed sequentially. Optimizing each project individually can then lead to infeasible plans for the resource. Besides avoiding infeasibilities, it is also key to plan the projects such that the resource is highly utilized. The problem of planning a set of activities from different projects for multiple resources is resource-constrained project scheduling. For an overview of this field we refer the reader to Brucker et al. (1999).

In Chapter 5 we develop a resource-constrained project scheduling model for a specific industry, namely the development of new drugs in the pharmaceutical industry. Projects executed require high investments, with an average of USD 2.6 Billion per drug (DiMasi et al., 2016). The development of a drug is a highly uncertain process. This is emphasized in Figure 1.4. This figure shows the timeline of the development of a drug, based on average data from the Food and Drug Administration (FDA) of the United States. For every drug receiving approval from the FDA, many other drug development projects are terminated early. There can be many reasons for early termination, such as limited or no efficacy, severe side effects, or the development costs exceeding the potential revenue of a drug.

The figure also shows that, if a drug reaches the phase of FDA approval, a long development period has passed. It is only after this long development period that the drugs starts generating revenue to return the investments.

As acknowledged by Colvin and Maravelias (2011), the high risk of early termination is what distinguishes drug development projects from most other projects, where it is beforehand clear which activities need to be executed to complete the project. This makes it difficult to schedule activities of projects that share resources. If a resource is scheduled for a specific activity, it might be idle when that project is cancelled. On the other hand, not reserving the resource can result in delays of the project in the case the project is continued. Both outcomes are costly and should be avoided.

[2]Modified from https://www.slideshare.net/rahul_pharma/drug-discovery-and-development-10698574

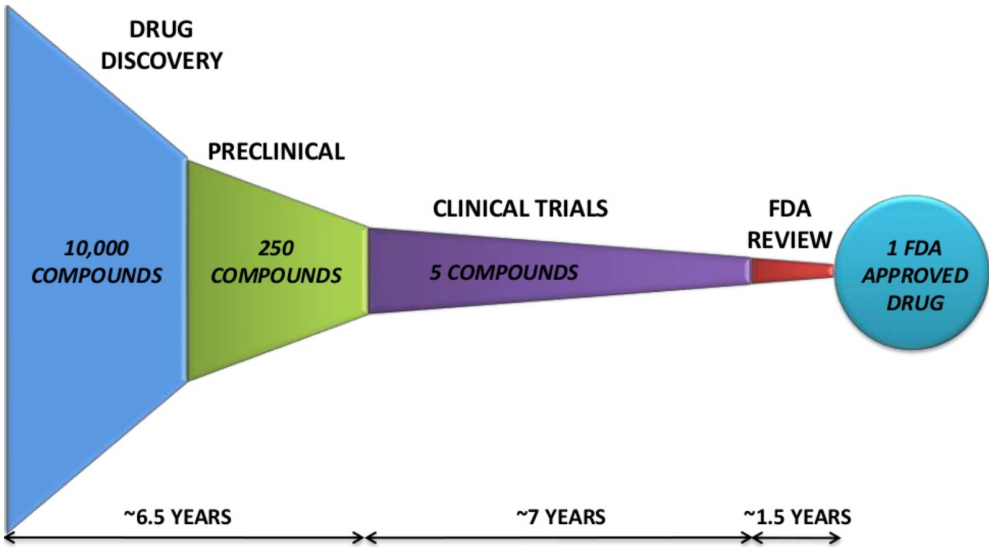


Figure 1.4: Drug development timeline [2]

To overcome this problem, pharmaceutical companies outsource parts of their drug development projects to Contract Research Organizations (CRO's). For a pharma company outsourcing reduces the risk of under-utilizing resources and it gives flexibility (Howells et al., 2008). A CRO executes parts of drug development projects for a fixed fee. Typically, it has a large amount of resources, such as laboratory and hospital facilities, but also highly skilled employees, such as nurses and doctors. A CRO executes multiple projects for multiple pharma companies. In this way it can reach economies of scale and improve the utilization of its resources.

To achieve this, it is key for the CRO to properly manage its capacity. According to Song et al. (2019) capacity management involves two central questions. On the one hand a firm needs to determine how much capacity it should install and on the other hand how to best utilize existing capacity. A CRO can affect its capacity utilization by selecting the right project proposals from pharma companies. When a pharma company offers a project to a CRO, the CRO has to decide whether it accepts or rejects the proposal. Besides having the right project, it is also key for a CRO to have the right amount of resources available. Choosing the amount of resources is difficult, since demand from pharma companies is uncertain. Therefore, it is our objective to develop models that maximize CRO's profitability, by choosing

the right amount of resources and selecting the right projects.

Research Objective 4 *Develop mathematical models that capture the portfolio selection problem from a CRO point of view.*

The portfolio selection problem is not new. One could argue that the problem only shifted from a pharma company to a CRO. However, the CRO has different incentives than a pharma company. Hence, the CRO solves the problem differently. A pharma company takes into account the market potential of a drug when selecting which drugs to develop. For a CRO this is irrelevant, since it gets paid for executing an activity, regardless of whether the pharma company decides to terminate or continue the project, after this activity is executed. What is relevant for a CRO are the characteristics of the offer compared to other project offers it receives.The difficulty here is that it is uncertain which offers it will receive in the future.

1.3. Outline of the Thesis

The remainder of this thesis is outlined as follows. In Chapter 2 we introduce the 'pay as planned' costing scheme for an assembly system with 1 final node and multiple parallel predecessors, all feeding this node. We compare the results to the 'pay as realized' cost function using existing results from the literature. This chapter contains an overview of the relevant literature in Section 2.2. This literature is also relevant for Chapter 3 and Chapter 4 since both chapters consider extensions of the same problem. Therefore, these chapters do not contain a separate literature section.

In Chapter 3 we extend the 'pay as planned' costing scheme to general networks with multiple end nodes and fork-join structures. In the final chapter on the planned leadtime problem, Chapter 4, we develop structural results for the 'pay as realized' cost structure for converging networks with one end node.

In Chapter 5 we switch to a different problem setting, where we analyze a capacity-constrained project selection problem. The related literature for this problem can be found in Section 5.2. Instead of focusing on high-tech production systems, we use the process of new drug development as our application area.

We would like to note that the problems discussed in Chapters 2, 3 and 3 are related. Therefore, we rely mostly on the same notation. However, the differences in the network structures and cost functions require different and additional notation. For the sake of completeness, we choose to define all the required notation in each individual chapter. This approach makes the mathematical formulation of each chapter self sufficient.

2

Leadtime Planning for Assembly Systems

2.1. Introduction

Companies that aim to operate with a minimum cost and a high service level face a challenging task of identifying the best operational strategies for achieving these objectives. Attaining these goals is difficult as companies face many uncertainties on the supply, production and demand sides. The magnitudes and effects of these uncertainties on the operations of a company depend on its production strategy. The best strategy can be determined by identifying how deeply the customer order penetrates the company's supply chain (Atan et al., 2017).

A production strategy known as 'configure-to-order' strategy is commonly utilized by companies that offer customized and, mostly, expensive products. The end product fulfills the exact customer needs. This is an appealing strategy for high-tech, car manufacturing and white goods industries.

The supply chain of a company using a configure-to-order strategy distinguishes between two main phases. The first phase concerns the orders placed with the suppliers. This phase is forecast-driven and is executed in advance to prevent excessive customer leadtimes. The second phase is driven by the customer order

and concerns the production of the final customer-specific product. The point that divides these two phases is known as the customer order decoupling point (de Kok and Fransoo, 2003). The focus of this chapter is on the second phase of the configure-to-order strategy.

We consider a company that keeps inventory of components only. Production of multiple subassemblies is triggered by a customer order. These subassemblies are then assembled into a final product. In many production environments, production and assembly leadtimes are random (Dolgui et al., 2013). To overcome uncertainties in leadtimes, companies typically use safety stocks (Whybark and Williams, 1976; Chopra et al., 2004). However, the subassemblies and final product are customer-specific and thus cannot be kept on stock. Therefore, safety stocks cannot be used as a technique to absorb the uncertainties in the production process. A plan, which sets planned production leadtimes for subassembly activities and final assembly activities, is used to buffer against uncertainties. We call these production leadtimes *planned leadtimes*. The planned leadtime is the sum of the average leadtime and a safety time. The difficulty of planning the second phase of the configure-to-order strategy arises from the interactions among multiple processes. A good plan should coordinate the production of subassemblies and ensure on-time delivery of the final product.

The company incurs a cost for holding each subassembly and the final product. These costs are incurred from the planned start time of an activity (subassembly or final) until the delivery of the final product to the customer. Since each activity adds value to the final product, the total holding costs per time unit increase over time and are maximal at the final assembly activity. A penalty cost is incurred for a late delivery of the product to the customer. The objective of this chapter is to determine the planned leadtimes for all the activities such that the sum of expected holding and penalty costs is minimized.

Setting planned leadtimes to ensure timely delivery of customer orders is a key tactical decision for many companies (Atan et al., 2016). Multiple researchers have developed models to assist the companies with this decision. The distinguishing feature of our work is the way holding costs are accounted for. All previous work assumes that the holding costs are incurred from the actual start times of the activities. This so-called 'pay as realized' cost accounting scheme is realistic when materials are supplied exactly when they are needed, for example in a Just

In Time (JIT) environment. In this chapter, instead of incurring a holding cost from the actual start time of an activity, we start to incur holding cost of an activity from its planned start time. This is motivated by practice. Companies that rely on safety times instead of safety stocks to protect against uncertainties are mostly the ones that produce capital-intensive products. These companies allocate expensive material to production activities. The material is ordered from external suppliers or from other departments within the same company. The material is ready at the planned start times. If production cannot start at the planned start time, the material needs to be stored and interest for invested capital should be paid. These are the costs that we consider as holding costs in this chapter. As these cost need to be paid for, regardless of whether a production activity is started at its planned start time or not, we incur these holding cost from the planned start time. We refer to this cost accounting scheme as the 'pay as planned' scheme.

In addition to introducing a new holding cost accounting scheme, we contribute to the literature by introducing the concept of a 'blame policy', which can be applied to all planned leadtime optimization problems for any assembly system structure with stochastic leadtimes. For each realization of leadtimes, the blame policy identifies the activity that causes the late delivery of the product. For a system operating according to the optimal planned leadtime solution, we prove that the blame probability of each activity satisfies a Newsvendor equation. This equation states that the probability that an activity is blamed for late delivery is proportional to the value the activity adds to the final product. We end up with a set of Newsvendor equations, which can be numerically solved to obtain the unique optimal solution to the cost minimization problem.

We compare the 'pay as planned' cost function with the results obtained for the 'pay as realized' cost function used in Atan et al. (2016). The authors study the same configure-to-order system with the holding costs incurred from the actual start times of the activities. We derive structural results for the cost difference. We prove that our optimal solution allocates more time to each subassembly activity and less time to the final assembly activity compared to the optimal solution to the cost function of Atan et al. (2016). We present a numerical experiment illustrating that if the service level requirement is high, the difference between the optimal costs is marginal. However, the difference between the optimal planned leadtimes is significant. Our solution leads to a significantly higher probability that intermediate

deadlines are met.

Our numerical experiments also unveiled that in some cases the optimal solution of the 'pay as realized' cost function in Atan et al. (2016) can contain negative planned leadtimes. Negative planned leadtimes are counter-intuitive and can lead to difficulties when implementing the production plan in practice. We prove that for the 'pay as planned' cost function, the optimal planned leadtimes are always non-negative.

The remainder of this chapter is organized as follows. In Section 2.2 we provide a brief review of the literature on setting the planned leadtimes. In Section 2.3 we introduce notations and formulate the optimization problem. In Section 2.4 we define the blame policy and derive a set of Newsvendor equations. We describe properties of the cost function and the optimal solution in Section 2.5 and compare the optimal solution with the one obtained by Atan et al. (2016). Numerical results are presented in Section 2.6. We provide concluding remarks and discuss future research directions in Section 2.7.

2.2. Literature Review

Most of the earlier work on setting planned leadtimes focuses on single-activity systems and systems with specific structures. For a single-activity system, Weeks (1981) shows the equivalence of the planned leadtime problem to the well-known Newsvendor problem. Subsequently, Matsuura and Tsubone (1993); Matsuura et al. (1996) and Buzacott and Shanthikumar (1994) study the single-activity problem and conclude that safety times should be preferred over safety stock if the company has accurate forecasts on the shipments over the leadtime.

Multi-activity systems are more difficult to analyze as optimal decisions across activities are not independent. Earlier work on *serial* multi-activity systems includes Yano (1987a) and Gong et al. (1996). Yano (1987a) develops an algorithm to solve the problem of determining the optimal planned leadtimes for serial systems under the 'pay as realized' costing scheme. Gong et al. (1996) proves that this problem is mathematically equivalent to the problem of determining the optimal base-stock levels in serial inventory systems. Clark and Scarf (1960) solves this problem to optimality and provides a recursive algorithm to determine optimal base-stock

levels.

Elhafsi (2002) also studies a serial production system with the objective to determine the optimal planned leadtimes at each activity. Different from Yano (1987a) and Gong et al. (1996), Elhafsi (2002) penalizes tardiness of intermediate activities, while in the former ones the penalty cost is only incurred if the last activity is late. It can be shown that for specific values of the cost parameters, the optimal solution is identical to the solution under the 'pay as planned' costing scheme. The authors propose a recursive algorithm to compute the expected cost and solve the resulting convex nonlinear optimization problem to approximate the optimal planned leadtimes. Furthermore, the authors show that when all activities have an identical exponential leadtime distribution, an exact solution can be obtained.

For inventory systems, Rosling (1989) showed that an assembly system can be remodeled as a serial inventory system. This is useful, since results for the assembly system can be obtained by solving the equivalent serial system. For the planned leadtime problem the assembly system cannot be converted into an equivalent serial system. Therefore, solving the planned leadtime problem for assembly systems is more challenging than determining the optimal basestock levels in an assembly system. Research on setting the planned leadtimes for assembly systems has been initiated by Yano (1987b). The author considers a system with two subassemblies and a final assembly. The system incurs inventory holding costs for each activity from the moment the activity starts until the product is delivered to the customer. If the product is available after the promised delivery date, a penalty cost is charged per unit late. The objective is to find the optimal planned leadtimes. Yano (1987b) formulates the problem as a non-linear program. Although the cost function is not convex in all planned leadtimes, it has some properties that enable Yano (1987b) to solve the problem numerically. For systems with more than three activities, this approach often leads to computational problems.

After the seminal work by Yano (1987b) researchers developed approximation methods to solve the leadtime optimization problem for larger assembly systems. Hopp and Spearman (1993) present a model to determine the optimal leadtimes for purchasing components for a manufacturing system which performs the final assembly of these components. The authors develop an iterative approximation procedure to determine the optimal purchasing leadtimes. Shore (1995) studies the same problem and proposes an alternative procedure that results in closed-

form expressions for the decision variables. Song et al. (2001) develop a recursive procedure to estimate the distributions of activity leadtimes and propose a method to calculate activity due dates so that a specific service target is met. Considering cost a 'pay as realized' cost structure, Axsäter (2005) studies a multi-echelon assembly system and suggests a decomposition technique to set the activity start times. Different from Axsäter (2005), Chauhan et al. (2009) study a single-period setting. The authors develop an approximation procedure to set the activity start times. In Ben-Ammar et al. (2018) planned leadtimes for multi-level assembly systems are determined. The authors relax specific assumptions on cost parameters and develop a Branch and Bound algorithm to find optimal planned leadtimes for discrete activity leadtime distributions.

In another recent study, Atan et al. (2016) study an assembly system that consists of a number of parallel multi-activity stages feeding a multi-activity final assembly stages. Each process has a stochastic leadtime. From the moment a production activity starts until the final product is delivered to the customer, the system incurs a marginal holding cost and a penalty cost is charged for late deliveries to the customer. The authors derive recursive equations for the tardiness and earliness of all processes and determine an exact expression for the total expected cost. They develop an iterative heuristic procedure to calculate the optimal planned leadtimes. This procedure is based on a conjecture which claims that the probability that an activity is responsible for the lateness of the overall system is proportional to the value added by that activity.The conjecture forms the starting point of Chapter 4 in which we derive structural results for a more general system.

The research in this thesis initiated from a production planning perspective. We believe that project management problems can benefit from the results of studies on leadtime planning and vice versa. Although language and model formulation are different, both modeling approaches are similar. An example of a project planning approach to production planning is Márkus et al. (2003). The authors use a deterministic project planning model for two real-life production planning case studies.

Another relevant work in project planning is Ronen and Trietsch (1988). In this paper, the authors develop a model for purchasing materials for large projects. The model supports the decision regarding when and from whom to order each component. The model used for determining the optimal ordering moment for

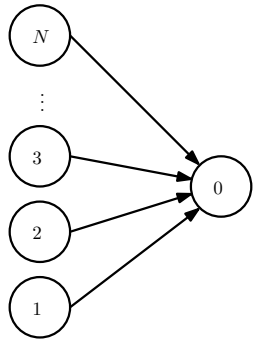


Figure 2.1: An assembly system with N subassembly activities and one final activity

each component can be interpreted as a model under the 'pay as planned' costing scheme. As all components should be ready at the same time, the network can be modeled as an assembly system with the final assembly node having a duration of zero.

Another interesting work is Trietsch (2006) which considers the planning of activities with uncertain durations and precedence constraints. The author shows that optimal buffering can be achieved by setting optimal release dates for all activities in a network. In particular, the criticality of a release date should match the relative cost of setting it earlier. Although the problem is modeled differently (the time buffers are seen as separate activities), the problem can be remodeled as a version of the 'pay as planned' costing scheme. Trietsch and Baker (2012) implement the optimization method of Trietsch (2006) in a decision support system for project planning.

2.3. Problem Formulation

We consider a configure-to-order assembly system consisting of N parallel activities delivering subassemblies to a single final assembly activity. The set of subassembly activities is $V = \{1, 2, ..., N\}$ and 0 is the final assembly activity (see Figure 2.1). Subassembly activities are initiated by a customer demand for a unit of product and the product is ready for delivery at completion of activity 0. Each activity $i \in \{0\} \cup V$ requires the execution of multiple tasks. We refer to the total time required to finish all tasks within an activity as the *leadtime* of this activity. Non-negative

random variable T_i with cumulative distribution $F_i(\cdot)$ and density $f_i(\cdot)$ represents the leadtime of activity i[1]. We assume that $f_i(t_i) > 0$ for all $t_i > 0$ and leadtimes at different activities are independent. Although we consider independent random variables, the results also hold for most dependent leadtimes. In Chapter 3 we derive conditions under which the independent leadtime assumption can be relaxed.

The system operates according to a plan that assigns a leadtime to each activity. This leadtime is called *the planned leadtime*. The planned leadtime of activity i is x_i. We define s_i as the planned start time of activity i. Without loss of generality, we set the planned customer delivery time to 0. Then the planned start times can be easily calculated from the planned leadtimes as

$$\begin{aligned} s_0 &= -x_0, \\ s_i &= -x_i - x_0 \quad i \in V. \end{aligned}$$

The actual start time of an activity is a random variable and depends on the actual leadtimes of all predecessors. For all activities with no predecessor, i.e. for all $i \in V$, we assume that all required resources are available at the planned start time and thus these activities can always start on time. However, their completion times are random. If $T_i = x_i$, then activity i is on time. Otherwise, it is either early or late. We define E_i and L_i as the earliness and lateness of activity i, respectively. Let $(x)^+ = \max\{0, x\}$. Then, for all $i \in V$,

$$\begin{aligned} E_i &= \Big(x_i - T_i\Big)^+, \\ L_i &= \Big(T_i - x_i\Big)^+. \end{aligned}$$

Activity 0 can only start after all subassemblies have been delivered. If one or more subassemblies are late, activity 0 starts immediately after the latest subassembly has been finished. If all subassemblies finish early, activity 0 starts at its planned start time s_0. This assumption of holding back production when preceding activities finish early is common in the literature (Yano, 1987a; Axsäter, 2005). It represents the situation where materials and workforce are available at the planned start time,

[1]Unless otherwise stated, all definitions are valid for $i \in \{0\} \cup V$

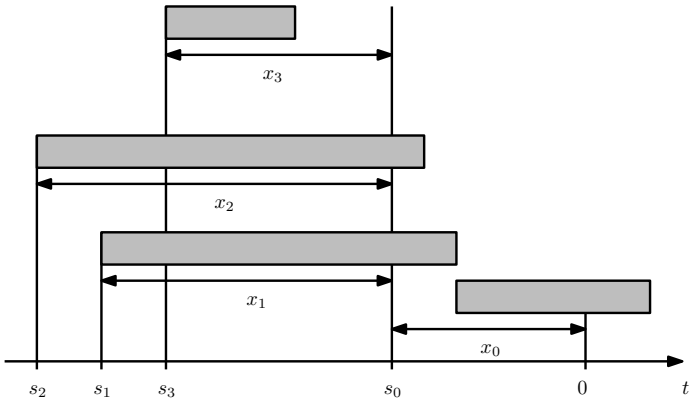


Figure 2.2: Timeline for a realization of a production plan in a four-activity assembly system

and not earlier. Hence, the earliness and lateness expressions for activity 0 are

$$\begin{aligned} E_0 &= \Big(x_0 - T_0 - \max_{i \in V}\{L_i\}\Big)^+, \\ L_0 &= \Big(T_0 + \max_{i \in V}\{L_i\} - x_0\Big)^+. \end{aligned}$$

We define W_i as the waiting time of activity i due to the lateness of other parallel activities,

$$W_i = \max_{j \in V}\{L_j\} - L_i, \quad i \in V.$$

Since W_0 has no parallel activities, $W_0 = 0$. Note that the random variables E_i, L_i and W_i depend on the vector of decision variables $\mathbf{x} = \{x_0, x_1, ..., x_N\}$. We do not explicitly indicate this dependence in the notation, except when it improves readability (as in Lemma 2.1, where we write $L_0(\mathbf{x})$). For an assembly system with 4 activities, a production plan and a realization of leadtimes are shown in Figure 2.2.

A realization of the random variables $T_0, T_1, ..., T_N$ is indicated by $\omega = (t_0, t_1, ..., t_N) \in \Omega = \mathbb{R}_+^{N+1}$. In particular, $T_i(\omega) = t_i$ is a realization of T_i.

The timeline in Figure 2.2 shows the planned start and finish time for each activity. The gray horizontal bars denote the realizations $T_i(\omega)$ of the random variables T_i.

In this case, activity 1 and 2 finish late, while activity 3 finishes early. Activity 1 is the latest one and subassemblies produced at activities 2 and 3 have to wait for activity 1. The final assembly activity starts at $s_0 + L_1(\omega)$ and finishes later than its planned finish time. Thus the final product is delivered late.

The system incurs a marginal holding cost $h_i > 0$ from the planned start time of activity i until the final product is delivered to the customer. The total holding costs per time unit at activity i are h_i^c. We have $h_i^c = h_i$ for $i \in V$ and $h_0^c = \sum_{i \in V} h_i^c + h_0$. In addition to the holding costs, the system incurs a penalty cost p per unit time late for delivery to the customer. The total holding costs for the planned duration of activity i is $h_i^c x_i$ for $i \in \{0\} \cup V$. If the product is delivered late, in addition to the penalty cost, a holding cost h_0^c is also charged per unit late. Hence, the total costs for lateness of the final activity are $(h_0^c + p)L_0$. Let $\mathbf{x} = (x_0, \ldots, x_N)$ be the vector of all planned leadtimes. We define $C(\mathbf{x})$ as the total expected cost of a product produced according to the planned leadtimes $\mathbf{x}$. Then

$$C(\mathbf{x}) = \sum_{i=0}^{N} h_i^c x_i + (h_0^c + p)\mathbb{E}[L_0]. \tag{2.1}$$

The cost $C(\mathbf{x})$ is composed of two terms. The first term is the deterministic term of the planned leadtimes. The second one is the variability term depending on stochastic and planned leadtimes. The objective is to determine the planned leadtimes that minimize the total expected cost $C(\mathbf{x})$.

2.4. Solution Approach

Our solution approach is based on solving a set of Newsvendor equations. We show that the probability that an activity is 'blamed' for the lateness of the system equals a Newsvendor fractile. We first explain the concept of blaming by defining a blame policy in Section 2.4.1 and then derive the corresponding Newsvendor equations for our cost function in Section 2.4.2.

2.4.1 Blame Policy

When a product is delivered late, one may ask which activity or activities caused the lateness. For each late delivery, there are one or more activities with leadtimes exceeding their planned leadtimes. These activities are candidates to be blamed. In this section, we define a policy to identify the single activity to be blamed for the lateness of the system for a given realization of the random leadtimes. We call this policy *the blame policy*. For each possible event that results in lateness of the system, the blame policy identifies the activity to be blamed for the lateness.

Next, we define a set of rules, referred as *blame conditions*. For each activity $i \in \{0\} \cup V$, we define the event θ_i as the set of realizations $\omega = (t_0, t_1, \ldots, t_N)$ satisfying the following blame conditions:

1. Activity i starts at its planned start time.
2. The leadtime $T_i(\omega)$ of activity i exceeds its planned leadtime, i.e., $T_i(\omega) > x_i$.
3. The successor of activity i is able to start immediately after activity i finishes, i.e., $W_i(\omega) = 0$.
4. The final product is delivered late, i.e., $L_0(\omega) > 0$.

Equivalently, we have

$$\begin{aligned} \theta_0 &= \{\omega : \max_{j \in V}\{L_j(\omega)\} = 0, L_0(\omega) > 0\}, \\ \theta_i &= \{\omega : T_i(\omega) > x_i, \max_{j \in V}\{L_j(\omega)\} = L_i(\omega), L_0(\omega) > 0\}, \quad i \in V. \end{aligned}$$

Event θ_i represents the set of all realizations ω in which activity i is blamed for the late delivery of the final product. Note that an activity can only be blamed if the final product is late, i.e., $L_0(\omega) > 0$. If a subassembly activity finishes late, but this lateness is compensated by the final activity, none of the activities is blamed. Also, if both the subassembly activity i and activity 0 exceed their planned leadtimes, i.e., $T_i(\omega) > x_i$ and $T_0(\omega) > x_0$, only activity i is blamed. Activity 0 can only be blamed if it starts on time, that is $L_i(\omega) = 0$ for all $i \in V$, and the final product is late. Finally, since we assume the random variable T_i to be continuous, the events θ_i are mutually exclusive for all $i \in \{0\} \cup V$ and together cover all possible realizations

$\omega = (t_0, t_1, \ldots, t_N)$ for which delivery is late. For all $i \in \{0\} \cup V$ we define Ω as the supporting probability space of $P(\theta_i)$ and hence it follows that $P(L_0 > 0) = \sum_{i \in \{0\} \cup V} P(\theta_i)$.

For a given realization of the random variables, the blame policy identifies the specific activity that is responsible for the late delivery of the final product. As an example, consider the realization in Figure 2.2. The activity to be blamed is activity 1, because it starts on time, exceeds its planned leadtime, activity 0 starts immediately after activity 1 finishes and delivery is late. Activity 2 also exceeds its planned leadtime, but cannot be blamed, since activity 0 starts after activity 1 finishes, i.e., Condition 3 is violated. Activity 3 cannot be blamed because conditions 2 and 3 are violated, $T_3(\omega) < x_3$ and $W_3(\omega) > 0$. Activity 0 cannot be blamed, because it starts later than planned and thus Condition 1 is violated.

It is crucial to note that the blame policy is independent of the cost function. It can be applied to all planned leadtime optimization problems for all assembly systems that face stochastic leadtimes and have the structure as shown in Figure 2.1. The goal of the blame policy is to define a set of rules which identify a single node that is responsible for the lateness of the final product.

It is possible to define other blame policies. For example, one blame policy might be that no matter what happens with the leadtimes of the subassemblies, the final assembly is always responsible for the lateness of the system. Even if the final activity starts late due to its predecessors, we still blame this node. The optimal planned leadtimes under this alternative blame policy and our blame policy are *the same*. Note that our policy is fairer compared to this alternative policy since each node can be responsible regardless of where it is located in the network.

What makes our blame policy special and different from its alternatives is that it provides an insight on the optimality equations of the 'pay as planned' cost function. These optimality equations, i.e. equations satisfied by the set of planned leadtimes that minimize (2.1), relate the probability that an activity is blamed to the value that activity adds to the final product. Hence, under the optimal solution of the 'pay as planned' cost function, the probability that each node is blamed is equal to a Newsvendor fractile. We provide the details in Section 2.4.2.

In Chapter 4 we study the 'pay as realized' cost function and define an alternative blame policy in Section 4.5. We show that this alternative policy plays a crucial role

in the optimality equations of the 'pay as realized' cost function.

2.4.2 Newsvendor Equations

Newsvendor equations originate from the field of inventory management. A Newsvendor equation describes the optimal *stockout* probability and it can be used to calculate the optimal order-up-to level. At optimality, penalty and inventory holding costs are balanced. For our problem, Newsvendor equations describe the probability that a specific activity is blamed for late delivery. The optimal planned leadtimes can be obtained by solving these Newsvendor equations. The optimal solution balances holding costs for being early and penalty costs for being late. Summing these Newsvendor equations over all activities results in an equation describing the probability that the product is delivered late. We call this 'the general Newsvendor equation.'

We define $\mathbf{x}^*$ as the optimal planned leadtime solution, where superscript $*$ is used to denote state variables at optimality. For example, $L_i^* = (T_i - x_i^*)^+$ is the lateness at activity i when the planned leadtimes are set to their optimal values. Lemma 2.1 presents the partial derivatives of $\mathbb{E}[L_0]$. The proof of this lemma and those of all other results in this chapter are deferred to Appendix 2.A.

Lemma 2.1

$$\frac{\partial \mathbb{E}[L_0(\mathbf{x})]}{\partial x_0} = -\sum_{i=0}^{N} P(\theta_i), \quad \frac{\partial \mathbb{E}[L_0(\mathbf{x})]}{\partial x_i} = -P(\theta_i), \quad i \in V. \tag{2.2}$$

From (2.1) and Lemma 2.1 it follows that the gradient of $C(\mathbf{x})$ is given by

$$\begin{aligned} \nabla C(\mathbf{x}) &= \left(\frac{\partial C(\mathbf{x})}{\partial x_0}, \ldots, \frac{\partial C(\mathbf{x})}{\partial x_N} \right) \\ &= \left(h_0^c - (h_0^c + p) \sum_{i=0}^{N} P(\theta_i), h_1^c - (h_0^c + p) P(\theta_1), \ldots, h_N^c - (h_0^c + p) P(\theta_N) \right). \end{aligned}$$

$\nabla C(\mathbf{x})$ vanishes at optimality. Hence, we obtain the following optimality equations,

i.e., equations that are satisfied by the optimal solution $\mathbf{x}^*$

$$P(\theta_i) = \frac{h_i^c}{h_0^c + p} = \frac{h_i}{h_0^c + p}, \quad i \in V \tag{2.3}$$

and

$$\sum_{i=0}^{N} P(\theta_i) = \sum_{i=0}^{N} \frac{h_i}{h_0^c + p} = \frac{h_0^c}{h_0^c + p}. \tag{2.4}$$

The left-hand side of (2.4) is the probability that the product is delivered late, that is, $P(L_0^* > 0)$. This equation is referred to as the general Newsvendor equation. Subtracting (2.3) from (2.4) yields

$$P(\theta_0) = \frac{h_0}{h_0^c + p}. \tag{2.5}$$

These findings are summarized in the next theorem stating that at optimality, the probability that activity i is blamed for late delivery is proportional to the value h_i this activity adds to the final product.

Theorem 2.1 *The optimal planned leadtime solution $\boldsymbol{x}^*$ of the unconstrained optimization problem $\min_{\boldsymbol{x}} C(\boldsymbol{x})$ satisfies the following set of Newsvendor equations:*

$$P(\theta_i) = \frac{h_i}{h_0^c + p}, \quad i \in \{0\} \cup V. \tag{2.6}$$

Clearly, if h_i is high, it is costly to finish activity i early as holding cost will accumulate. Hence, under the optimal solution, activities which add high value to the final product are planned such that, with high probability, they are to be blamed for the late delivery.

The blame probabilities in Theorem 2.1 can be formulated in terms of the leadtime densities and decision variables $\mathbf{x}$. For activity $i \in V$ we have

$$P(\theta_i) \quad = \quad P\Big(T_i > x_i, W_i = 0, L_0 > 0\Big), \tag{2.7}$$

where the random variables W_i and L_0 depend on the leadtime T_i. Conditioning on

T_i yields

$$\begin{aligned}
P(\theta_i) &= \int_{x_i}^{\infty} P\Big(W_i = 0, L_0 > 0 | T_i = t_i\Big) f_i(t_i) dt_i \\
&= \int_{x_i}^{\infty} P\Big(\max_{j \in V}\{L_j\} = L_i, L_0 > 0 | T_i = t_i\Big) f_i(t_i) dt_i \\
&= \int_{x_i}^{\infty} \prod_{j=1, j\neq i}^{N} P\Big(T_j - x_j < t_i - x_i\Big) P\Big(t_i - x_i + T_0 - x_0 > 0\Big) f_i(t_i) dt_i \\
&= \int_{x_i}^{\infty} \prod_{j=1, j\neq i}^{N} \left(\int_{0}^{x_j + t_i - x_i} f_j(t_j) dt_j \right) \left(\int_{x_0 - (t_i - x_i)}^{\infty} f_0(t_0) dt_0 \right) f_i(t_i) dt_i.
\end{aligned}$$

For activity 0 we get

$$\begin{aligned}
P(\theta_0) &= P(\max_{j \in V}\{L_j\} = 0, L_0 > 0) \\
&= \prod_{j=1}^{N} P(T_j - x_j < 0) P(T_0 - x_0 > 0) \\
&= \prod_{j=1}^{N} \left(\int_{0}^{x_j} f_j(t_j) dt_j \right) \int_{x_0}^{\infty} f_0(t_0) dt_0.
\end{aligned}$$

The above expressions can be used to numerically solve the optimal planned leadtimes from the Newsvendor equations in Theorem 2.1.

2.5. Structural Results

Our cost function fundamentally differs from cost functions studied in the literature. In literature most papers use a 'pay as realized' cost accounting scheme, while we introduced the 'pay as planned' cost accounting scheme. This has implications for the structure of the optimal solution. In Section 2.5.1, we describe properties of the 'pay as planned' cost function and its optimal solution. In Section 2.5.2 we compare our optimal solution with the one obtained by Atan et al. (2016) for the 'pay as realized' cost function.

2.5.1 Properties of the Cost Function and the Optimal Solution

Since $\min_{\mathbf{x}} C(\mathbf{x})$ is an unconstrained optimization problem, one or more planned leadtimes of the optimal solution may be negative. For cost functions studied in the literature, this turns out to be possible, as shown in Section 2.6.2. Negative planned leadtimes, however, are often not accepted in practice: it is counter-intuitive to have the planned finish time before the planned start time of a task. The next theorem states that, for the 'pay as planned' cost function, vectors $\mathbf{x}$ with negative planned leadtimes are never optimal.

Theorem 2.2 *For any* $\mathbf{x}$ *with at least one negative planned leadtime* x_i, $i \in \{0\} \cup V$, *there exists a vector of non-negative planned leadtimes with lower cost.*

This theorem implies that the optimization problem can be restricted to $\mathbf{x} \geq 0$. Since $C(\mathbf{x}) \geq \sum_{i=0}^{N} h_i^c x_i$, the cost of the all-zero vector $\mathbf{0}$ is less than the cost of any $\mathbf{x}$ with $\sum_{i=0}^{N} h_i^c x_i > C(\mathbf{0})$. Hence, optimization can be further restricted to the bounded region $\{\mathbf{x} : \mathbf{x} \geq 0, \sum_{i=0}^{N} h_i^c x_i \leq C(\mathbf{0})\}$. Since $C(\mathbf{x})$ is continuous, we can conclude that the optimal $\mathbf{x}^*$ indeed exists.

Proposition 2.1 *The optimal solution* $\mathbf{x}^*$ *of* $\min_{\boldsymbol{x}} C(\boldsymbol{x})$ *exists and is non-negative.*

Using that the solution space can be restricted to non-negative planned leadtimes, we next derive monotonicity properties for the blame probabilities of different activities. Lemma 2.2 holds for any $i \in V$, while Lemma 2.3 holds for activity 0.

Lemma 2.2 *For activity* $i \in V$ *and* $\mathbf{x} \geq 0$, *the probability* $P(\theta_i)$ *is strictly decreasing in* x_i *and* x_0 *and strictly increasing in* $x_j, j \in V, j \neq i$.

Lemma 2.3 *For activity* 0 *and* $\mathbf{x} \geq 0$, *the probability* $P(\theta_0)$ *is strictly decreasing in* x_0 *and strictly increasing in* $x_i, i \in V$.

According to the above lemmas, blame probability $P(\theta_i)$ is decreasing in x_0 and x_i and increasing in any other planned leadtime. Hence, final activity 0 is the only activity for which a larger planned leadtime x_0 leads to a reduction of all blame probabilities.

Theorem 2.1 states that the optimal solution $\mathbf{x}^*$ satisfies a set of Newsvendor equations. The monotonicity results in Lemma 2.2-2.3 are instrumental to establish the following theorem.

Theorem 2.3 *The Newsvendor equations* (2.6) *have a unique non-negative solution.*

Corollary 2.1 *The optimal solution* $\mathbf{x}^*$ *of* $\min_{\boldsymbol{x}} C(\boldsymbol{x})$ *is unique.*

2.5.2 Comparison with 'Pay as Realized' Cost Function

As explained in Section 2.1, our cost function charges the holding costs from the planned start time of the activities while in other studies the holding cost is charged only after the activities actually start. Except for this difference in the cost function, the modeling assumptions in Atan et al. (2016) are exactly the same as ours. In this section, we provide an analytical comparison of the cost functions.

Let $C^a(\mathbf{x})$ denote the total expected cost used in Atan et al. (2016). The expression for $C^a(\mathbf{x})$ is

$$C^a(\mathbf{x}) = \sum_{i=0}^{N} h_i^c x_i - h_0 \mathbb{E}\left[\max_{i\in V}\{L_i\}\right] + (h_0^c + p)\mathbb{E}[L_0] \tag{2.8}$$

The only difference between (2.1) and (2.8) is the second term of the cost function $C^a(\mathbf{x})$. This term contains the random variable $\max_{i\in V}\{L_i\}$, which is the lateness of the latest subassembly. Atan et al. (2016) exclude the expected holding cost that might be incurred at activity 0 during the time it waits for all subassemblies to finish. This is exactly the difference indicated in Figure 1.3.

At first sight, the 'pay as planned' cost function looks simpler. However, it is more challenging to solve. Atan et al. (2016) rely on the fact that, under their accounting scheme the optimality equations are decoupled. This is why they can use a recursive procedure to solve for the optimal planned leadtimes. On the other hand, under the new accounting scheme, the optimality equations are not decoupled. Therefore, we cannot use the recursive procedure to solve for the optimal planned leadtimes. This is why a new solution procedure is required.

We define $\mathbf{x}^{\mathbf{a}*}$ as the optimal solution to the optimization problem $\min_{\mathbf{x}} C^a(\mathbf{x})$. For any planned leadtime vector $\mathbf{x}$ we can compute the expected costs using (2.1) and (2.8). Since our cost function penalizes the lateness of intermediate time points

while $C^a(\mathbf{x})$ does not, our function always leads to higher expected costs. This result is stated in the following proposition:

Proposition 2.2 *For the cost functions $C(\cdot)$ and $C^a(\cdot)$ the following inequalities hold:*

1. $C(\mathbf{x}) \geq C^a(\mathbf{x})$ *for all* $\mathbf{x}$*;*
2. $C(\mathbf{x}^*) \geq C^a(\mathbf{x}^{\mathbf{a}*})$.

Obviously, $\mathbf{x}^*$ and $\mathbf{x}^{\mathbf{a}*}$ are not necessarily the same. The next proposition compares the two optimal planned leadtimes.

Proposition 2.3 *For the optimal planned leadtime solutions $\mathbf{x}^*$ and $\mathbf{x}^{\mathbf{a}*}$ the following relations hold:*

1. $x_0^* < x_0^{a*}$ *and* $x_i^* > x_i^{a*}$ *for all* $i \in V$*;*
2. *There exist* $i \in V$ *such that* $x_i^* + x_0^* \geq x_i^{a*} + x_0^{a*}$.

Proposition 2.3(a) states that solution $\mathbf{x}^*$ allocates more time to each of the subassembly activities and allocates less time to the final assembly activity compared to $\mathbf{x}^{a*}$. As a consequence, the solution $\mathbf{x}^*$ ensures that the intermediate deadline is met more often. This is in line with the intended practical applications of both solutions. Solution $\mathbf{x}^*$ should be used when exceeding the intermediate deadline is costly because materials are waiting. Solution $\mathbf{x}^{\mathbf{a}*}$ should be used when materials are delivered JIT. Part (b) implies that for a two-activity serial system (i.e., $N = 1$), the total leadtime is longer for our solution. This is an intuitive result, since safety time added to a subassembly activity cannot compensate for lateness in the final assembly, while safety time in the final assembly can compensate for lateness in both the subassembly and the final assembly.

The difference between our expected cost and the one in Atan et al. (2016) is $h_0 \mathbb{E}\left[\max_{i \in V}\{L_i\}\right]$. Hence, an important parameter driving differences between $\mathbf{x}^*$ and $\mathbf{x}^{\mathbf{a}*}$ is h_0. Below we study for a two-activity serial system the asymptotic behavior of the optimal planned leadtimes as $h_0 \to \infty$, while keeping service level α constant. The service level is defined as

$$\alpha = P(L_0 = 0) = \frac{p}{h_0^c + p}.$$

As h_0 changes, we keep service level α constant by adapting $p = \frac{\alpha}{1-\alpha} h_0^c$ (and not changing h_1).

Proposition 2.4 *For a two-activity serial system, as $h_0 \to \infty$ while keeping service level α constant:*

$$\begin{aligned}
\lim_{h_0 \to \infty} \{x_0^{a*}\} &= F_0^{-1}(\alpha), \\
\lim_{h_0 \to \infty} \{x_1^{a*}\} &= \infty, \\
\lim_{h_0 \to \infty} \{x_0^{*}\} &= F_0^{-1}(\alpha), \\
\lim_{h_0 \to \infty} \{x_1^{*}\} &= \infty.
\end{aligned}$$

We see that the asymptotic behavior of both optimal solutions is similar. The optimal planned leadtimes of the final activity approach the same limiting value. The leadtimes of the first activity grow without bound, which is due to the blame probability $P(\theta_1) = \frac{h_1}{h_0^c + p}$ approaching to 0. Hence, in the limit, lateness of the complete system is equal to lateness of the final activity. The above proposition does not provide information on the asymptotic behavior of the difference between the optimal planned leadtimes of the first activity. Numerical experiments indicate that $x_1^* - x_1^{a*}$ does converge, though not to 0 (as for the final activity) but to a positive number.

We obtain additional analytical results for the case with exponentially distributed leadtimes. We consider a two-activity system. In the following propositions we provide the optimality equations for both cost functions.

Proposition 2.5 *For a two-activity serial system with independent and identically distributed exponential leadtimes with rate λ, the optimal solution $\mathbf{x}^*$ satisfies the following set of Newsvendor equations:*

$$\begin{aligned}
(1 + \lambda x_0^*) e^{-\lambda(x_0^* + x_1^*)} &= \frac{h_1}{h_0^c + p} \\
e^{-\lambda x_0^*} - e^{-\lambda(x_0^* + x_1^*)} &= \frac{h_0}{h_0^c + p}
\end{aligned}$$

Proposition 2.6 *For a two-activity serial system with independent and identically distributed exponential leadtimes with rate λ, the optimal solution $\mathbf{x}^{\mathbf{a}*}$ satisfies the following*

set of Newsvendor equations:

$$\lambda x_0^{a*} e^{-\lambda(x_1^{a*}+x_0^{a*})} = \frac{h_1}{h_0^c + p}$$
$$e^{-\lambda x_0^{a*}} = \frac{h_0}{h_0^c + p}$$

The expressions in Proposition 2.6 suggests that we can obtain x_0^{a*} and x_1^{a*} recursively. On the other hand, the expressions in Proposition 2.5 are not decoupled. Hence, under our cost accounting scheme, obtaining the optimal solution is more challenging.

In addition to the optimality equations, for the special case with exponential leadtimes, we can determine the expression for the difference $x_1^* - x_1^{a*}$ as $h_0 \to \infty$.

Proposition 2.7 *For a two-activity serial system with independent and identically distributed exponential leadtimes with rate λ, we have*

$$\lim_{h_0 \to \infty} \{x_1^* - x_1^{a*}\} = \frac{1}{\lambda}(\ln(\lambda x_0^* + 1) - \ln(\lambda x_0^*)).$$

2.6. Numerical Analysis

Section 2.5.2 outlines differences between our optimal solution and the one in Atan et al. (2016). In this section we provide the numerical results which do not only quantify differences in optimal costs but also structural differences in the optimal solutions.

2.6.1 Cost Comparison

In this section we provide a representative example to show differences between the optimal solutions and the corresponding optimal costs. We consider an assembly system with two subassemblies (activity 1 and 2) and a final assembly activity (activity 0). We assume equal echelon unit holding costs for all activities, thus $h_0 = h_1 = h_2 = 1$. We set the unit penalty costs to $p = 27$. Hence, under the optimal solution, the service level is $\alpha = \frac{p}{h_0^c + p} = 90\%$. We assume that leadtimes

	x_0	x_1, x_2	Total	$P(\max\{L_1, L_2\} > 0)$	$P(L_0 > 0)$
$\mathbf{x}^*$	3.01	1.73	4.75	0.32	0.1
$\mathbf{x}^{\mathbf{a}*}$	3.40	1.18	4.58	0.52	0.1

Table 2.1: Structural differences between the two optimal planned leadtime solutions.

$C(\mathbf{x}^*)$	$C^a(\mathbf{x}^*)$	$C(\mathbf{x}^{\mathbf{a}*})$	$C^a(\mathbf{x}^{\mathbf{a}*})$
16.03	15.69	16.18	15.61

Table 2.2: Expected costs under the optimal planned leadtime solutions.

are exponentially distributed with rate $\lambda_i = 1$ for all $i \in \{0\} \cup V$. Table 2.1 provides the optimal planned leadtimes and some properties of the optimal solutions and Table 2.2 summarizes the expected total costs.

The expected costs for each of the optimal leadtime solutions $\mathbf{x}^*$ and $\mathbf{x}^{\mathbf{a}*}$ can be computed according to both cost functions $C(\cdot)$ and $C^a(\cdot)$. These are also shown in Table 2.2. Suppose a company uses cost function $C^a(\cdot)$ and determines the optimal solution $\mathbf{x}^{\mathbf{a}*}$. The total expected cost for this solution is 15.61. However, this cost function does not take into account costs that occur if activity 0 cannot start in time. Cost function $C(\cdot)$ does take these additional costs into account. Given that the production is planned according to solution $\mathbf{x}^{\mathbf{a}*}$, the expected cost according to cost function $C(\cdot)$ will be 16.18. This is an increase of 3.6%. Thus, for this case, cost function $C^a(\cdot)$ neglects a significant part of the total expected costs. However, the solution $\mathbf{x}^{\mathbf{a}*}$ is not optimal for cost function $C(\cdot)$. If a company would optimize according to this cost function, the solution is $\mathbf{x}^*$. This leads to a total costs of 16.03, which implies an improvement of 0.9%.

One could argue that an improvement of 0.9% in expected costs is relatively small and thus that both solutions $\mathbf{x}^{\mathbf{a}*}$ and $\mathbf{x}^*$ can be used. There are a few reasons why the cost difference is this small. First, the holding cost during the average duration of each activity, i.e., $\sum_{i=0}^{N} h_i^c \mathbb{E}[T_i]$, appears in both cost functions and it is constant. In this example, the value of this constant term is 5.0, i.e., almost one third of the total costs. Second, the cost for being late at the final activity are $(h_0^c + p)\mathbb{E}[L_0]$. A high service level requirement, such as 90%, makes this term significant. Under solution $\mathbf{x}^*$ this term equals 3.52 (22.0% of the total costs), while under solution $\mathbf{x}^{\mathbf{a}*}$ this term equals 3.61 (22.3% of the total costs).

Although the costs for both solutions are close, there are significant differences in the structure of the optimal solutions. In Table 2.1, we provide the planned leadtimes of each activity and lateness probabilities. We observe that $\mathbf{x}^*$ allocates significantly less time to the final activity (3.01 vs 3.40). Since activities 1 and 2 have equal leadtime distributions and equal holding costs, these activities have the same optimal planned leadtimes in both solutions. We see that $\mathbf{x}^*$ plans more time for the subassembly activities (1.73 versus 1.18), a difference of 46%. This has a significant impact on the probability that one or more subassemblies finish late. This probability is 0.32 if $\mathbf{x}^*$ is used while it is 0.52 if $\mathbf{x}^{\mathbf{a}*}$ is used. Hence, the probability that the final assembly can start in time is 0.68 for our solution. This probability is 0.48 for the other solution. Therefore, the advantage of using $\mathbf{x}^*$ instead of $\mathbf{x}^{\mathbf{a}*}$ is that intermediate deadlines are met more often, while expected costs and total leadtime only change marginally. From a production planner's perspective, this is useful, since it makes scheduling resources for different products easier.

The insights of this numerical example can be generalized to other parameter combinations. The main driver in the cost difference between our solution and the one by Atan et al. (2016) is the relative importance of meeting the intermediate deadline (start of final assembly) compared to the final deadline (delivery to the customer). A high $h_0/(h_1+h_2)$ ratio makes the intermediate deadline important, while a high p/h_0^c ratio makes the final deadline important. Increasing h_0 while keeping other parameters constant increases the importance of the intermediate deadline and decreases the importance of the final deadline. This leads to higher cost differences. However, in such an example, the service level would decrease significantly. For example, when $h_1 = h_2 = 1$, $h_0 = 10$ and $p = 27$, the optimal service level is 27/39=69%.

Similarly, increasing p while keeping other parameters constant makes the final deadline important and reduces the cost difference between our solution and the one by Atan et al. (2016). To show this, we perform a numerical experiment in which we vary p. We consider a two-activity serial system with exponentially distributed leadtimes: $\lambda_1 = \lambda_0 = 1$. Holding cost parameters are $h_1 = h_0 = 1$. We calculate the optimal solutions $\mathbf{x}^*$ and $\mathbf{x}^{\mathbf{a}*}$ and the corresponding expected cost $C(\mathbf{x}*)$ and $C^a(\mathbf{x}^{\mathbf{a}*})$. In Figure 2.3 the relative cost difference, defined as $\frac{C(\mathbf{x}^*)-C^a(\mathbf{x}^{\mathbf{a}*})}{C(\mathbf{x}^*)}$, is shown. According to this figure, under the chosen parameter setting, for very low values of p the relative cost difference can be as high as 17%, while it is around 3% for higher

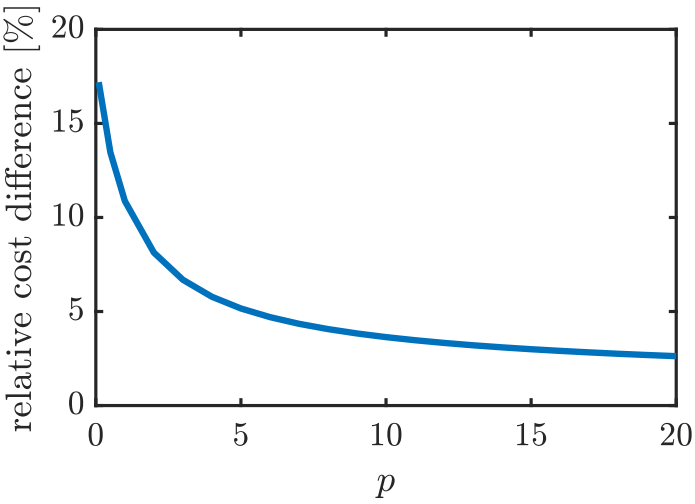


Figure 2.3: Relative cost difference between the two cost functions.

p values. We would like to note that obtaining analytical expressions for the relative cost difference under the optimal solutions is not possible even for this special case with exponential leadtimes. The challenge stems from the fact that the optimality equations under our cost accounting scheme are not decoupled.

2.6.2 Negative Planned Leadtimes

As stated in Proposition 2.1, the planned leadtimes in our optimal solution are always non-negative. However, this is not case for the optimal solution by Atan et al. (2016). In this section, we show that $\mathbf{x}^{\mathbf{a}*}$ can have negative components.

We consider a two-activity serial system with exponentially distributed leadtimes with rate $\lambda_i, i = 0, 1$. We set $\lambda_1 = 1$ and solve multiple instances by varying λ_0, and hence $\mathbb{E}[T_0] = \lambda_0^{-1}$. In all instances $h_0 = h_1 = 1$ and the desired service level $\alpha = 90\%$. In Figure 2.4, we plot x_1^* and x_1^{*a} as function of $\mathbb{E}[T_0]$.

When the average leadtime of activity 0 gets much longer than the average leadtime of activity 1, the planned lead time x_1^{a*} decreases and becomes negative. On the other hand, x_1^* stays positive and close to $\mathbb{E}[T_1]$. Hence, when $\mathbb{E}[T_0] > \mathbb{E}[T_1]$, our cost function leads to a meaningful optimal solution. When $\mathbb{E}[T_1] > \mathbb{E}[T_0]$, both cost functions lead to a similar optimal solution.

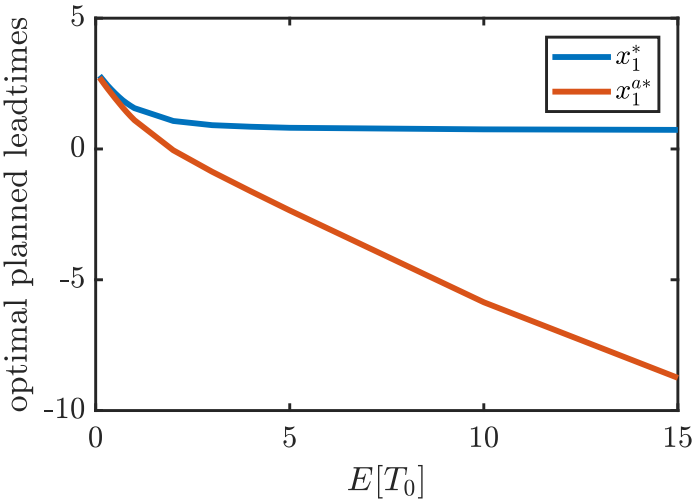


Figure 2.4: Optimal planned leadtimes for activity 1 for different values of $\frac{1}{\lambda_0}$.

The same behavior can be observed for symmetric assembly systems (equal holding costs and identical distributions for all subassemblies) by comparing $\max_{i \in V}\{T_i\}$ to T_0. For asymmetric assembly systems, a negative optimal planned leadtime is typically expected for a subassembly activity i with $\mathbb{E}[T_0] >> \mathbb{E}[T_i]$.

When an optimal solution contains negative planned leadtimes, this means that the activity is planned to start later than its successor is planned to start. To bypass this problem in practice, the plan is adapted as follows. In the adapted plan $x_1^{a'} = 0$ and $x_0^{a'} = x_0^a + x_1^a$. This doesn't change the realization of the plan. Activity 1 will start at the same time and activity 0 will start immediately after it. These alternative solutions that for the 'pay as realized' cost function lead to the same cost are further discussed in Chapter 4.

2.7. Concluding Remarks

In this chapter we study the problem of setting planned leadtimes in a configure-to-order assembly system with random leadtimes. We introduce the 'pay as planned' cost accounting scheme, which besides holding costs and a penalty cost for late delivery, also accounts for not meeting intermediate deadlines. The objective is to determine the optimal planned leadtimes, which minimize the total expected cost.

Our methodology relies on introducing a blame policy. Given that a product is delivered late, this policy identifies which activity should be blamed for the lateness. We show that for a system that is planned according to the optimal planned leadtime solution, the probability that a activity is blamed is proportional to the value the activity adds to the final product. We prove this statement by deriving a set of Newsvendor equations, that are only satisfied by the optimal planned leadtime solution.

The same system with a 'pay as realized' cost function has been studied in Atan et al. (2016). We compare our optimal solution with their optimal solution and provide analytical results to point multiple structural differences. In addition, the two solutions are compared in numerical experiments. These experiments show that if the service level requirement is high, the difference between the optimal costs is marginal. However, the difference in the optimal planned leadtimes is significant. Our solution results in a significantly higher probability that intermediate deadlines are met. Also, negative optimal planned leadtimes, a problem faced when using the other cost function, are not possible for our cost function.

In this chapter we introduced the 'pay as planned' cost accounting scheme for an assembly system. In Chapter 4 we extend the definition of the framework to more general networks and generalize the structural results. This chapter also considers systems with dependent leadtimes. As leadtimes for different suppliers often correlate, this is a very relevant research direction. Finally, note that we do not penalize the activities for being blamed for the lateness of the system. Another future work might be to consider a cost function such that each activity is penalized for the late delivery. Then, different from our study, the optimal solution would depend on the blame policy. Finding the optimal blame policy can be challenging yet very interesting and relevant extension.

2.A. Proofs

Proof of Lemma 2.1

First we show that

$$\frac{\partial \mathbb{E}[L_0(\mathbf{x})]}{\partial x_0} = \mathbb{E}\left[\frac{\partial L_0(\mathbf{x})}{\partial x_0}\right],$$

or equivalently, that for every sequence $h_1, h_2, \ldots$ converging to 0,

$$\lim_{n\to\infty} \frac{\mathbb{E}[L_0(\mathbf{x}+h_n\mathbf{e}_0)] - \mathbb{E}[L_0(\mathbf{x})]}{h_n} = \mathbb{E}\left[\frac{\partial L_0(\mathbf{x})}{\partial x_0}\right], \tag{2.9}$$

where $\mathbf{e}_0$ is the unit vector with 1 at position 0. Since

$$L_0(\mathbf{x})(\omega) = \left(t_0 + \max_{i\in V}(t_i - x_i)^+ - x_0\right)^+$$

, the derivative $\frac{\partial L_0(\mathbf{x})}{\partial x_0}(\omega)$ exists for almost all ω and

$$\frac{\partial L_0(\mathbf{x})}{\partial x_0}(\omega) = \begin{cases} -1, & t_0 + \max_{i\in V}(t_i - x_i)^+ - x_0 > 0, \\ 0, & t_0 + \max_{i\in V}(t_i - x_i)^+ - x_0 < 0. \end{cases}$$

For all ω,

$$\left|\frac{L_0(\mathbf{x}+h_n\mathbf{e}_0)(\omega) - L_0(\mathbf{x})(\omega)}{h_n}\right| \leq 1,$$

so by bounded convergence we can conclude that (2.9) holds. Hence,

$$\frac{\partial \mathbb{E}[L_0(\mathbf{x})]}{\partial x_0} = \mathbb{E}\left[\frac{\partial L_0(\mathbf{x})}{\partial x_0}\right] = -P(L_0 > 0) = -P(\bigcup_{i=0}^{N}\theta_i) = -\sum_{i=0}^{N} P(\theta_i).$$

This proves the lemma for the derivative of $\mathbb{E}[L_0]$ with respect to x_0. The proof of the derivatives with respect to x_i for $i \in V$ proceeds along the same lines.

Proof of Theorem 2.2

We consider a planned leadtime vector $\mathbf{x}$. If $x_i < 0$ for some $i \in V$, we can write L_0 as

$$L_0 = \left(T_0 + \max_{j\in V}\{(T_j - x_j)^+\} - x_0\right)^+ = \left(T_0 + \max_{j\in V}\{(T_j - x_j + x_i)^+\} - x_0 - x_i\right)^+ \tag{2.10}$$

and $\mathbf{x}'$ can be constructed as follows:

$$\begin{aligned} x_i' &= 0, \\ x_j' &= x_j - x_i, \quad j \in V, j \neq i, \end{aligned}$$

$$x'_0 \quad = \quad x_0 + x_i.$$

From (2.10) it follows that $L'_0 = L_0$ and thus $C(\mathbf{x}') = C(\mathbf{x}) + (h_0 + h_i^c)x_i < C(\mathbf{x})$. Note that $x'_j > x_j$ for all $j \in V$. So we can successively apply this construction to the negative subassembly planned leadtimes (as long as there are any), resulting in a vector $\mathbf{x}$ with lower costs than the original one and $x_i \geq 0$ for all $i \in V$. Finally, if $x_0 < 0$, we can write L_0 as

$$L_0 = \left(T_0 + \max_{i \in V}\{L_i\} - x_0 \right)^+ = T_0 + \max_{i \in V}\{L_i\} - x_0$$

and construct a vector $\mathbf{x}'$ as follows:

$$\begin{aligned} x'_0 &= 0, \\ x'_i &= x_i, \quad i \in V. \end{aligned}$$

The lateness L'_0 for $\mathbf{x}'$ is equal to $L_0 + x_0$, so $C(\mathbf{x}') = C(\mathbf{x}) + px_0 < C(\mathbf{x})$.

Proof of Lemma 2.2

For a subassembly activity $i \in V$, (2.7) can be written as:

$$P(\theta_i) = P(T_i - x_i > 0, T_i - x_i > \max_{j \in V \setminus \{i\}}\{T_j - x_j\}, T_i - x_i + T_0 - x_0 > 0). \qquad (2.11)$$

If x_i increases, each of the three events in the right-hand side decreases and thus $P(\theta_i)$ decreases as well. Only the third event depends on x_0. This event decreases if x_0 increases. The second event is the only one depending on x_j and it increases if x_j increases. Hence, we can conclude that $P(\theta_i)$ decreases in x_i and x_0 and increases in any x_j, $j \in V \setminus \{i\}$.

Proof of Lemma 2.3

For the final assembly activity, we have:

$$P(\theta_0) = P(\max_{i \in V}\{T_i - x_i\} \leq 0, T_0 - x_0 > 0).$$

The first event in the right-hand side increases in each x_i and the second event does not depend on x_i. Therefore $P(\theta_0)$ increases in each x_i. Only the second event depends on x_0 and it decreases if x_0 increases.

Proof of Theorem 2.3

Theorem 2.1 and Proposition 2.1 imply that there exists a non-negative solution to the Newsvendor equations of Theorem 2.1. To prove uniqueness we consider two non-negative vectors $\mathbf{y}$ and $\mathbf{z}$ with $y_0 \leq z_0$ and show that the blame probabilities corresponding to $\mathbf{y}$ and $\mathbf{z}$ are not identical unless $\mathbf{y} = \mathbf{z}$. We define the set U as

$$U = \{i : i \in V, z_i > y_i\}.$$

For vector $\mathbf{y}$, the probability that a activity in U is blamed is equal to (cf. (2.11))

$$\sum_{i \in U} P(\theta_i) = P\left(\max_{i \in U}\{T_i - y_i\} > 0, \right.$$
$$\left. \max_{i \in U}\{T_i - y_i\} > \max_{j \in V \setminus U}\{T_j - y_j\}, \max_{i \in U}\{T_i - x_i\} + T_0 - y_0 > 0\right). \quad (2.12)$$

Note that (i) if y_i for $i \in U$ increases, then all three events in the right-hand side decrease, (ii) if y_j for $j \in V \setminus U$ decreases, then only the second event decreases, and (iii) if y_0 increases, only the third event decreases. Hence, if U is not empty, blame probability (2.12) decreases if $\mathbf{y}$ is replaced by $\mathbf{z}$. If U is empty, we consider

$$P(\theta_0) = P(\max_{j \in V}\{T_j - y_j\} \leq 0, T_0 - y_0 > 0).$$

By Lemma 2.3 we conclude that $P(\theta_0)$ decreases if $\mathbf{y}$ is replaced by $\mathbf{z}$.

Proof of Proposition 2.2

Since $h_0 > 0$ and $L_0 \geq 0$ it follows that for any $\mathbf{x}$

$$C(\mathbf{x}) = C^a(\mathbf{x}) + h_0 \mathbb{E}[L_0] \geq C^a(\mathbf{x}) \geq C^a(\mathbf{x}^{\mathbf{a}*}).$$

In particular this inequality is valid for $\mathbf{x} = \mathbf{x}^*$.

Proof of Proposition 2.3

For cost function $C^a(\cdot)$ the optimal planned leadtime x_0^{a*} satisfies the Newsvendor equation

$$P(T_0 > x_0^{a*}) = \frac{h_0}{h_0^c + p} \tag{2.13}$$

and for cost function $C(\cdot)$ the optimal planned leadtime x_0^* satisfies

$$P(\theta_0) = P(T_0 > x_0^*, \max_{i \in V}\{L_i\} = 0) = P(T_0 > x_0^*)P(\max_{i \in V}\{L_i\} = 0) = \frac{h_0}{h_0^c + p}.$$

Since $P(\max_{i \in V}\{L_i\} = 0) < 1$, it follows that $x_0^* < x_0^{a*}$. Similar as in the proof of Theorem 2.3, we define the set U as

$$U = \{i : i \in V, x_i^{a*} \geq x_i^*\}.$$

If U is not empty, then it holds for x^* that the blame probability of an activity in U is equal to

$$\begin{aligned}
\sum_{i \in U} \frac{h_i}{h_0^c + p} &= P\left(\max_{i \in U}\{T_i - x_i^*\} > \max_{j \in V \setminus U}\{(T_j - x_j^*)^+\},\right. \\
&\quad \left.\max_{i \in U}\{T_i - x_i^*\} + T_0 - x_0^* > 0\right) \\
&\geq P\left(\max_{i \in U}\{T_i - x_i^{a*}\} > \max_{j \in V \setminus U}\{(T_j - x_j^{a*})^+\},\right. \\
&\quad \left.\max_{i \in U}\{T_i - x_i^{a*}\} + T_0 - x_0^* > 0\right) \\
&> P\left(\max_{i \in U}\{T_i - x_i^{a*}\} > \max_{j \in V \setminus U}\{(T_j - x_j^{a*})^+\}, T_0 < x_0^{a*},\right. \\
&\quad \left.\max_{i \in U}\{T_i - x_i^{a*}\} + T_0 - x_0^* > 0\right).
\end{aligned}$$

The last sum at the right-hand is the blame probability of an activity in U for x^{a*}. It should not be less but equal to the left-hand side. Hence, we conclude that U is empty and thus $x_i^{a*} < x_i^*$ for all $i \in V$. This completes the proof of part (a). Part b

is also proved by contradiction. Let us assume that

$$x_i^* + x_0^* < x_i^{a*} + x_0^{a*}, \quad i \in V. \tag{2.14}$$

The probability of a positive lateness L_0^* under $\mathbf{x}^*$ is equal to the Newsvendor fractile,

$$\begin{aligned}
\frac{h_0^c}{h_0^c + p} \;=\; P(L_0^* > 0) \;&=\; P(\max_{i \in V}\{(T_i - x_i^*)^+\} + T_0 - x_0^* > 0) \\
&=\; P(\max_{i \in V}\{(T_i - x_i^{a*} - x_i^* + x_i^{a*})^+\} + T_0 - x_0^* > 0) \\
&\geq\; P(\max_{i \in V}\{(T_i - x_i^{a*})^+ - x_i^* + x_i^{a*})\} + T_0 - x_0^* > 0) \\
&\geq\; P(\max_{i \in V}\{(T_i - x_i^{a*})^+\} - \max_{i \in V}\{x_i^* - x_i^{a*}\} + T_0 - x_0^* > 0) \\
&>\; P(\max_{i \in V}\{(T_i - x_i^{a*})^+\} - x_0^{a*} + x_0^* + T_0 - x_0^* > 0) \\
&=\; P(\max_{i \in V}\{(T_i - x_i^{a*})^+\} + T_0 - x_0^{a*} > 0) \;=\; P(L_0^{a*} > 0),
\end{aligned}$$

where we used $(b-a)^+ \geq b^+ - a$ for numbers $a > 0$ and b in the first inequality and assumption (2.14) in the third. This is a contradiction, since $P(L_0^{a*} > 0)$ should also be equal to the Newsvendor fractile at the left-hand side. Hence, (2.14) is not valid, so there is an activity i for which $x_i^* + x_0^* \geq x_i^{a*} + x_0^{a*}$.

Proof of Proposition 2.4

To derive the limiting behavior of x_0^{a*}, we rewrite (2.13) as:

$$P(T_0 \leq x_0^{a*}) = 1 - \frac{h_0}{h_0^c + p} = \alpha + \frac{h_1}{h_0^c + p},$$

so

$$x_0^{a*} = F_0^{-1}\left(\alpha + \frac{h_1}{h_0^c + p}\right).$$

Since $\frac{h_1}{h_0^c + p} \to 0$ as $h_0 \to \infty$, it follows from the continuity of $F_0^{-1}(\cdot)$ that

$$\lim_{h_0 \to \infty}\{x_0^{a*}\} = F_0^{-1}(\alpha).$$

The planned leadtime x_1^{a*} satisfies the Newsvendor equation

$$P(T_1 > x_1^{a*}, T_0 < x_0^{a*}, T_1 + T_0 > x_1^{a*} + x_0^{a*}) = \frac{h_1}{h_0^c + p}.$$

The right-hand side converges to 0 as $h_0 \to \infty$. Hence the blame probability at the left-hand side also converges to 0. Since x_0^{a*} converges to a proper limit, this implies that x_1^{a*} has to grow to infinity. From Proposition 2.3 it follows that $x_1^* > x_1^{a*}$, so also x_1^* tends to infinity. Finally, x_0^* satisfies (2.6), which can be rewritten to

$$P(T_1 \leq x_1^*)P(T_0 > x_0^*) = \frac{h_0}{h_0^c + p} = 1 - \alpha - \frac{h_1}{h_0^c + p}.$$

Since $x_1^* \to \infty$, and thus $P(T_1 \leq x_1^*) \to 1$ as $h_0 \to \infty$, it follows from the continuity of $F_0^{-1}(\cdot)$ that

$$\lim_{h_0 \to \infty} \{x_0^{a*}\} = F_0^{-1}(\alpha).$$

Proof of Proposition 2.5

To prove the first equation of Proposition 2.5, we start from (2.6). For activity 1, this equation can be written as

$$P(T_0 + T_1 > x_1^* + x_0^*, T_1 > x_1^*) = \frac{h_1}{h_0^c + p}$$
$$P\left(T_0 + T_1' > x_0^* | T_1 > x_1^*\right) P(T_1 > x_1^*) = \frac{h_1}{h_0^c + p}.$$

In the second expression, T_1' denotes the leadtime of activity 1, given that it exceeds x_1^*. Due to the memoryless property, this random variable is again exponentially distributed with parameter λ. The sum of 2 independent exponential variables with rate λ is an Erlang-2 distribution and thus we find the following expressions hold

$$P\left(T_0 + T_1' > x_0^* | T_1 > x_1^*\right) = (1 + \lambda x_0^*)e^{-\lambda x_0^*}$$
$$P(T_1 > x_1^*) = e^{-\lambda x_1^*}.$$

Hence, we obtain

$$P\left(T_0 + T_1 > x_1^* + x_0^*, T_1 > x_1^*\right) = (1 + \lambda x_0^*)e^{-\lambda(x_0^* + x_1^*)},$$

which leads to

$$(1 + \lambda x_0^*)e^{-\lambda(x_0^* + x_1^*)} = \frac{h_1}{h_0^c + p}.$$

To prove the second equation of Proposition 2.5 we start from (2.6). For activity 0, this equation can be written as

$$P(\theta_0) = P(T_1 < x_1^*)P(T_0 > x_0^*) = \frac{h_0}{h_0^c + p}.$$

Assuming exponential distributions leads to

$$(1 - e^{-\lambda x_1^*})e^{-\lambda x_0^*} = e^{-\lambda x_0^*} - e^{-\lambda(x_1^* + x_0^*)} = \frac{h_0}{h_0^c + p}.$$

Proof of Proposition 2.6

In Atan et al. (2016), it is proven that

$$P(T_1 > x_1^{a*}, T_0 < x_0^{a*}, T_1 + T_0 > x_1^{a*} + x_0^{a*}) = \frac{h_1}{h_0^c + p}.$$

Assuming identically exponentially distributed leadtimes, this equation can be rewritten as

$$\begin{aligned}
&\int_{x_1^{a*}}^{\infty} \int_{(x_1^{a*} + x_0^{a*} - t_1)^+}^{x_0^{a*}} \lambda e^{-\lambda t_1} \lambda e^{-\lambda t_0} dt_0 dt_1 \\
&\qquad = \int_{x_1^{a*}}^{x_1^{a*} + x_0^{a*}} \int_{x_1^{a*} + x_0^{a*} - t_1}^{x_0^{a*}} \lambda^2 e^{-\lambda(t_1 + t_0)} dt_0 dt_1 + \int_{x_1^{a*} + x_0^{a*}}^{\infty} \int_{0}^{x_0^{a*}} \lambda^2 e^{-\lambda(t_1 + t_0)} dt_0 dt_1. \quad (2.15)
\end{aligned}$$

The first term of the right-hand side is computed as follows:

$$\int_{x_1^{a*}}^{x_1^{a*}+x_0^{a*}} \int_{x_1^{a*}+x_0^{a*}-t_1}^{x_0^{a*}} \lambda^2 e^{-\lambda(t_1+t_0)} dt_1 dt_0 = \lambda \int_{x_1^{a*}}^{x_1^{a*}+x_0^{a*}} \left[-e^{-\lambda(t_1+t_0)}\right]_{x_1^{a*}+x_0^{a*}-t_1}^{x_0^{a*}} dt_1$$

$$= \lambda \int_{x_1^{a*}}^{x_1^{a*}+x_0^{a*}} e^{-\lambda(x_1^{a*}+x_0^{a*})} dt_1 - \lambda \int_{x_1^{a*}}^{x_1^{a*}+x_0^{a*}} e^{-\lambda(t_1+x_0^{a*})} dt_1$$

$$= \lambda x_0^{a*} e^{-\lambda(x_1^{a*}+x_0^{a*})} + e^{-\lambda(x_1^{a*}+x_0^{a*}+x_0^{a*})} - e^{-\lambda(x_1^{a*}+x_0^{a*})}.$$

The second term of the right-hand side of (2.15) can be rewritten as

$$\int_{x_1^{a*}+x_0^{a*}}^{\infty} \int_0^{x_0^{a*}} \lambda^2 e^{-\lambda(t_1+t_0)} dt_0 dt_1 = P(T_1 > x_1^{a*} + x_0^{a*}) P(T_0 < x_0^{a*})$$

$$= e^{-\lambda(x_1^{a*}+x_0^{a*})}(1 - e^{-\lambda x_0^{a*}})$$

$$= e^{-\lambda(x_1^{a*}+x_0^{a*})} - e^{-\lambda(x_1^{a*}+x_0^{a*}+x_0^{a*})}.$$

Finally, inserting these results into (2.15) leads to $\lambda x_0^{a*} e^{-\lambda(x_1^{a*}+x_0^{a*})} = \frac{h_1}{h_0^c+p}$.

In Atan et al. (2016) it is also proven that

$$P(T_0 > x_0^{a*}) = \frac{h_0}{h_0^c + p}.$$

By inserting the inverse cumulative probability for an exponential distribution, which is $e^{-\lambda x_0^{a*}}$ it directly follows that

$$e^{-\lambda x_0^{a*}} = \frac{h_0}{h_0^c + p}.$$

Proof of Proposition 2.7

For this proof we use the optimality equations for activity 1 from Propositions 2.5 and 2.6 and rewrite the terms:

$$(1 + \lambda x_0^*) e^{-\lambda(x_0^*+x_1^*)} = \lambda x_0^{a*} e^{-\lambda(x_1^{a*}+x_0^{a*})}$$

$$e^{-\lambda(x_0^*+x_1^*-x_1^{a*}-x_0^{a*})} = \frac{\lambda x_0^{a*}}{1 + \lambda x_0^*}$$

$$x_1^* - x_1^{a*} = -\frac{1}{\lambda} \ln\left(\frac{\lambda x_0^{a*}}{1+\lambda x_0^*}\right) + x_0^{a*} - x_0^*$$

From Proposition 2.4 it follows that $\lim_{h_0 \to \infty} \{x_0^* - x_0^{a*}\} = 0$ and thus we find

$$\lim_{h_0 \to \infty} \{x_1^* - x_1^{a*}\} = -\frac{1}{\lambda} \ln\left(\frac{\lambda x_0^*}{1+\lambda x_0^*}\right).$$

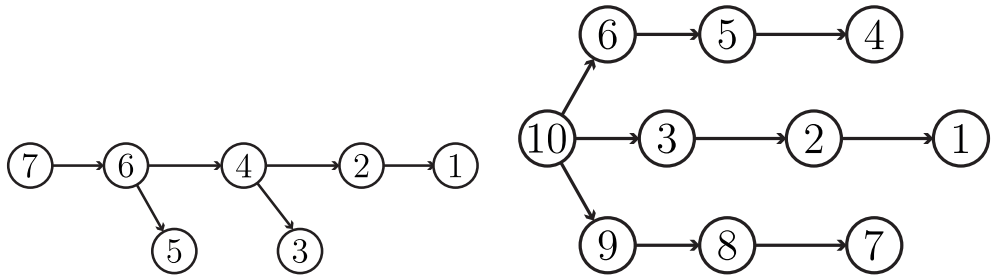
3

Modeling of Networks Under 'Pay as Planned' Costing Scheme

3.1. Introduction

We consider production systems that manufacture high-value, low-volume, customer-specific products, such as airplanes and lithography systems. The manufacturing process of such products is divided into multiple activities, which can only start when preceding activities are finished. High product complexity implies variability in activity leadtimes. In turn, random activity leadtimes imply randomness in the end product completion times and randomness in the customer delivery date. Several studies considered these production systems with the objective of determining the planned leadtimes of the activities. They addressed network structures such as two nodes (Yano, 1987a), serial networks (Elhafsi, 2002) and converging networks (Axsäter, 2005; Trietsch, 2006; Atan et al., 2016; Jansen et al., 2019). A common assumption in all these studies is that the network has a single final node, i.e., there is only one end product.

In this chapter, we aim to generalize the results for the 'pay as planned' cost structure obtained in Chapter 2 to networks that have multiple end nodes. This generalization is motivated by practice. For capital-intensive expensive products,



(a) A project with intermediate deadlines *(b)* Aligning one task of multiple projects

Figure 3.1: Examples of diverging networks

customers pay in several installments, where each installment is linked to the completion of an activity. For example, when building a house, the first installment is often paid when the foundation is finished. An example of a network representation for such a situation is depicted in Figure 3.1a. In this example, the customer pays installments after completion of activities represented by nodes 5, 3 and 1.

It is common practice that products for different customers are produced simultaneously. This is another motivation for our study. In fact, it can be beneficial to align identical activities for different products. For example, if the hull of a new vessel is built in Asia and the rest of the activities are performed in Europe, it makes sense to transport multiple hulls on a big vessel from Asia to Europe. Transporting an extra hull on the same vessel only marginally increases the total cost, while it can be very expensive and inefficient to transport each hull individually. An example of a network representation for such a system is depicted in Figure 3.1b. Node 10 is the shared transportation activity. After this activity, each project is completed separately.

In previous studies, different cost structures have been considered, but they all rely on the same basic principle: each activity in the network adds value to the final product. This value is represented by the activity holding costs. If the production of an end product is finished earlier than planned, a holding cost is incurred until the due date. If the product is completed later than planned, an extra penalty cost needs to be paid to the customer. We extend this cost concept to networks with multiple end nodes. Since an activity can add value to multiple end products, we define holding cost for each of the end nodes relevant to the activity. Similar to

networks with a single end node, there is a penalty cost for late completions of each end node. Our objective is to set the start times of all activities so that the total expected holding and penalty costs are minimized.

In Chapter 2 we determined a *planned leadtime* for all nodes in the system. However, planned leadtime solutions are ambiguous in networks with multiple end nodes. Therefore, the perspective in this chapter is different: we describe the solution in terms of a *planned start time* for each node and a *planned finish time* for each end node. This formulation easily applies to networks with single or multiple end nodes.

We also introduce the concept of tardy paths. For a specific realization of activity leadtimes, a path is tardy if it leads to an end node that is behind schedule. This is different from the concept of critical paths introduced by Kelley Jr and Walker (1959). While a critical path is solely based on planned leadtimes, a tardy path depends on the whole plan, i.e. start and finish times, and also on the realization of the random leadtimes.

For an assembly system with a single end node, we showed in Chapter 2 that under the optimal solution the probability of an activity delaying final delivery is proportional to the value it adds to the end product. In this chapter we extend this result. For given start and finish times and leadtime distributions, we derive an expression for the probability of a path being tardy under the optimal solution. We show that Newsvendor equations hold for specific sets of nodes. The remainder of this chapter is organized as follows. We formulate the model in Section 3.2. In Section 3.3, we introduce the concept of tardy paths. Structural results are presented in Section 3.4. We provide a numerical example in Section 3.5 and conclude in Section 3.6.

3.2. Model Formulation

Let $G = (\mathcal{V}, \mathcal{E})$ be a directed acyclic graph with N nodes. $\mathcal{V} = \{1, 2, ..., N\}$ is the set of all nodes and $\mathcal{E}$ is the set of all directed edges. Each node represents an activity and each edge represents a precedence relation. The edge from node i to node j is denoted by $\langle i, j \rangle$. We define $\mathcal{P}(i) := \{j \in \mathcal{V} : \langle j, i \rangle \in \mathcal{E}\}$ and $\mathcal{S}(i) := \{j \in \mathcal{V} : \langle i, j \rangle \in \mathcal{E}\}$ as the sets of immediate predecessors and immediate successors of node i, respectively. In addition, $\mathcal{R} := \{i \in \mathcal{V} : \mathcal{P}(i) = \emptyset\}$ and $\mathcal{L} := \{i \in \mathcal{V} :$

$\mathcal{S}(i) = \emptyset\}$ are the sets of root and leaf nodes, respectively. The set of all nodes that can reach node i is denoted by $\mathcal{Y}(i)$. This set can be recursively defined as $\mathcal{Y}(i) := \mathcal{P}(i) \cup \left(\bigcup_{j\in\mathcal{P}(i)} \mathcal{Y}(j)\right)$, where $\mathcal{Y}(i) = \emptyset$ for all $i \in \mathcal{R}$. The set of all nodes that are reachable from node i is $\mathcal{Z}(i)$, which can be also be recursively defined as $\mathcal{Z}(i) := \mathcal{S}(i) \cup \left(\bigcup_{j\in\mathcal{S}(i)} \mathcal{Z}(j)\right)$, where $\mathcal{Z}(i) = \emptyset$ for all $i \in \mathcal{L}$. We assume that nodes are numbered such that for any two nodes, if $i \in \mathcal{Y}(j)$ then $i > j$. Finally, we define $\mathcal{W}(i, j)$ as the set of all paths from node $i \in \mathcal{V}$ to $j \in \mathcal{L}$. Note that $\mathcal{W}(i, j)$ might be empty.

Activities in the network have uncertain durations. The duration of the activity in node i is modeled by a random variable T_i that is continuous and exists on the entire domain $(0, \infty)$. Random variables can be dependent, however, the joint probablity distribution of all random leadtimes should be continuous and it should exist on the domain $\mathbb{R}_+^N$. The planned start time of node i is t_i^s. An activity can only start after all its predecessors are finished. If all predecessors of i finish earlier than t_i^s, node i starts at the planned start time. Due to the uncertainty in duration of predecessors, the actual start times of non-root nodes are uncertain. The random variable A_i represents this actual start time and is defined as follows:

$$A_i \quad = \quad \max\{\max_{j\in\mathcal{P}(i)} \{A_j + T_j\}, t_i^s\}, \quad i \in \mathcal{V}.$$

For any realization of the activity leadtimes, A_i can be recursively computed for all nodes, starting from the root node with the highest index. Each leaf node j also has a planned finish time t_j^f. This is the due date communicated to the customer. We define the actual finish time for each leaf node as:

$$B_j \quad = \quad \max\{t_j^f, A_j + T_j\}, \quad j \in \mathcal{L}.$$

Note that if a product is completed before its planned finish time, it is delivered to the customer at its planned finish time and not earlier. Clearly, the random variables A_i and B_i depend on the vectors of planned start and finish times $\mathbf{t^s}$ and $\mathbf{t^f}$, respectively. We do not explicitly indicate this dependence in the notation, except when it enhances readability.

In this chapter, we use the 'pay as planned' cost structure. The motivation behind this cost structure is that, when an activity is started, investments in resources need to be made. These investments are earned back when the product is delivered to the

customer, i.e. after completion of a leaf node. During this period, holding costs are incurred, representing the interest paid on the investment. These costs are incurred regardless of whether this activity is actually started, it could also be delayed due to preceding activities.

We define $h_{i,j}$ as the holding cost per time unit paid from the planned start time t_i^s of node i until the completion of leaf node j. We have $h_{i,j} > 0$ when $j \in \mathcal{L} \cap (\mathcal{Z}(i) \cup \{i\})$ and $h_{i,j} = 0$ otherwise. The unit echelon holding cost for node i is calculated as $h_i^e = \sum_{j \in \mathcal{L} \cap (\mathcal{Z}(i) \cup \{i\})} h_{i,j}$. The unit local holding cost for leaf node j is defined as $h_j^c = \sum_{i \in \mathcal{Y}(j) \cup \{j\}} h_{i,j}$. If leaf node j is completed later than its planned finish time t_j^f, a penalty cost p_j is incurred per unit time late. This leads to the following expression for the expected cost of a production plan defined by the vectors of planned start times $\mathbf{t^s}$ and finish times $\mathbf{t^f}$ under the 'pay as planned' costing scheme.

$$C(\mathbf{t^s}, \mathbf{t^f}) = \sum_{j \in \mathcal{L}} \left(\sum_{i \in \mathcal{Y}(j) \cup \{j\}} (t_j^f - t_i^s) h_{i,j} + (h_j^c + p_j) \mathbb{E}[B_j - t_j^f] \right) \tag{3.1}$$

Since holding costs are incurred from the planned start time, there is no uncertainty in holding cost to be paid until the due date. The only uncertainty is in the actual completion times of the leaf nodes, which in turn depend on the completion times of all predecessor nodes. Our objective is to solve the optimization problem (P) defined as $\min_{\mathbf{t^s}, \mathbf{t^f}} \{C(\mathbf{t^s}, \mathbf{t^f})\}$.

3.3. Tardy Paths

Randomness of the activity leadtimes implies that the actual start and finish times might differ from the planned start and finish times. A realization of the leadtimes $T_1, \ldots, T_N$ is indicated by $\omega = (t_1, \ldots, t_N) \in \Omega = \mathbb{R}_+^N$. So $T_i(\omega) = t_i$ denotes a realization of T_i. For each realization of leadtimes, we can identify *tardy paths*. For a realization $\omega \in \Omega$, we say that a path from node i to leaf node j is tardy if there is no waiting time, or 'slack' on that path. The formal definition is as follows:

Definition 3.1 For a realization $\omega \in \Omega$, a path from node i to leaf node j is tardy iff

1. Node i starts at the planned start time: $A_i(\omega) = t_i^s$.

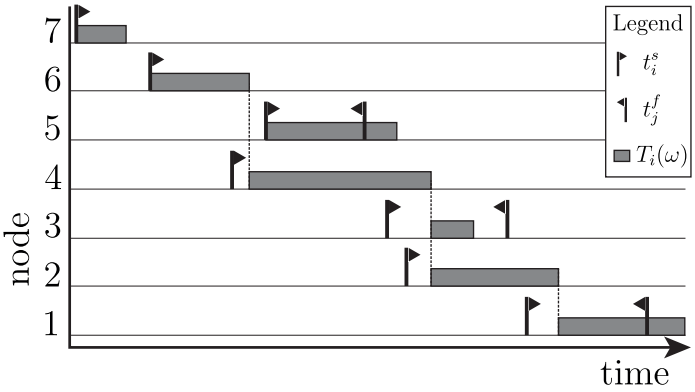


Figure 3.2: Production plan $(\mathbf{t^s}, \mathbf{t^f})$ and a realization ω for the network in Figure 3.1a

2. For each edge $\langle k, l \rangle$ on that path, the actual finish time of node k equals the actual start time of node l: $A_k(\omega) + T_k(\omega) = A_l(\omega)$.

3. Leaf node j finishes later than planned: $B_j(\omega) > t_j^f$.

The definition implies that there exists a tardy path to leaf node j only if node j is late, and if so, it is unique with probability 1, i.e. there is exactly one tardy path to node j. To find this path, one starts from node j following the path with no slack, until a node is found that starts on time. Since there is a tardy path for each leaf node that is late, networks with multiple leaf nodes can have multiple tardy paths.

To explain the idea of tardy paths further, we consider the networks in Figure 3.1. For both networks, we show the production plan obtained from a candidate solution $(\mathbf{t^s}, \mathbf{t^f})$ to the optimization problem (P) together with a realization ω of this plan. Each start time t_i^s is denoted by a 'flag' pointing to the right and each finish time t_i^f by a flag pointing to the left. The realization of each random variable $T_i(\omega)$ is denoted by a gray bar.

Figure 3.2 shows the production plan for the network in Figure 3.1a. In this network, nodes 1, 3 and 5 are leaf nodes and thus have planned finish times. Nodes 5, 6 and 7 start at their planned start times. All other nodes are delayed. Leaf nodes 1 and 5 are late. Although the actual start time of node 3 is later than planned, it makes up for this lateness and finishes in time. We have two leaf nodes that are late, thus we have two tardy paths. For node 5, the tardy path simply is $\{5\}$, since this node

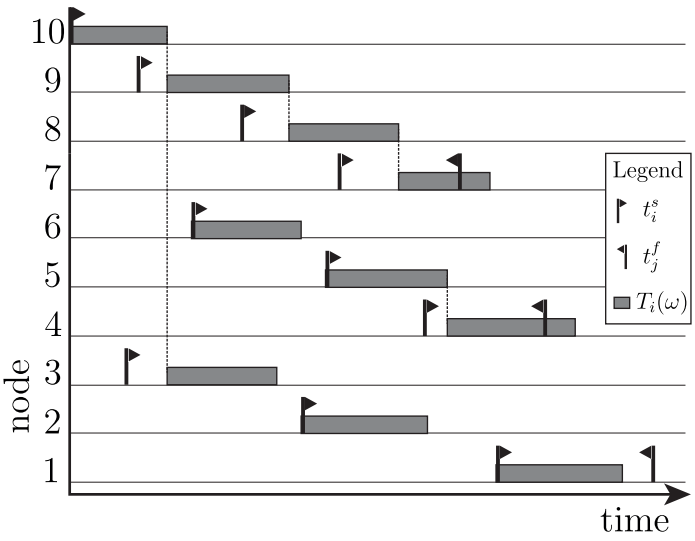


Figure 3.3: Production plan $(\mathbf{t^s}, \mathbf{t^f})$ and a realization ω for the network in Figure 3.1b

starts at its planned start time. The tardy path to node 1 is $\{6, 4, 2, 1\}$ because (i) node 6 starts at its planned start time, (ii) nodes 4, 2 and 1 start later than planned and (iii) node 1 finishes later than planned.

A production plan for the network in Figure 3.1b is depicted in Figure 3.3. Leaf node 1 is in time, while leaf nodes 4 and 7 are late. Critical paths for these nodes are $\{5, 4\}$ and $\{10, 9, 8, 7\}$, respectively.

Tardy paths are defined for realizations $\omega \in \Omega$. For each node i and leaf node j, we now define $\theta_{i,j}$ as the event that a path from i to j is tardy, i.e., the set of all realizations ω for which there is a tardy path from i to j.

3.4. Structural Results

In this section, we derive structural results for the optimization problem (P). We show that (P) is convex and derive optimality equations. The complicated terms in the cost function (3.1) are $\mathbb{E}[B_j(\mathbf{t^s}, \mathbf{t^f}) - t_j^f]$, where we now explicitly indicate that B_j is a function of $\mathbf{t^s}$ and $\mathbf{t^f}$. Below we show that the first order partial derivatives with respect to t_i^s and t_i^f can be expressed as (a sum of) probabilities $P(\theta_{i,j})$:

Lemma 3.1 *For all leaf nodes $j \in \mathcal{L}$,*

$$\frac{\partial}{\partial t_k^f}\mathbb{E}[B_j(\mathbf{t^s}, \mathbf{t^f}) - t_j^f] = \begin{cases} -\sum_{i\in\mathcal{Y}(j)\cup\{j\}} P(\theta_{i,j}), & \text{if } k = j \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

$$\frac{\partial}{\partial t_i^s}\mathbb{E}[B_j(\mathbf{t^s}, \mathbf{t^f})] - t_j^f] = \begin{cases} P(\theta_{i,j}), & \text{if } i \in \mathcal{Y}(j)\cup\{j\} \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

Proof. First we show that

$$\frac{\partial\mathbb{E}[B_j(\mathbf{t^s}, \mathbf{t^f}) - t_j^f]}{\partial t_i^s} = \frac{\partial\mathbb{E}[B_j(\mathbf{t^s}, \mathbf{t^f})]}{\partial t_i^s} = \mathbb{E}\left[\frac{\partial B_j(\mathbf{t^s}, \mathbf{t^f})}{\partial t_i^s}\right]$$

or equivalently, that for every sequence $g_1, g_2, \ldots$ converging to 0,

$$\lim_{n\to\infty}\frac{\mathbb{E}[B_j(\mathbf{t^s} + g_n\mathbf{e_i}, \mathbf{t^f})] - \mathbb{E}[B_j(\mathbf{t^s}, \mathbf{t^f})]}{g_n} = \mathbb{E}\left[\frac{\partial B_j(\mathbf{t^s}, \mathbf{t^f})}{\partial t_i^s}\right] \tag{3.4}$$

where $\mathbf{e_i}$ is the unit vector with 1 at position i. To find the derivative, we rewrite $B_j(\mathbf{t^s}, \mathbf{t^f})$ as follows. For every realization $\omega = (t_1, \ldots, t_N)$ we have

$$B_j(\mathbf{t^s}, \mathbf{t^f})(\omega) = \max\left\{t_j^f, \max_{\substack{k\in\mathcal{Y}(j)\cup\{j\},\\ w\in\mathcal{W}(k,j)}}\left\{t_k^s + \sum_{l\in w} T_l(\omega)\right\}\right\} \tag{3.5}$$

Hence the derivative $\frac{\partial B_j(\mathbf{t^s}, \mathbf{t^f})(\omega)}{\partial t_i^s}$ exists for almost all ω and

$$\frac{\partial B_j(\mathbf{t^s}, \mathbf{t^f})(\omega)}{\partial t_i^s} = \begin{cases} 1, & \text{if } B_j(\mathbf{t^s}, \mathbf{t^f})(\omega) = t_i^s + \sum_{l\in w} T_l(\omega), \\ 0, & \text{otherwise.} \end{cases}$$

For all ω,

$$\left|\frac{B_j(\mathbf{t^s} + g_n\mathbf{e_i}, \mathbf{t^f})(\omega) - B_j(\mathbf{t^s}, \mathbf{t^f})(\omega)}{g_n}\right| \leq 1,$$

so by bounded convergence we can conclude that (3.4) holds. Since the derivative

with respect to t_i^s is 1 if a path from i to j is tardy and 0 otherwise, we get

$$\frac{\partial \mathbb{E}[B_j(\mathbf{t^s}, \mathbf{t^f}) - t_j^f]}{\partial t_i^s} = \mathbb{E}\left[\frac{\partial B_j(\mathbf{t^s}, \mathbf{t^f})}{\partial t_i^s}\right] = P(\theta_{i,j}),$$

which proves (3.3). The proof of (3.2) proceeds along the same lines. □

Observe that the derivative is only a function of planned start and finish times of nodes that share the same leaf node.

Lemma 3.2 *The cost function $C(\mathbf{t^s}, \mathbf{t^f})$ is jointly convex in $(\mathbf{t^s}, \mathbf{t^f})$.*

Proof. Convexity of (3.1) immediately follows from convexity of (3.5) in $(\mathbf{t^s}, \mathbf{t^f})$ for every ω. □

This property guarantees that any local optimal production plan $(\mathbf{t^s}, \mathbf{t^f})$ is also a global optimum. For the optimal production plan in an assembly network with a single leaf node we showed in Chapter 2 that the probability that a tardy path starts in node $i \in \mathcal{V}$ is proportional to the value this activity adds to the final product (Theorem 2.1). We show that this result also holds for specific nodes in production networks with multiple leaf nodes, depending on their location in the network. Before stating the optimality equations for the production plan, we define $\mathcal{U}$ as the set of nodes, which have the property that they can reach exactly one leaf node: $\mathcal{U} = \{i : |\mathcal{L} \cap (\mathcal{Z}(i) \cup \{i\})| = 1\}$. According to this definition, $\mathcal{U} = \{1, 2, 3, 5\}$ and $\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ for the networks in Figures 3.1a and 3.1b, respectively. A more complicated network and its corresponding set $\mathcal{U}$ are shown in Figure 3.4. So far, the networks studied in literature have only one leaf node, for which it immediately follows that $\mathcal{U} = \mathcal{V}$. We denote the optimal solution to (P) as $(\mathbf{t^{s*}}, \mathbf{t^{f*}})$. Under the optimal solution the probability of a path being tardy is denoted by $P(\theta_{i,j}^*)$. The optimality equations for $(\mathbf{t^{s*}}, \mathbf{t^{f*}})$ are formulated in Theorem 3.1.

Theorem 3.1 *The optimal production plan $(\mathbf{t^{s*}}, \mathbf{t^{f*}})$ of (P) satisfies the following equations:*

$$\sum_{i \in \mathcal{Y}(j) \cup \{j\}} P(\theta_{i,j}^*) = \frac{h_j^c}{h_j^c + p_j}, \quad j \in \mathcal{L} \tag{3.6}$$

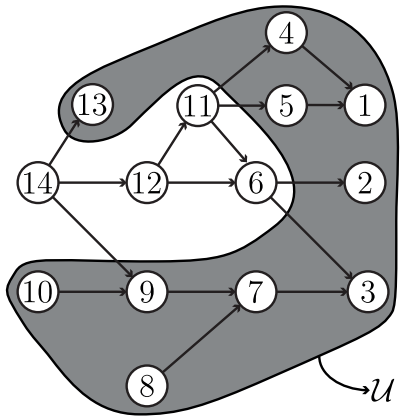


Figure 3.4: Example of set $\mathcal{U}$ for a network of 14 nodes

$$\sum_{j\in(\mathcal{Z}(i)\cup\{i\})\cap\mathcal{L})} (h_j^c + p_j)P(\theta_{i,j}^*) = h_i^e\,, \quad i \in \mathcal{V} \tag{3.7}$$

$$P(\theta_{i,j}^*) = \frac{h_{i,j}}{h_j^c + p_j}, \quad i \in \mathcal{U}, j \in \mathcal{L} \tag{3.8}$$

Proof. To prove (3.6) we take the partial derivative of (3.1) with respect to t_j^f. By Lemma 3.1,

$$\frac{\partial C(\mathbf{t^s}, \mathbf{t^f})}{\partial t_j^f} = -\sum_{i\in\mathcal{Y}(j)\cup\{j\}} h_{i,j} + (h_j^c + p_j) \sum_{i\in\mathcal{Y}(j)\cup\{j\}} P(\theta_{i,j}), \quad j \in \mathcal{L}$$

At optimality this derivative should vanish, so

$$\sum_{i\in\mathcal{Y}(j)\cup\{j\}} P(\theta_{i,j}^*) = \frac{h_j^c}{h_j^c + p_j}.$$

To prove (3.7) and (3.8) we take the derivative of (3.1) with respect to t_k^s, yielding

$$\frac{\partial C(\mathbf{t^s}, \mathbf{t^f})}{\partial t_k^s} = -h_k^e + \sum_{j\in\mathcal{Z}(k)\cap\mathcal{L}} (h_j^c + p_j)P(\theta_{k,j}), \quad k \in \mathcal{V}.$$

Since this derivative vanishes at optimality, we find

$$\sum_{j\in(\mathcal{Z}(k)\cup\{k\})\cap\mathcal{L}} (h_j^c + p_j)P(\theta_{k,j}^*) = h_k^e, \quad k \in \mathcal{V}.$$

If $k \in \mathcal{U}$, then k can reach only one leaf node. Hence the above equation reduces to

$$P(\theta_{k,j}^*) = \frac{h_k}{h_j^c + p_j}, \quad k \in \mathcal{U},$$

which concludes the proof. □

The interpretation of Theorem 3.1 is as follows: (3.6) states that the general Newsvendor equation holds for every leaf node. This equation depends only on the local holding cost h_j^c and penalty cost p_j of the corresponding leaf node and is independent from the local holding and penalty costs of other leaf nodes. The left-hand side of (3.7) is a summation over all leaf nodes reachable from i. Each term in the summation denotes the probability that a path from i to j is tardy, multiplied by the sum of holding and penalty costs at that leaf node. This summation equals the echelon holding cost h_i^e. If $i \in \mathcal{U}$ the summation of (3.7) reduces to only one term and thus the the equation can be simplified to (3.8) which is a closed form expression for the probability of a tardy path. This equation states that under the optimal solution, the probability that a path from i to j is tardy, is proportional to the value node i adds to leaf node j, i.e., $h_{i,j}$. A special case in which it is possible to derive a closed-form expression for $P(\theta_{i,j}^*)$ in case $i \notin \mathcal{U}$ is stated in the following lemma.

Lemma 3.3 *If for $i \notin \mathcal{U}$ and $j \in \mathcal{L}$, it holds that $\{i\} = \mathcal{Y}(j)\backslash\mathcal{U}$, then*

$$P(\theta_{i,j}^*) = \frac{h_{i,j}}{h_j^c + p_j} \tag{3.9}$$

Proof. In case $\{i\} = \mathcal{Y}(i)\backslash\mathcal{U}$, the tardy path probability can be written as

$$P(\theta_{i,j}) = \sum_{k\in\mathcal{Y}(j)} P(\theta_{k,j}) - \sum_{k\in\mathcal{Y}(j)\cap\mathcal{U}} P(\theta_{k,j})$$

Using (3.6) and (3.8) we find that

$$P(\theta^*_{i,j}) = \frac{h^c_j}{h^c_j + p} - \frac{\sum_{k \in \mathcal{U}} h_{k,j}}{h^c_j + p} = \frac{h_{i,j}}{h^c_j + p}$$

□

Lemma 3.3 is useful for the network shown in Figure 3.1b. For this network, nodes 1 to 9 belong to the set $\mathcal{U}$ and hence (3.8) holds for these nodes. Node 10 is not part of $\mathcal{U}$, but the conditions in Lemma 3.3 are satisfied and hence (3.9) holds for this node.

The optimality equations can only be solved for special network structures. In general, the solution is complicated, since the expected lateness at each end node depends on the leadtime distributions of all its parents. The optimality equations are also useful to validate solutions obtained via approximate methods as the resulting tardy path probabilities should satisfy these equations (see Section 3.5). Furthermore, the optimality equations can be used to derive structural properties of the optimal solution. This is demonstrated below.

A strictly converging network is a network in which each node has exactly one successor, except for the only leaf node, node 1. Hence $\mathcal{U} = \mathcal{V}$, $\mathcal{L} = \{1\}$, $\mathcal{Y}(1) \cup \{1\} = \mathcal{V}$ and thus (3.8) applies to each node in the network. The following monotonicity properties hold for the tardy path probabilities in strictly converging networks.

Lemma 3.4 *For each node $i \in \mathcal{V}$ in a strictly converging network, $P(\theta_{i,1})$ is*

1. *increasing in t^s_i,*
2. *decreasing in t^f_1,*
3. *decreasing in t^s_k, for all $k \in \mathcal{V} \backslash \{i\}$.*

Proof.

(a) Consider a realization $\omega \in \theta_{i,1}$. Then the path from i to 1 is tardy and it remains so by increasing t^s_i.

(b-c) Consider a realization $\omega \notin \theta_{i,1}$. Then the path from i to 1 is not tardy. It remains not tardy by increasing t_1^f or by increasing t_k^s for any $k \in \mathcal{V}\backslash\{i\}$.

□

We can also look at properties of combinations of the tardy path probabilities.

Lemma 3.5 *For any nonempty subset $\mathcal{Q} \subseteq \mathcal{V}$ in a strictly converging network, $\sum_{k\in\mathcal{Q}} P(\theta_{k,1})$ is increasing in each $t_i^s, i \in \mathcal{Q}$.*

Proof. Let $i \in \mathcal{Q}$. We need to show that the event $\bigcup_{k\in\mathcal{Q}} \theta_{k,1}$ increases as t_i^s increases. Consider a realization $\omega \in \bigcup_{k\in\mathcal{Q}} \theta_{k,1}$. If $\omega \in \theta_{i,1}$, then the path from i to 1 is tardy and it remains so by increasing t_i^s. If $\omega \in \theta_{k,1}$ with $k \in \mathcal{Q}\backslash\{i\}$, then the path from k to 1 is tardy. By increasing t_i^s, either this path stays tardy, or the path from i to 1 becomes tardy. In both cases, the realization stays in $\theta_{k,1} \cup \theta_{i,1}$. □

Using the above properties, we derive monotonicity properties of the optimal solution with respect to the penalty cost parameter p_1.

Lemma 3.6 *For each node $i \in \mathcal{V}$ in a strictly converging network, the planned start time t_i^{s*} of the optimal solution $\mathbf{t^{s*}}$ is decreasing in p_1.*

Proof.

Let $p\prime_1 > p_1$ and denote the corresponding optimal solutions by $\mathbf{t^{s\prime}}$ and $\mathbf{t^{s*}}$. Increasing p_1 decreases the Newsvendor fractile on the right hand side of (3.6) and (3.8). Hence, to satisfy (3.6) for $p\prime$, the probability $\sum_{i\in\mathcal{V}} P(\theta_{i,1})$ should reduce. From Lemma 3.5 it follows that this probability is increasing in any t_i^s. Hence $\mathbf{t^{s\prime}}$ contains at least one start time $t_i^{s\prime} < t_i^{s*}$. Let $\mathcal{Q}$ be the set of all nodes k for which $t_k^{s\prime} < t_k^{s*}$. We now show that $\mathcal{Q} = \mathcal{V}$. Lemmas 3.4(c) and 3.5 imply that the sum of tardy probabilities $\sum_{k\in\mathcal{V}\backslash\mathcal{Q}} P(\theta_{k,1})$ for $\mathbf{t^{s\prime}}$ is greater than the one for $\mathbf{t^{s*}}$. Hence, there is at least one node $k \in \mathcal{V}\backslash\mathcal{Q}$ for which $P(\theta_{k,1})$ for $\mathbf{t^{s\prime}}$ is greater than the one for $\mathbf{t^{s*}}$. But then it is impossible that for $\mathbf{t^{s\prime}}$, optimality equation (3.8) is satisfied for this node k. Hence, we conclude that $\mathcal{V}\backslash\mathcal{Q} = \emptyset$. □

An increase in the penalty cost leads to a higher optimal service level, i.e. more on-time completions. To achieve this level, each individual node is planned to start earlier. This clearly shows the trade-off between planned leadtimes and service level.

3.5. Numerical Example

In this section, we provide a numerical example for the network in Figure 3.4. As stated by Elhafsi (2002), an exact calculation of $C(\mathbf{t^s}, \mathbf{t^f})$ is computationally intensive for networks with many nodes, specifically if nodes are in series. This is due to multi-dimensional integrals that need to be computed to obtain the expected lateness of each leaf node.

To overcome this problem, we use samples of the stochastic leadtime distributions. We generate a (large) number of samples $\omega_1, \ldots, \omega_n$ and estimate $C(\mathbf{t^s}, \mathbf{t^f})$ by its sample average. Computing the cost for each sample is relatively easy, since it only involves maximum, summation and multiplication operations. After computing the cost for each sample we take the average to find the cost. Note that this sample average is also jointly convex (see the proof of Lemma 3.2).

The set-up of our experiment is as follows. For each activity we use a Gamma distribution with shape parameter $k = 3$ and scale parameter $\theta = 4$. For the holding costs we have $h_{i,j} = 1$ if $j \in \mathcal{Y}(i) \cup \{i\}$ and 0 otherwise. For the penalty costs, we have $p_j = 10$ for all $j \in \mathcal{L}$. The sample size for the leadtime distributions is set to $n = 10^6$. This means that for each activity we have 10^6 samples of its leadtime distribution. Without loss of generality we set $t_1^f = 0$. Since the problem is convex, we use a standard function available in Matlab called fmincon[1]. This is a function that finds the minimum of constrained nonlinear multivariable function using an interior point algorithm as described by Byrd et al. (2000).

The results are presented in Figure 3.5. It seems counter-intuitive that the start times of nodes with the same predecessor are not necessarily equal. For example, nodes 12 and 13 have the same predecessor (node 14), but have different start times. These nodes have different network locations and therefore the parameters in the optimality equations for their planned start times are different. In specific cases,

[1] https://nl.mathworks.com/help/optim/ug/fmincon.html

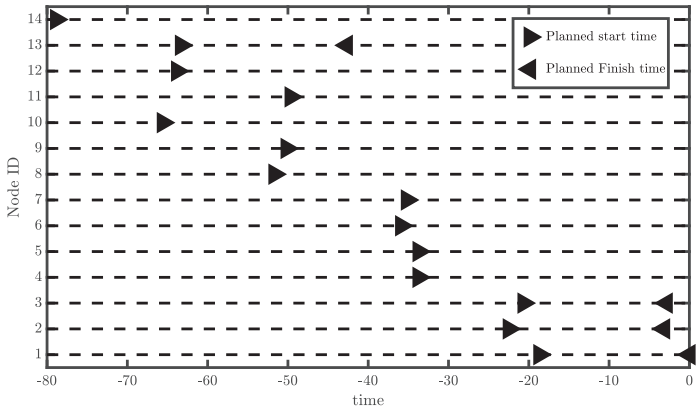


Figure 3.5: Production plan for the network in Figure 3.4.

start times are equal, for example for nodes 4 and 5, which is due to symmetry.

To validate the obtained solution we simulate the production plan using 10^6 new samples of the activity lead times. We compute the tardy path probabilities $P(\theta_{i,j})$ and verify whether these probabilities satisfy the derived optimality equations. The results are shown in Tables 3.1, 3.2 and 3.3. The tables show the numerical results for equations (3.6), (3.7) and (3.8) respectively. In each table, the first column denotes the right-hand side (RHS) of an optimality equation, which can be directly computed from the cost parameters. The second column denotes the simulated, left-hand side (LHS) of an equation. At optimality LHS and RHS should be equal, so the closer the LHS (simulated) is to the RHS (target value), the better the better the obtained solution is. The third column denotes the relative gap, defined as $\frac{LHS-RHS}{RHS} \cdot 100\%$.

Clearly, gaps are all within 3% and thus we conclude that the simulation results are sufficiently close to optimal. For equation (3.6) the gap is even smaller; it is at most 0.58%. For this example we use 10^6 samples, which allows us to obtain a solution within seconds. The solution can be improved by increasing the number of samples used in the simulation.

j	$\frac{h_j^c}{h_j^c+p}$	$\sum_{i\in\mathcal{Y}(j)\cup\{j\}} P(\theta_{i,j})$	Relative Gap (%)
1	0.3750	0.3772	0.58
2	0.3333	0.3337	0.12
3	0.4737	0.4753	0.34
13	0.1667	0.1655	-0.68

Table 3.1: Validation of (3.6) for nodes 1, 2, 3 and 13 of the network in Figure 3.4.

i	h_i^e	$\sum_{j\in(\mathcal{Z}(i)\cup\{i\})\cap\mathcal{L})}(h_j^c+p_j)P(\theta_{i,j}^*)$	Relative Gap (%)
6	2	2.0235	1.17
11	3	3.0537	1.79
12	3	2.9598	-1.34
14	4	3.9899	-0.25

Table 3.2: Validation of (3.7) for nodes 6, 11, 12 and 14 of the network in Figure 3.4.

3.6. Conclusion & Future Research Directions

In this chapter, we generalize the results derived in Chapter 2 for the problem of setting planned leadtimes in assembly and serial production systems under a 'pay as planned' costing structure. For a large class of networks, we prove optimality equations, which have a Newsvendor structure for specific networks. For these results, independence of leadtimes is not necessary, which increases the applicability of the 'pay as planned' cost function over the 'pay as realized' cost function.

The derivation of structural results for the 'pay as planned' cost function raises the question whether it is possible to derive structural results for the 'pay as realized' cost function. In the current literature only structural results are available for specific networks structures. The next chapter in this thesis addresses this question.

i, j	$\frac{h_i}{h_j^e + p_j}$	$P(\theta_{i,j})$	Relative Gap (%)
1,1	.0625	.0617	-1.23
2,2	.0667	.0667	0.09
3,3	.0526	.0531	0.87
4,1	.0625	.0643	2.80
5,1	.0625	.0637	1.97
7,3	.0526	.0519	-1.48
8,3	.0526	.0529	0.47
9,3	.0526	.0538	2.14
10,3	.0526	.0516	-2.04
13,13	.0833	.0829	-0.51
14,13	.0833	.0826	-0.86

Table 3.3: Validation of (3.8) for all nodes in $\mathcal{U}$ and (3.9) for node 14 of the network in Figure 3.4.

4

Modeling of Networks Under 'Pay as Realized' Costing Scheme

4.1. Introduction

Over the years, many papers addressed variations of the planned leadtime problem under the 'pay as realized'costing scheme. For special network structures and special types of leadtime distributions, optimality equations for planned leadtimes have been derived. For more complicated systems, researchers rely on simulations and heuristics to obtain solutions for the planned leadtime problem. Recently Atan et al. (2016) conjectured optimality equations that have a Newsvendor form for multistage assembly systems. The authors came up with a stochastic event distinction, where each realization of the leadtimes that results in a late delivery is assigned to a specific node. At optimality, the probability of this event should equal a Newsvendor fractile. The authors were able to prove the equations for a two stage assembly system.

In this chapter we generalize results of previous works that focus on specific network structures. We consider the class of strictly converging networks. A strictly converging network is a network where each activity has either no or exactly one successor. This allows us to model networks with multiple assembly points, a

structure that is very common in the production of complicated products. We model these networks by directed acyclic graphs, where each node represents an activity and each edge a precedence relation.

To describe the production plan, we borrow the concept of planned start and finish times of Chapter 3. These planned start and finish times describe the planned starting time (a time *point*). By using planned start times, the solution of the optimization problem directly shows when to start a specific activity. It simplifies notation for mathematical results and proofs in the paper and more importantly, it is essential for proving structural results for larger networks.

Because leadtimes are uncertain, realizations always deviate from the production plan. Some deadlines in the plan are met, others are exceeded. To study the impact of these deviations, we introduce the concept of tardy paths. Using planned start times and leadtime distributions, we define stochastic events which we refer to as tardy paths. A tardy path is a sequence of activities executed back to back, with no waiting time in between. All activities on this path finish behind schedule. This definition is similar to Definition 3.1 used in Chapter 2 for the 'pay as planned' cost function. The difference in this chapter is that we define tardy paths by only considering subsystems of the original subsystem. Via these subsystems, we derive an optimal production plan for the original system.

The optimal production plan is a production plan that minimizes the expected cost for deviations of the plan. The expected costs consist of holding costs and penalty costs, which increase linearly in time. Holding costs are charged per activity, from the moment the activity starts until the final product is delivered to the customer. Penalty costs are charged for late delivery of the final product. We prove that the optimal solution to this optimization problem satisfies a set of Newsvendor equations. In these equations we relate the probability of a specific tardy path to a Newsvendor fractile.

For specific network structures the equations are decoupled and can be solved recursively, leading to a unique solution. For other networks, it is complicated to solve the system of optimality equations exactly. However, the equations are extremely useful for the validation of a given solution. This gives us a tool to validate the performance of known heuristics in literature.

4.2. Problem Description

We consider the production planning problem of a single product. The product is ordered by a customer, who requests a certain configuration with a delivery due date. Examples of such products are airplanes, lithography machines, container vessels, etc. Orders for these products are placed in advance, since each customer might have specific requirements. For example, airplanes can differ in terms of painting, engine power and cabin interior. When an airline places an order at an airplane manufacturer, a due date is agreed upon. Late deliveries can be costly since the airline might have scheduled flights. These need to be cancelled and/or other costly alternative measures need to be taken. Early deliveries might not be possible due to transportation requirements. Even if it is possible, it is costly to keep these capital-intensive products idle. Therefore, it is crucial for the manufacturer to deliver the airplane exactly at the agreed due date.

Manufacturing processes of these products consist of many tasks. Multiple tasks can be executed in parallel, but the execution of some tasks can only start after completion of others. All required tasks and the precedence relations among them are known to the manufacturers.

The duration of each task is uncertain. Historical data and expert estimations can provide information about this uncertainty. Due to the uncertainty, on time completions of tasks can not be guaranteed. A guaranteed delivery would mean that the manufacturer plans for the worst case scenario, which is the longest possible duration for each task. This would result in an unacceptably long leadtime for the product.

The product manufacturer incurs penalty and holding costs. Penalty cost is paid if the product is delivered after the due date and is linear in the difference between delivery date and due date. Holding cost is paid if the product is completed earlier than the agreed delivery date. During this period of idle time, cost for stocking the product and cost for invested capital is incurred. In addition, holding costs are paid when there is idle time between two succeeding tasks. This can be due to other tasks that are not yet finished. Holding costs are linear in idle times.

The manufacturer aims to plan the production of the product such that the expected total cost is minimized. An idea could be to start with all tasks that have no

predecessors and proceed with other tasks as soon as all their predecessors are finished. This plan can result in long waiting times between tasks since tasks that have the same successor are not planned such that they finish at (roughly) the same time. In addition, when the due date is far in advance, the product will be completed very early. It is easy to verify that such a plan minimizes the throughput time of the product, but it is not guaranteed that such a solution also minimizes the expected total cost.

In this study, we develop a production plan that minimizes the expected total cost. This plan consists of a *planned start time* for each task. According to this plan, if all predecessors of a task are completed before the task's planned start time, the task starts at its planned start time. If there are predecessors that are not completed at the planned start of the task, it starts as soon as the latest predecessor finishes. The finish time of the task is solely determined by its actual start time and its stochastic leadtime. Note that the actual start time is a result of the planned start times and leadtime realizations of all predecessors.

In the rest of this section, we initially define the production network, the task leadtimes and the production plan.

4.2.1 Network Formulation

We represent the production process as a graph $G = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. The graph is directed, acyclic and strictly converging. There are $N \geq 1$ nodes in $\mathcal{V}$, numbered as $i = 1, ..., N$. For the final node, we have $i = 1$. To ease notation, we introduce an artificial node 0 to denote the completion of the project, but this node is not part of $\mathcal{V}$. The set $\mathcal{P}(i)$ contains the direct predecessors of node $i \in \mathcal{V}$. Since G is strictly converging, each node, except node 1, has a single successor, which we denote as $s(i)$ for $i \in \mathcal{V}$. Nodes are numbered such that $s(i) < i$, $i \in \mathcal{V}$. We define $\mathcal{R} := \{i \in \mathcal{V} : \mathcal{P}(i) = \emptyset\}$ as the set of root nodes. The set of all nodes that can reach node i is denoted by $\mathcal{Y}(i)$. This set can be recursively defined as $\mathcal{Y}(i) := \mathcal{P}(i) \cup \left(\bigcup_{j \in \mathcal{P}(i)} \mathcal{Y}(j)\right)$, where $\mathcal{Y}(i) = \emptyset$ for $i \in \mathcal{R}$. Finally, we define $w_{i,j}$ as a path from node i to node j. It is the set of nodes that connects node i to node j, including nodes i and j.

4.2.2 Stochastic Leadtimes and Production Plan

A node in the network represents a task that needs to be executed. The duration of task i is the leadtime of node i. This leadtime is stochastic and modeled by the random variable T_i, $i \in \mathcal{V}$, contained in the vector of random variables $\mathbf{T} = (T_1, T_2, ..., T_N)$. The support of T_i is $(0, \infty)$ and T_i has a continuous distribution with finite expectation. The leadtimes $T_1, T_2, ...T_N$ are independent. The leadtime distributions are given and cannot be changed by our planning. For example, historical data can be retrieved from the manufacturer's SAP system to compute the leadtime distributions. If no data is available, the distribution can be estimated using existing techniques as described in Keefer and Verdini (1993).

A realization of the random vector $\mathbf{T}$ is indicated by $\omega = (t_1, t_2, ..., t_N) \in \Omega = \mathbb{R}_+^N$, where Ω is the sample space. In particular, $T_i(\omega)$ is a realization of T_i.

The variable t represents the time dimension. Without loss of generality the due date that is agreed upon with the customer is set to $t_0^s = 0$. In order to meet this due date, we control the production system by setting a planned start time t_i^s for all nodes $i \in \mathcal{V}$. The vector $\mathbf{t^s} = (t_1^s, ..., t_N^s)$ contains all planned start times and describes the production plan. The vectors $\mathbf{T}$ and $\mathbf{t^s}$ completely determine the realization of the production plan.

4.2.3 Example

We illustrate the definitions through an example. Consider the network in Figure 4.1. The same network is used as running example in the rest of the paper to clarify definitions and concepts.

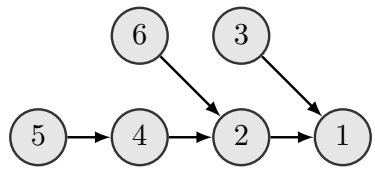


Figure 4.1: Example network

The example network consists of 6 nodes, i.e., there are 6 tasks. Nodes 6, 5 and 3

are root nodes. Node 4 can only start after node 5 is finished. The first assembly point is at node 2. Nodes 4 and 6 are the direct predecessors of it. The second assembly point is at node 1. Nodes 2 and 3 are direct predecessors of node 1. The network contains 2 assembly points and this makes it more complicated to analyze and optimize than an assembly system with only one assembly point.

An example of a production plan and a realization are shown in a Gantt chart in Figure 4.2.

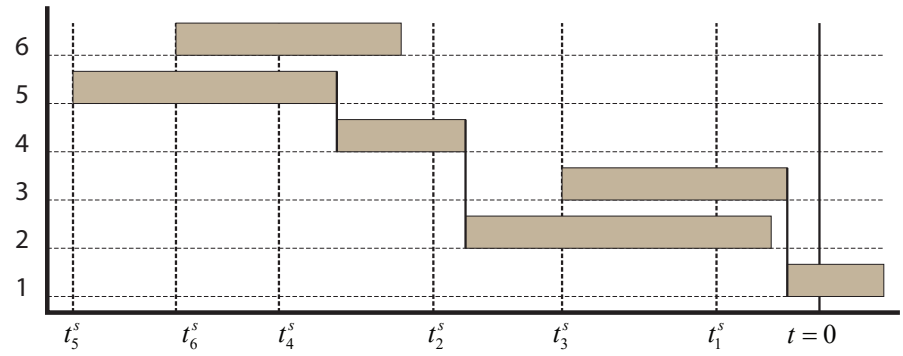


Figure 4.2: A candidate plan $\mathbf{t^s}$ and a random realization ω of the leadtime vector $\mathbf{T}$

Figure 4.2 shows a timeline with a planned start time and a realization of the leadtime for each node. In this example, the vector of planned start times is $\mathbf{t^s} = (-4, -15, -10, -21, -29, -25)$ and realizations of random leadtimes are $\omega = (3.75, 11.875, 8.75, 5, 10.25, 8.75)$. The start times are not in decreasing order, but if node i is a (direct) predecessor of node j, then $t_i^s < t_j^s$. The grey bars denote realizations of the random variables in $\mathbf{T}$. Node 5 is the first node to start and since it is a root node, it starts at its planned start time. It finishes later than the planned start time of node 4. Node 4 is a successor of node 5 and hence node 4 starts immediately after node 5 finishes. Meanwhile node 6, another root node, starts at its planned start time. Node 2 has two predecessors: nodes 6 and 4. In this example, node 4 is the latest and node 2 starts immediately after it. Node 6 finishes earlier than 4 and hence there is a waiting time between the finish times of nodes 6 and 2. Node 3 is a root node and starts at its planned start time and finishes later than t_1^s. For node 1, node 2 and 3 must be finished and hence node 1 starts immediately after node 3 finishes. Node 1 finishes later than $t = 0$ and hence the product is delivered after the due date. This situation is undesired, but this is only one realization of the

leadtime distributions. Exceeding intermediate deadlines, which in this example happens with nodes 1, 2 and 4 is also undesired. Therefore, we need to develop a production plan that meets intermediate deadlines and the final deadline as much as possible. Due to the uncertainty in the leadtimes it is impossible to always meet all deadlines. However, a plan that results in a minimum expected total cost can be developed. Such a plan finds the right balance between holding and penalty costs.

4.3. Tardy Paths

When a deadline in the production plan is exceeded, the start of any succeeding task is delayed. These delays can propagate throughout the network, causing even more delays further in the network. In the worst case, delays cause the final node to be late. On the other hand, succeeding nodes might be able to make up for delays. To relate a delay at a specific node to the cause of that delay, we introduce the concept of tardy paths. A tardy path can be seen as a sequence of deadlines that are exceeded. On this path, there is no waiting time between any adjacent tasks, i.e., tasks are executed back to back. The tardy path depends on the realization of the random leadtimes and the production plan. Thus, a path can be tardy in one realization, while it is not tardy in another realization. Whether a path is tardy or not can only be determined after all tasks on this path have been completed.

4.3.1 Actual Start and Finish Times

To simplify the definition of tardy paths, we introduce additional random variables that describe the actual start and finish times of a task. We define $A_i(\omega)$ and $B_i(\omega)$ as the actual start and finish times of task $i \in \mathcal{V}$ under the realization $\omega \in \Omega$. We have

$$A_i(\omega) = t_i^s, \quad i \in \mathcal{R}, \tag{4.1}$$

$$B_i(\omega) = A_i(\omega) + T_i(\omega), \quad i \in \mathcal{V}, \tag{4.2}$$

$$A_i(\omega) = \max\left\{\max_{j\in\mathcal{P}(i)}\{B_j(\omega)\}, t_i^s\right\}, \quad i \in \mathcal{V}\cup\{0\}\backslash\mathcal{R}. \tag{4.3}$$

Although A_i and B_i depend on $\mathbf{T}$ and $\mathbf{t^s}$, we do not denote this explicitly for ease of notation. In (4.1), each realization $A_i(\omega)$ equals t_i^s because i is a root node. In that case we could treat A_i as a deterministic variable, but for consistency amongst start times, we assume it is a random variable. The actual finish time of a node is the sum of its actual start time and its random leadtime (4.2). The actual start of a node with at least one predecessor depends on its own planned start time and the actual finish time of its predecessors (4.3). The start time $A_0(\omega)$ represents the completion time of the project, which is either the due date t_0 or later.

For the example network in Figure 4.1 and the realization in Figure 4.2, we have the following actual start times: $A_6(\omega) = t_6^s$, $A_5(\omega) = t_5^s$, $A_4(\omega) = t_5^s + T_5(\omega)$, $A_3(\omega) = t_3^s$, $A_2(\omega) = t_5^s + T_5(\omega) + T_4(\omega)$, $A_1(\omega) = t_3^s + T_3(\omega)$ and $A_0(\omega) = t_3^s + T_3(\omega) + T_1(\omega)$. The actual finish times are $B_6(\omega) = t_6^s + T_6(\omega)$, $B_5(\omega) = t_5^s + T_5(\omega)$, $B_4(\omega) = t_5^s + T_5(\omega) + T_4(\omega)$, $B_3(\omega) = t_3^s + T_3(\omega)$, $B_2(\omega) = t_5^s + T_5(\omega) + T_4(\omega) + T_2(\omega)$ and $B_1(\omega) = t_3^s + T_3(\omega) + T_1(\omega)$. Because this realization is an example of a late completion, it holds that $A_0(\omega) = B_1(\omega)$.

From equations (4.1)-(4.3) it is clear that the actual start time of a node depends only on the planned start times and random leadtimes of all nodes that can reach that node, and the planned start time of the node itself. This result is formalized in Lemma 4.1.

Lemma 4.1 *For $i \in \mathcal{V}$, the actual start time A_i only depends on the leadtimes T_j, $j \in \mathcal{Y}(i)$ and the planned start times t_j^s, $j \in \mathcal{Y}(i) \cup \{i\}$.*

For the proofs of all lemmas and theorems in this paper, we refer to Appendix 4.A. Note that this lemma also implies that the actual start time A_i does not depend on the leadtimes T_j, $j \in \mathcal{V} \setminus \mathcal{Y}(i)$ and the planned start times t_j^s, $j \in \mathcal{V} \setminus (\mathcal{Y}(i) \cup \{i\})$.

Similar to the actual start times we can formulate dependence and independence properties for the actual finish times. The only difference is that the actual finish time of an activity also depends on the leadtime realization of that activity, while the actual start time does not.

Lemma 4.2 *For all $i \in \mathcal{V}$ the actual finish time B_i only depends on the leadtimes T_j, $j \in \mathcal{Y}(i) \cup \{i\}$ and the planned start times t_j^s, $j \in \mathcal{Y}(i) \cup \{i\}$.*

Note that this lemma also implies that the actual finish time B_i does not depend

on the leadtimes T_j, $j \in \mathcal{V}\backslash(\mathcal{Y}(i) \cup \{i\})$ and the planned start times t_j^s, $j \in \mathcal{V}\backslash(\mathcal{Y}(i) \cup \{i\})$. Before defining tardy paths, we first introduce subgraphs and the corresponding actual start and finish times.

4.3.2 Subgraphs

Besides the original graph G, we can also consider so-called subgraphs. The idea is that we analyze part of the system as if it is a standalone system. We consider the subgraphs $G^{(i)} = (\mathcal{V}^{(i)}, \mathcal{E}^{(i)})$, $i \in \mathcal{V}$. Subgraph $G^{(i)}$ is obtained from G by removing the set of nodes $\mathcal{Y}(i)$. So $\mathcal{V}^{(i)} = \mathcal{V}\backslash\mathcal{Y}(i)$. If $i \in \mathcal{R}$, then $G^{(i)} = G$. In the example we have $\mathcal{V}^{(1)} = \{1\}, \mathcal{V}^{(2)} = \{1,2,3\}$ and $\mathcal{V}^{(4)} = \{1,2,3,4,6\}$. For the remaining nodes we have $G^{(3)} = G^{(5)} = G^{(6)} = G$.

For the subgraphs we use similar notation as for the original graph, but we add a superscript (i) to refer to the subgraph $G^{(i)}$. For example, $\mathcal{R}^{(i)}$ contains the root nodes of subgraph $G^{(i)}$. We compute the actual start and finish times of the activities, similar to (4.1)-(4.3):

$$A_k^{(i)}(\omega) = t_k^s, \quad k \in \mathcal{R}^{(i)}, \tag{4.4}$$

$$B_k^{(i)}(\omega) = A_k^{(i)}(\omega) + T_k(\omega), \quad k \in \mathcal{V}^{(i)}, \tag{4.5}$$

$$A_k^{(i)}(\omega) = \max\left\{\max_{j \in \mathcal{P}(k)}\{B_j^{(i)}(\omega)\}, t_k^s\right\}, \quad k \in \left(\mathcal{V}^{(i)} \cup \{0\}\right)\backslash\mathcal{R}^{(i)}. \tag{4.6}$$

Compared to the original graph, the planned start time t_k^s and the duration $T_k(\omega)$ of all nodes remain unchanged. Also for all nodes k not on path $w_{i,1}$ it holds that $A_k^{(i)}(\omega) = A_k(\omega)$ and $B_k^{(i)}(\omega) = B_k(\omega)$. Only start and finish times of nodes on the path $w_{i,1}$ can be shifted: $A_k^{(i)}(\omega) \leq A_k(\omega)$ and $B_k^{(i)}(\omega) \leq B_k(\omega)$. Using the start and finish times of subgraphs we can now define tardy paths.

Definition 4.1 Let $i \in \mathcal{V}$. For $\omega \in \Omega$, path $w_{k,j}$ in graph $G^{(i)}$ is called tardy if

(a) node k starts at its planned start time, $A_k^{(i)}(\omega) = t_k^s$,

(b) the successor of each node on path $w_{k,j}$, if there is one, starts immediately after that node is finished, $A_{s(l)}^{(i)}(\omega) = B_l^{(i)}(\omega)$ for all $l \in w_{k,j}$.

For nodes $i \in \mathcal{R}$ it holds that $G^{(i)} = G$ and thus a path $w_{k,j}$ is tardy in the original graph G. For any other subgraphs this is not necessarily the case. For example if $i = k$, then k is a root node in $G^{(i)}$ and thus by definition condition (a) holds. In the original graph G however, this condition is not satisfied if the predecessors of node i are late. Using the definition of a tardy path, we can define the event $\theta_{k,j}^{(i)}$. Event $\theta_{k,j}^{(i)}$ is the set of all realizations ω for which the path $w_{k,j}$ is tardy in subgraph $G^{(i)}$.

Definition 4.2 Let $i \in \mathcal{V}$. For path $w_{k,j}$ in $G^{(i)}$ the event $\theta_{k,j}^{(i)}$ is defined as

$$\theta_{k,j}^{(i)} = \left\{ \omega : A_k^{(i)} = t_k^s, A_{s(l)}^{(i)}(\omega) = B_l^{(i)}(\omega), l \in w_{k,j} \right\}.$$

For root nodes it holds that $G^{(i)} = G$ and thus we omit the superscript and define $\theta_{k,j}^{(i)} = \theta_{k,j}, i \in \mathcal{R}$. In the rest of the paper, we mainly focus on events $\theta_{i,j}^{(i)}$, corresponding to tardy paths starting in root node i. In subgraph $G^{(i)}$, the actual finish time of each node on a path $w_{i,j}$ must equal the actual start time of its successor. In other words, there can be no waiting time between two consecutive nodes on path. In case $j = 1$, we require that $A_0^{(i)} > t_0^s$, implying that the final due date is exceeded. By considering the subgraph $G^{(i)}$ we ignore possible delays occurring before i. From a theoretical perspective, this definition helps us in shortening notation of events in the sample space Ω, which is especially useful in the analysis of systems with a large number of nodes. Writing this event in terms of $\mathbf{T}$ and $\mathbf{t^s}$ would be cumbersome.

4.3.3 Example

Next, we clarify the concept of tardy paths through an example. We refer to Figure 4.2 and start with the root nodes, 6,5 and 3. Node 6 finishes earlier than the actual start time of its successor, node 2: $A_2(\omega) > B_6(\omega)$. Hence, the realization ω is not included in the events $\theta_{6,6}$, $\theta_{6,2}$ or $\theta_{6,1}$. Next we consider node 5. Its successor, node 4, starts immediately after node 5 finishes. There is also no waiting time between nodes 5 and 2, but there is waiting time between nodes 2 and 1. Hence, the realization ω is included in the events $\theta_{5,5}$ and $\theta_{5,4}$, but not included in the events $\theta_{5,2}$ and $\theta_{5,1}$. The final root node is node 3. Since there is no waiting time between node 3 and node 1 and node 1 finishes after $t = 0$, the realization ω is included in

both events $\theta_{3,3}$ and $\theta_{3,1}$.

Node 4 is not a root node and hence we consider the subgraph $G^{(4)}$ in which node 5, the only predecessor of node 4 is excluded. We consider the same realization ω. In G $A_4(\omega) = t_5^s + T_5(\omega) = -18.25$ but in the subgraph $G^{(4)}$ the start time is shifted and we have $A_4^{(4)}(\omega) = t_4^s = -21$. Thus, activity 4 is shifted in time, but the duration itself remains unchanged. We can do this for each activity until the actual finish time of its latest predecessor or its own actual start time is reached. The Gantt chart for subsystem $G^{(4)}$ with shifted realizations and start and finish times $A_i^{(4)}(\omega), B_i^{(4)}, i \in \mathcal{V}\backslash\mathcal{Y}(4)$ is shown in Figure 4.3a. Due to the shifts, waiting times between nodes can change. For example, there was no waiting time between node 4 and its successor, node 2, in the original system, but there is waiting time between nodes 4 and 2 in the subsystem $G^{(4)}$. Hence, the realization ω is not included in the events $\theta_{4,4}^{(4)}$, $\theta_{4,2}^{(4)}$ and $\theta_{4,1}^{(4)}$.

In Figure 4.3b the Gantt chart for subsystem $G^{(2)}$ with shifted leadtimes is shown. In this subsystem, all predecessors of 2, i.e., nodes 4,5 and 6, are excluded. Shifting the actual start time of node 2 back to its planned start time results in even more waiting time between nodes 2 and 1. Due to this waiting time, (which is caused by node 3) the realization ω is not included in the events $\theta_{2,2}^{(1)}$ and $\theta_{2,1}^{(1)}$.

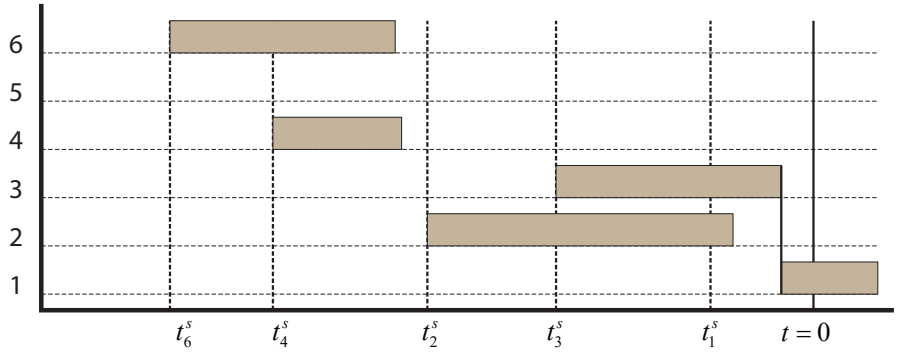
Finally in Figure 4.3c all predecessors of node 1 are excluded, which makes it the only node. Due to the shift to its planned start time, node 1 now finishes in time and thus the realization ω is not included in the event $\theta_{1,1}^{(1)}$.

4.3.4 Properties of Tardy Paths

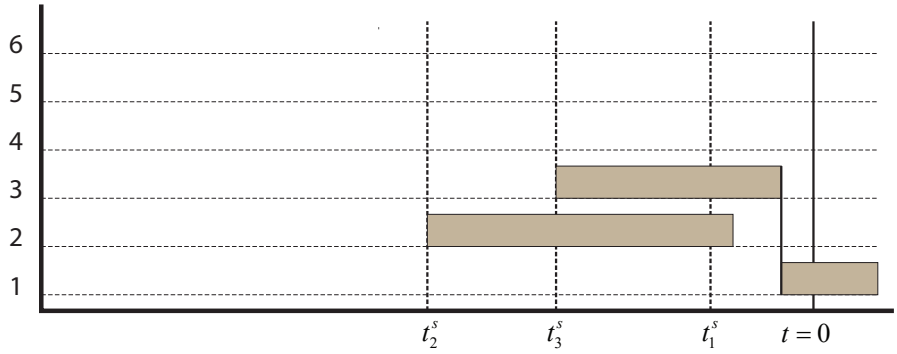
The events $\theta_{k,j}^{(i)}$ are defined on the sample space Ω, ut only depend on a subset of the random leadtimes in $\mathbf{T}$. For $i = k$, Lemma 4.3 indicates which leadtimes affect event $\theta_{i,j}^{(i)}$ and which not.

Lemma 4.3 *Event $\theta_{i,j}^{(i)}$ depends on $T_l, l \in \mathcal{Y}(s(j))\backslash\mathcal{Y}(i)$ and is independent of $T_l, l \in \mathcal{Y}(i) \cup \mathcal{V}\backslash\mathcal{Y}(s(j))$.*

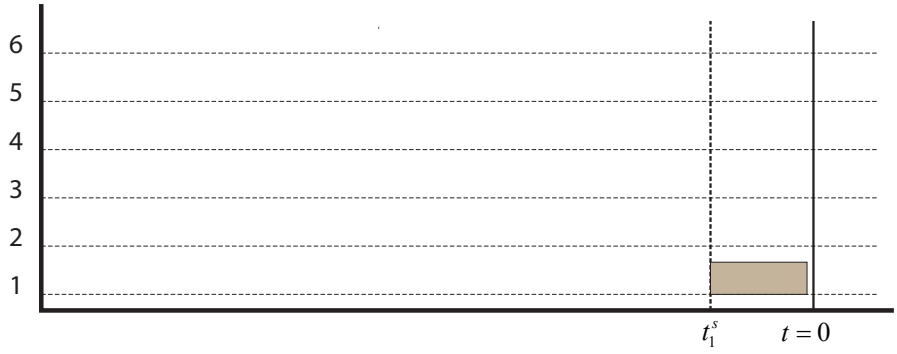
Figure 4.4 provides a schematic representation of the relations in Lemma 4.3. This figure shows a converging network and two arbitrarily chosen nodes i and j. We indicate the nodes of which the leadtimes affect the event $\theta_{i,j}^{(i)}$ and those who do



(a) Subsystem $G^{(4)}$



(b) Subsystem $G^{(2)}$



(c) Subsystem $G^{(1)}$

Figure 4.3: Realization ω for different subsystems of the system in Figure 4.1

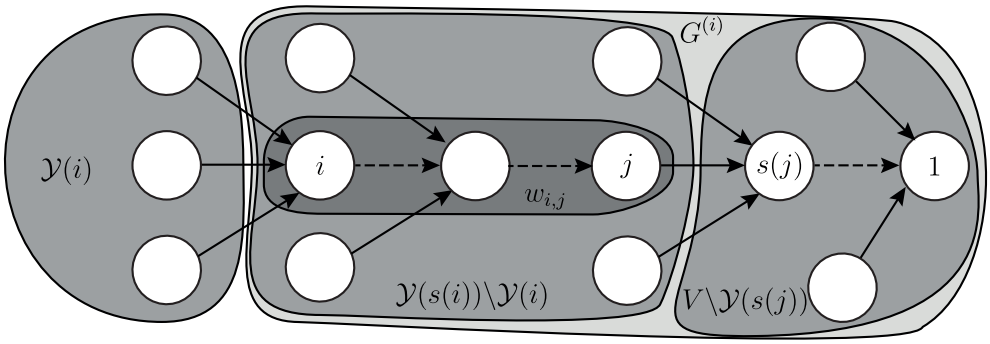


Figure 4.4: Schematic overview showing dependency of event $\theta_{i,j}^{(i)}$ on random leadtimes of different nodes.

not. Since event $\theta_{i,j}^{(i)}$ is defined on subgraph $G^{(i)}$, the event is independent of the leadtimes of nodes that can reach node i, i.e., all nodes in $\mathcal{Y}(i)$. For each node on path $w_{i,j}$ we require that $A_{s(k)}^{(i)}(\omega) = B_k^{(i)}(\omega)$. As a consequence, we require that all other predecessors of $s(k)$ finish earlier: $B_l^{(i)}(\omega) < B_k^{(i)}(\omega) \forall l \in \mathcal{P}(s(k))\backslash\{k\}$. Hence event $\theta_{i,j}^{(i)}$ depends on the leadtimes of all nodes $k \in \mathcal{Y}(s(j))\backslash\mathcal{Y}(i)$. Finally, event $\theta_{i,j}^{(i)}$ does not depend on the leadtimes of nodes that are not predecessors of $s(j)$.

For a given realization, there can be multiple tardy paths. However, the most interesting paths are tardy paths that end at node 1. If there is a tardy path ending at node 1, this means that the final deadline, which is the most important deadline, is exceeded. When a product is delivered late, it is important to look for the cause of this lateness.The tardy path concept helps identifying the nodes that cause lateness at the final deadline. From Definition 4.1 it directly follows that in a graph, there can be only one tardy path ending at node 1. For the realization shown in Figure 4.2 path $w_{3,1}$ is the tardy path ending at node 1. Also for the subgraphs it holds that there can be at most 1 tardy path. In the subgraphs $G^{(4)}$ and $G^{(2)}$ the path $w_{3,1}$ is also tardy, but in the subgraph $G^{(1)}$ there is no tardy path.

The events $\theta_{i,j}^{(i)}$ are not necessarily mutually exclusive. There can be multiple tardy paths starting at node i. For example, in Figure 4.1, if the path $w_{4,1}$ is a tardy path, paths $w_{4,2}$ and $w_{4,4}$ are also tardy. This property is formalized in Lemma 4.4.

Lemma 4.4 *If $w_{i,j}$ is a tardy path in $G^{(i)}$ then there is also a tardy path from node i to any*

other node on path $w_{i,j}$ in the subgraph $G^{(i)}$, i.e.,

$$\theta_{i,k}^{(i)} \subset \theta_{i,j}^{(i)} \quad k \in w_{i,j}.$$

This Lemma also implies that if a path from i to j is not tardy, then none of the paths from i to any other node on the path $w_{i,j}$ is tardy either.

The events $\theta_{k,j}^{(i)}$ are defined on the sample space Ω. We obtain the following connection between the original system and a subsystem i.

$$\begin{aligned} P(\theta_{i,j}) &= P(A_i = t_i^s, \theta_{i,j}^{(i)}) \\ P(\theta_{i,j}) &= P(A_i = t_i^s) P(\theta_{i,j}^{(i)}) \\ P(\theta_{i,j}^{(i)}) &= P(\theta_{i,j} | A_i = t_i^s) \end{aligned} \tag{4.7}$$

These probabilities are useful from a practitioner's perspective. Given a production plan with planned start times $\mathbf{t^s}$, these probabilities indicate which parts of the system are time critical and where delays are expected.

4.4. Cost Structure and Optimization Problem

In the previous sections, we introduce a stochastic network model and the concept of tardy paths. In this section we use this model to introduce a cost structure for the network and define the optimization problem.

4.4.1 Derivation of the Total Expected Cost

The objective of this study is to develop a production plan for a manufacturer such that the expected total cost is minimized. Before starting with producing the product, the manufacturer needs to purchase raw materials, parts and subassemblies from external suppliers. In addition, labor and special equipment might also be needed. We consider the costs for all these resources as constant. Although the costs are constant, the point in time these costs are paid might vary. These expenses are usually paid by the customer before the final product is completed.

Since we aim to optimize the production plan, we consider the costs that can be

affected by our planning. For example, companies aim to reduce the amount of working capital. Capital usage does not necessarily increase linearly over time but it heavily depends on the production planning. Postponing the assembly of an expensive part reduces the cost of working capital, but might delay the project. On the other hand, having cheap parts available early avoids delays in the production process.

In our model, we refer to the cost of working capital as holding cost. We assume that each node in the network incurs holding cost from the actual start time of this node. The system incurs a marginal holding cost $h_i > 0$ from the actual start time A_i of node i until the final product is delivered to the customer. The marginal holding cost h_i represents the value added by node i. We define total holding cost added by all nodes as $h_1^c = \sum_{i \in \mathcal{V}} h_i$, which is the total cost incurred during the execution of node 1. In addition to the holding costs, the manufacturer incurs a penalty cost p per unit time late for delivery to the customer. The value of p can be agreed with a customer as a penalty that is actually paid or it can be derived from a predetermined service level. The total cost $C(\mathbf{t^s})$ is the sum of the holding costs for all nodes and the expected penalty cost for late deliveries.

$$C(\mathbf{t^s}) = \sum_{i \in \mathcal{V}} h_i \left(A_0 - A_i\right) + p(A_0 - t_0^s). \tag{4.8}$$

The first term is the summation of the holding cost that is incurred at each node from the actual start time of the node until the delivery of the end product. The delivery time is $A_0 = \max\{B_1, t_0^s\}$. If the product is finished earlier than planned, it is delivered at t_0^s, otherwise it is delivered at B_1. The second term denotes the penalty cost, which is incurred only if $A_0 > t_0^s$. Note that A_i are stochastic variables and, although not explicitly stated, they depend on $\mathbf{T}$ and $\mathbf{t^s}$.

A similar cost structure is commonly considered in the literature (Atan et al., 2017; Elhafsi, 2002; Yano, 1987a; Axsäter, 2005). However, most studies use a different approach for calculating the holding cost. Instead of incurring holding cost throughout the complete production, it is incurred only when a node is waiting, i.e. when $A_{s(i)} > B_i$ for some node i. This waiting time is either due to the fact that i finishes before the planned start time of $s(i)$ or due to the fact that other direct predecessors of $s(i)$ are not yet finished. If the holding costs are charged only when

nodes are waiting, the total cost becomes

$$C'(\mathbf{t^s}) = \sum_{i \in \mathcal{V}\setminus\{1\}} \left[(A_{s(i)} - B_i) \sum_{k \in \mathcal{Y}(i)} h_k \right] + (A_0 - B_1) \sum_{i \in \mathcal{V}} h_i + (A_0 - t_0^s)p. \tag{4.9}$$

Subtracting (4.9) from (4.8) yields

$$\sum_{i \in \mathcal{V}} T_i \left(\sum_{j \in \mathcal{Y}(i)} h_j \right). \tag{4.10}$$

This term denotes the duration per node, multiplied by the holding cost incurred by that node and all its preceding nodes. Since this term does not depend on planned start times, the optimal production plans are the same independent of whether (4.8) or (4.9) is chosen as the total cost.

4.4.2 Optimization Problem

Given the leadtime distributions for all nodes, we want to choose $\mathbf{t^s}$ such that the expected cost $\mathbb{E}[C(\mathbf{t^s})]$ is minimized. Hence, our optimization problem is:

$$\min_{\mathbf{t^s}} \mathbb{E}[C(\mathbf{t^s})]. \tag{4.11}$$

Since any solution in $\mathbb{R}^N$ is feasible, a solution where $t_i^s > t_{s(i)}^s$ is valid. In that case, due to the precedence relations, the successor of node i can never start at its planned start time, since node i has not even started yet. We refer to these pairs of start times as reversed pairs. We define the set of solutions that do not contain any reversed pairs as

$$\mathbb{D} = \{\mathbf{t^s} : t_i^s < t_{s(i)}, i \in \mathcal{V}\}.$$

$\mathbb{D}$ is an open set. We define the closed version of $\mathbb{D}$ as

$$\mathbb{D}' = \{\mathbf{t^s} : t_i^s \leq t_{s(i)}, i \in \mathcal{V}\}.$$

This set also contains the solutions where planned start times of consecutive nodes can be equal.

Solutions that contain reversed pairs are not desired in practice, but are mathematically valid. Therefore, we need to consider them in our optimization problem. However, in the case of a reversed pair planned start times $t_i^s > t_{s(i)}^s$, the variable $t_{s(i)}^s$ does not have any effect on the actual start times A_i and $A_{s(i)}$. In this case the expected costs do not change when we change $t_{s(i)}^s$. We can modify the solution in such a way that we find another solution leading to the same cost that is in $\mathbb{D}'$ but not in $\mathbb{D}$

Lemma 4.5 *For each solution* $\mathbf{t^s} \notin \mathbb{D}'$ *there exists a solution* $\mathbf{t^{s\prime}} \in \mathbb{D}'$ *such that* $C(\mathbf{t^s}) = C(\mathbf{t^{s\prime}})$.

This lemma helps us to find a solution to (4.11), since we only need to consider solutions in $\mathbb{D}'$. A solution that is optimal within $\mathbb{D}'$, is also a global optimum:

$$\min_{\mathbf{t^s}} \mathbb{E}[C(\mathbf{t^s})] = \min_{\mathbf{t^s} \in \mathbb{D}'} \mathbb{E}[C(\mathbf{t^s})].$$

In the next section we further analyze the optimization problem and identify necessary and sufficient conditions for the optimal solution.

4.5. Optimal Solution and Critical Tardy Paths

In Section 4.4.2 we introduced our optimization problem and in Section 4.3 we introduced the concept of tardy paths. In this section, we use the tardy path concept to derive optimality equations for the optimization problem and define critical tardy paths.

4.5.1 General Newsvendor Equation

The optimization problem (4.11) balances holding cost incurred after the start of each node and penalty costs incurred after t_0^s. The problem is unconstrained and hence any solution $\mathbf{t^s} \in \mathbb{R}^N$ is feasible. In order to find the optimal solution of this problem we derive a necessary condition for an optimum. This condition is the general Newsvendor equation, which equals the probability of exceeding the final deadline to the total holding cost divided by the total holding and penalty cost.

Lemma 4.6 *Any solution* $\mathbf{t^s} \in \mathbb{R}^N$ *that is an optimal solution to* (4.11) *satisfies*

$$P(A_0 > 0) = \frac{h_1^c}{h_1^c + p}. \tag{4.12}$$

4.5.2 Optimality Equations

We take the partial derivatives of (4.8). The leadtime distributions are continuous and differentiable. Hence, $\mathbb{E}[C(\mathbf{t^s})]$ is differentiable as well. A necessary condition for optimality is $\frac{\partial}{\partial t_i^s}\mathbb{E}[C(\mathbf{t^s})] = 0,\ i \in \mathcal{V}$. In this way, we obtain a system of equations that should be satisfied by the optimal solution. This set of equations is formalized as follows.

Theorem 4.1 *An optimal solution* $\mathbf{t^s} \in \mathbb{D}$ *to* (4.11) *satisfies the following set of equations.*

$$P(\theta_{i,1}^{(i)}) = \frac{h_i}{h_1^c + p} + \sum_{j \in w_{i,1} \setminus \{1\}} \frac{h_{s(j)} P(\theta_{i,j}^{(i)})}{h_1^c + p}, \quad i \in \mathcal{V} \tag{4.13}$$

The theorem states that under the optimal solution the probability that the path $w_{i,1}$ is tardy equals to a sum of Newsvendor fractiles. The first fractile is the holding costs of node i divided by the total holding and penalty cost of the final product. The summation term consists of similar Newsvendor fractiles, one for each node on the path from i to node 1. Each term depends on the holding cost of its successor $s(j)$ and on the probability that a tardy path exists from i to j. Note that $j = i$ is also part of the summation. In that case $P(\theta_{i,i}^{(i)}) = P(A_{s(i)}^{(i)} = B_i^{(i)})$.

For a network with only one node the theorem reduces to the classical Newsvendor problem. Since $N = 1$, $i = 1$ and $w_{1,1} \setminus \{1\} = \emptyset$, the summation term vanishes. The total holding cost equals the holding costs of node 1 $h_1 = h_1^c$ and the event $\theta_{1,1}$ covers all cases in which the product is delivered late. Hence, we find the result of Lemma 4.6.

It is important to note that Theorem 4.1 and its proof heavily benefit from the definition of $\theta_{i,j}^{(i)}$. This definition enables having a Newsvendor-type of a solution. The usage of these events makes the optimality equations compact and easy to understand/interpret. It might also be possible to have the expressions in terms of $\mathbf{T}$ and $\mathbf{t^s}$ but especially for networks with multiple assembly points, the expressions

would be extremely long and difficult to interpret.

4.5.3 Critical Tardy Paths

For a given realization of leadtimes ω there can be multiple tardy paths. This is the case for the realization in Figure 4.3. We are especially interested in tardy paths that have node 1 as their final node. If such a path is tardy, the final product is delivered late. Following the nodes in a tardy path backwards can provide an indication on the cause of the late delivery. In fact, for each late delivery we can identify a sequence of activities that are executed back to back, i.e., nodes for which $B_k^{(i)}(\omega) = A_{s(k)}^{(i)}(\omega)$. Given that the leadtimes are continuous random variables, if we move backwards on a tardy path, we can find exactly one node for which $A_k^{(i)}(\omega) = t_k^s$. Once we find such a node, we know that the tardy path starts at node k, i.e, the path $w_{k,1}$ is tardy. This directly implies that there is no other tardy path that ends at node 1. We formalize this result in Lemma 4.7.

Lemma 4.7 *If* $\omega \in \theta_{k,1}^{(i)}$ *then* $\omega \notin \bigcup_{l \in \mathcal{V}^{(i)}} \theta_{l,1}^{(i)}$.

Next, using the definition of $\theta_{i,j}^{(i)}$, we define new sets of events α_i. If a realization ω is part of the event α_i, then it is the first node for which it holds that when analyzing the subgraph $G^{(i)}$ the path $w_{i,1}$ is tardy. We formalize the definition of these events as follows:

Definition 4.3 For $i \in \mathcal{V}$, the event α_i is defined as

$$\alpha_i = \{\omega : \omega \in \theta_{i,1}^{(i)}, \omega \notin \theta_{k,1}^{(k)}, k < i\}.$$

Definition 4.4 For a realization ω, the path $w_{i,1}$ is called a critical tardy path if $\omega \in \alpha_i$.

If the event α_i realizes, we know that the subsystem is late due to node i. If a subystem is late, also the original system is late. In the original system, node i is at least one of the nodes causing the exceedance of the due date. The definition of α_i doesn't require node i to start at its planned start time in the original system G. But if the event α_i realizes, node i is the first node (starting from the node 1) for which its corresponding subsystem is also late. In other words, in the subsystem $G^{(i)}$,

changing the planned start time of node i, t_i^s, by an infinitesimal amount δ would result in a change δ in $A_0^{(i)}(\omega)$. For all other nodes in the subsystem, changing the planned start time by an infinitesimal amount δ would have no effect on the finish time of node 1. Applying the definition of α_i, we find that the realization in Figure 4.2 is only part of $\theta_{3,1}^{(3)}$ and not of any other $\theta_{j,1}^{(j)}$. It then follows automatically that this realization is part of α_3 and the path $w_{3,1}$ is a critical tardy path in the original system.

In Corollary 4.1, we summarize the properties of events α_i. These properties help us in proving one of the main results of this study (see Theorem 4.2).

Corollary 4.1 *The sets α_i satisfy the following properties:*

1. *The sets α_i, $i \in \mathcal{V}$ are mutually exclusive.*
2. *The set $\bigcup_{i\in\mathcal{V}} \alpha_i$ covers all the events in which the system is late.*

From this corollary it follows directly that $\sum_{i\in\mathcal{V}} P(\alpha_i) = P(A_0 > 0)$, i.e. the probability of exceeding the final due date.

When changing one or more planned start times, the probabilities of the critical tardy paths can change as well. We formulate these monotonictiy properties in the following lemma.

Lemma 4.8 *The probability of a critical tardy path $P(\alpha_i)$ is*

1. *not depending on t_k^s, $\quad k \notin V^{(i)}$,*
2. *strictly increasing in t_i^s,*
3. *decreasing in t_j^s, $\quad j \in V^{(i)}\backslash\{i\}$.*

The first observation is that start times of preceding nodes have no effect on the critical tardy path probability of node i. This can be seen directly from the definition of α_i. Since α_i is defined on the subgraph $G^{(i)}$ it can not be a function of start times of nodes preceding i. Secondly, increasing t_i^s means that the planned start of node i is postponed and that the system will be late more often due to this node, i.e. the probability of a critical tardy path α_i increases. Finally, increasing another start time t_j^s also has an effect on the tardy path probability of α_i. When increasing j, the

probability $P(\alpha_j)$ increases, but due to mutual exclusivity of the events, this increase is partly at the expense of other events such as α_i.

The lemma shows the consequences of changing one start time for only one tardy path probability. However, we are also interested the effect of changing a start time on summations of tardy path probabilities. In particular, we are interested in the sum of all tardy path probabilities since this equals the probability that the final deadline is met.

Lemma 4.9 *Let $\mathcal{U} \subseteq \mathcal{V}$. The sum of critical tardy path probabilities $\sum_{i\in\mathcal{U}} P(\alpha_i)$ is*

1. *increasing in t_i^s, $i \in \mathcal{U}$,*
2. *strictly increasing in t_i^s, $i = \max \mathcal{U}$,*
3. *decreasing in t_j^s, $j \notin \mathcal{U}$.*

From Lemma 4.8 it follows that when increasing t_i^s some tardy path probabilities decrease, while $P(\alpha_i)$ increases. Lemma 4.9 gives us an insight in the magnitude of these changes. In any chosen set $\mathcal{U}$ the increase of $P(\alpha_i)$ is always higher than the combined decrease of the rest of the tardy path probabilities. When $\mathcal{U} = \mathcal{V}$ it then follows that an increase in t_i^s leads an increase in the probability that a system is late. This is an intuitive result, since increasing a start time means that there is overall less time to complete the project, resulting in more delays at the end. Besides providing these insights, the lemmas also help in deriving the optimality equations in the next section.

4.5.4 Newsvendor Equations

Using the definitions of events α_i and critical tardy paths, we can formulate the following system of Newsvendor equations.

$$P(\alpha_i) = \frac{h_i}{h_1^c + p}, \quad i \in \mathcal{V}. \tag{4.14}$$

Each equation consists of a tardy path probability and a Newsvendor fractile. Under a solution that satisfies this system of equations the probability of a critical tardy path starting in a node i is proportional to the value that is added by that node.

Each event α_i is a subset of the corresponding event $\theta_{i,1}^{(i)}$ and therefore only depends on the planned start times of nodes in $\mathcal{V}^{(i)}$. This means that for all nodes until the first assembly point the equations can be solved recursively. For node one, one needs to to solve

$$P(T_1 > t_0^s - t_1^s) = \frac{h_1}{h_1^c + p}.$$

In this equation t_1^s is the only decision variable and because T_1 has a continuous distribution, it is easy to see that solving this equation leads a unique solution. We can derive a similar result for the system of equations by using the monotonicity results of Lemmas 4.8 and 4.9.

Lemma 4.10 *A solution* $\mathbf{t^s}$ *that satisfies* (4.14) *is unique.*

This result can be combined with the result of Theorem 4.1. The equations of this theorem can under the assumption that $\mathbf{t^s} \in \mathbb{D}$ be rewritten to (4.14). Since this solution is unique, the equations are a necessary and sufficient condition for a global optimum.

Theorem 4.2 *A solution* $\mathbf{t^s} \in \mathbb{D}$ *is an optimal solution to* (4.11) *if and only if it satisfies* (4.14)*.*

According to Theorem 4.2, under the optimal solution, the more value a node adds to the final product, the higher the probability of a critical tardy path starting at that node. In practice this means that once an activity that adds a lot of value to the product is started, there should be no waiting time between this time point and the completion of the final product.

A solution $\mathbf{t^s} \in \mathbb{D}$ satisfying Theorem 4.2 does not always exist. A simple counter example is the two stage serial system discussed in Section 2.6.2. This example shows an optimal *planned leadtime* solution containing negative optimal planned leadtimes. Converting this solution to a *planned start time* solution yields a solution which is outside the domain $\mathbb{D}'$, but does satisfy (4.14). According to Lemma 4.5 there exists a corresponding solution in $\mathbb{D}'$ yielding the same expected cost. However, this solution does violate (4.14).

The problem with a solution $\mathbf{t^s} \notin \mathbb{D}'$ is that there is at least one node that never starts at its planned start time, i.e. $P(A_i = t_i^s) = 0$. In the proofs of Theorem 4.1 and

4.2 we divide by this probability, assuming that it is nonzero. Therefore, we cannot claim that a solution $\mathbf{t^s} \notin \mathbb{D}'$ that satisfies (4.14) is an optimum. However, our numerical experiments indicate that also in this case the solution is an optimum. We formulate this as a conjecture:

Conjecture 4.1 A solution $\mathbf{t^s} \notin \mathbb{D}'$ that satisfies the set of equations (4.14) is a global optimum to (4.11).

These properties can be used to develop accurate and efficient algorithms to solve real life problems. For example, in Atan et al. (2016) already a fast and accurate algorithm is developed for a system with 1 assembly point, using a bisection algorithm. One could further extend this algorithm to solve for the optimal start times of activities in general converging networks. The structural results in this study can be exploited to develop the algorithm, but also to verify the generated solutions.

4.6. Numerical Analysis

In this section, we will illustrate the structural results by numerical examples. We use an production network introduced by Axsäter (2005). The network consists of 8 activities and is shown in Figure 4.5. To obtain the production plans we use simulation based optimization. To analyze the performance of the obtained plans we simulate them using a different sample set. For the details of this procedure we refer to Section 3.5.

4.6.1 Service Level and Planned Leadtime

We are interested in how under the optimal solution, time is allocated to an activity depending on its location in the network. For ea activity in the network we assume a stochastic leadtime that is exponentially distributed with $\lambda = 1$. Furthermore, we assume that each node adds equal value to the final product, $h_i = 1, i \in \mathcal{V}$. Since the allocated time heavily depends on the desired service level, we change the penalty cost p in such a way that the optimal service level $\frac{p}{h_1^c+p}$ is changed between 0.6 and 0.95.

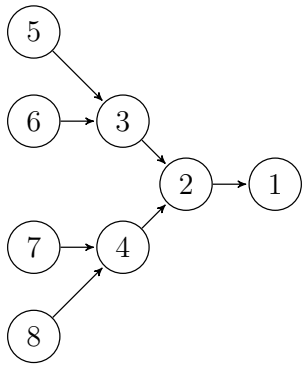


Figure 4.5: Production network modified from Axsäter (2005).

	60%			70%			80%			90%			95%		
i	t_i^s	$P(\alpha_i)$	gap	t_i^s	$P(\alpha_i)$	gap	t_i^s	$P(\alpha_i)$	gap	t_i^s	$P(\alpha_i)$	gap	t_i^s	$P(\alpha_i)$	gap
1	-2.99	0.050	0.1%	-3.28	0.037	-0.2%	-3.69	0.025	0.2%	-4.38	0.013	0.5%	-5.10	0.006	-2.7%
2	-4.10	0.050	-0.8%	-4.46	0.038	1.6%	-4.98	0.025	1.4%	-5.84	0.013	1.0%	-6.71	0.006	1.5%
3	-4.79	0.051	1.3%	-5.31	0.038	0.2%	-5.98	0.025	0.7%	-7.06	0.012	-1.1%	-8.04	0.006	2.8%
4	-4.81	0.049	-1.2%	-5.31	0.037	-1.0%	-5.98	0.025	0.3%	-7.07	0.012	-1.6%	-8.06	0.006	0.3%
5	-5.59	0.050	-0.1%	-6.21	0.037	-0.4%	-7.02	0.025	-1.5%	-8.24	0.012	-0.2%	-9.34	0.006	2.9%
6	-5.60	0.050	-0.5%	-6.22	0.038	0.2%	-7.00	0.025	0.1%	-8.25	0.012	-2.6%	-9.35	0.006	0.6%
7	-5.59	0.051	1.0%	-6.22	0.037	-0.6%	-7.01	0.025	0.7%	-8.25	0.012	-1.1%	-9.36	0.006	-1.2%
8	-5.59	0.050	0.4%	-6.21	0.038	0.1%	-7.01	0.025	0.9%	-8.24	0.012	-0.3%	-9.36	0.006	-0.3%

Table 4.1: Results service level experiment

To obtain the optimal production plan under the 'pay as realized' cost function we solve the minimization problem (4.11) The results are presented in Table 4.1. The table shows the optimal production plans to achieve a service level of 60%, 70%, 80%, 90% and 95%. For each service level we show the planned start time t_i^s for each node. Furthermore, we show the probability of a critical tardy path $P(\alpha_i)$ which is obtained via simulation. For each node, this probability should equal the Newsvendor fractile. For example, for the 70% service level the probability should equal $\frac{h_i}{h_1^c+p} = \frac{1}{8+18.667} = 0.0375$. We compute for each node the relative gap as

$$\frac{P(\alpha_i) - \frac{h_i}{h_1^c+p}}{\frac{h_i}{h_1^c+p}} \cdot 100\%.$$

The first observation is that due to identical distributions, equal holding costs and the symmetric network structure, the results for nodes 3 and 4 and the results for nodes 5,6,7 and 8 are equal. Furthermore, all planned start times are in the right order, i.e. the solution is in $\mathbb{D}$. The optimization method is quite accurate with relative gaps of the critical tardy path probabilities all below 3%. This gap could

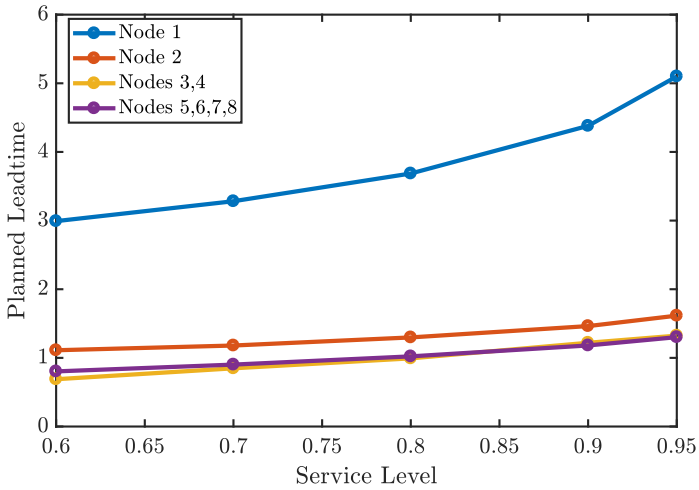


Figure 4.6: Optimal planned leadtimes under 'pay as realized' cost function for the network of Figure 4.5

be further decreased when increasing the sample sizes of the optimization problem and the subsequent simulation.

When looking at the planned start times one can clearly see the trade-off between throughput time and service level. The higher the desired service level, the longer the planned duration. For example, for a service level of 60%, the first nodes (5-8) start at $t = -5.6$, while for a 95% these nodes should start at $t = -9.36$.

Besides the planned start times, we can also look at the planned leadtime for each node. The planned leadtime is defined as $t^s_{S(i)} - t^s_i$. In Figure 4.6 we show the planned leadtimes for all nodes. Because of the symmetry, we combine the results of nodes 3 and 4 and nodes $5-8$. The figure shows for each node how the planned leadtime changes as function of the service level. For each node in the network, the planned leadtime increases as function of the service level. This also results in the fact that intermediate deadlines are met more often. However, the expected cost of the system increase too.

Although the activities have the same leadtime distribution and add the same value to the final product there is a big difference in the planned leadtime per node. For the 60% service level, node 1 has a planned leadtime of 2.99, almost 3 times its average duration. Nodes $5-8$ are allocated only 0.78, 22% less than their average duration. The reason for this is that nodes 5-8 always start on time and a late

	pay as realized			pay as planned		
	t_i^s	$t_{\mathcal{S}(i)}^s - t_i^s$	$P(A_i = t_i^s)$	t_i^s	$t_{\mathcal{S}(i)}^s - t_i^s$	$P(A_i = t_i^s)$
1	-4.38	4.38	0.410	-3.77	3.77	0.544
2	-5.84	1.46	0.254	-5.22	1.45	0.474
3	-7.06	1.22	0.482	-6.83	1.62	0.679
4	-7.07	1.22	0.482	-6.83	1.62	0.676
5	-8.24	1.18	1.000	-8.57	1.74	1.000
6	-8.25	1.19	1.000	-8.57	1.74	1.000
7	-8.25	1.19	1.000	-8.58	1.75	1.000
8	-8.24	1.18	1.000	-8.57	1.74	1.000

Table 4.2: Details of optimal production plans under 'pay as planned' and 'pay as realized' costing schemes.

completion of these nodes can be recovered later in the network. Node 1 is the final node, which is not guaranteed to start at its planned start time, due to the preceding stochastic activities. Furthermore, delays in node 1 cannot be recovered anymore by any succeeding nodes. Therefore it is assigned a longer planned leadtime. To some extend this also holds for node 2. In general one could say that the closer a node is to the final deadline the higher the optimal planned leadtime for that node.

4.6.2 Comparison with 'Pay as Planned' Solution

In Section 2.6 we already compared the 'pay as planned' and the 'pay as realized' cost function for an assembly system. We showed that the optimal planning solutions differ significantly. We can do a similar analysis for the network in Figure 4.5. We again use an exponential distribution with $\lambda = 1$ and holding cost $h_i = 1$ for each node. We use $p = 72$ leading a 90% service level.

For both cost functions the results are shown in Table 4.2. This table shows the planned start times, the planned leadtimes $t_{\mathcal{S}(i)}^s - t_i^s$ and the probability that a node starts at its planned start time $P(A_i = t_i^s)$. The results are graphically shown in Figure 4.7. This figure shows a timeline ending at $t = 0$, the planned finish time of both nodes. Each vertical line denotes a planned start time. Horizontal lines connect the start times according to the precedence relations of the network. Red lines show the optimal plan under the 'pay as planned' cost function while the blue lines indicate the optimal plan under the 'pay as realized' cost function.

Under the 'pay as planned' costing scheme production is started earlier ($t_8^s =$

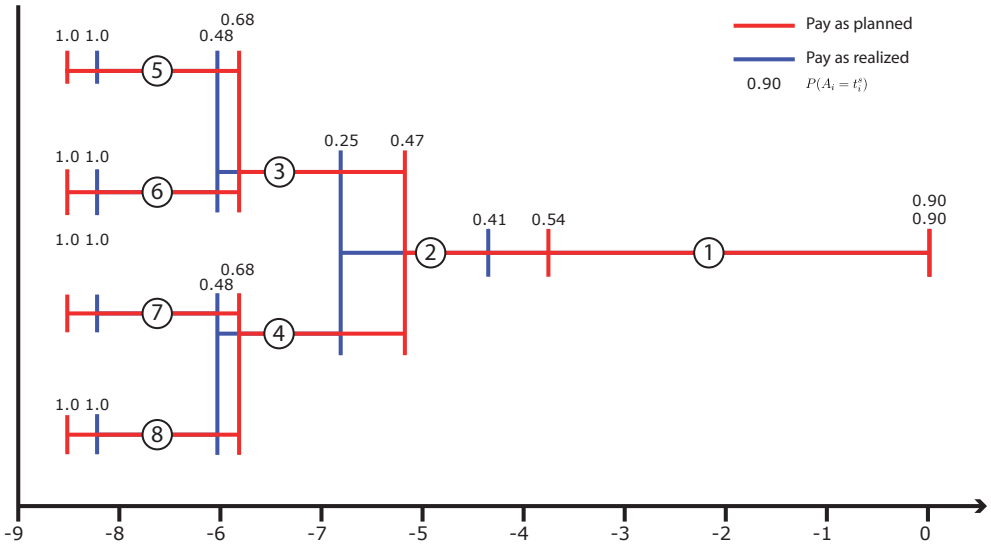


Figure 4.7: Comparison of 'pay as planned' and 'pay as realized' production plan

-8.54). than under the 'pay as realized' costing scheme ($t_8^s = -8.24$). This is in line with the result of Proposition 2.3. This proposition states that under a 'pay as planned' costing scheme production is started earlier than under the 'pay as realized' costing scheme. The proposition only concerns the assembly systems of Chapter 2 but this numerical result indicates that this structural result can be extended to more general network structures discussed in this chapter.

When looking at activities 5-8, their planned start time is earlier, but the planned finish time (or the planned start time of their successors) is also later, meaning that their is more time available for these tasks. Thus, under the pay as planned cost function, nodes 3 and 4 will start at their planned start time more often. For each planned start time, we show in the figure the probability that a node starts at its planned start time. Thus in this example node 3 and 4 start at their planned start time with probability 0.48 under the 'pay as realized' plan and with probability 0.68 under the 'pay as planned' plan. Nodes 3 and 4 get more time under the 'pay as planned' plan : $t^s_{\mathcal{S}(3)} - t^s_3 = 1.57$ vs 1.22 under the 'pay as realized' plan. Because of this, and the higher the probability starts at its planned start time the planned start time of node 2 is more often met. The probability that this node starts at its planned start time increased from 0.25 to 0.47.

Due to the fact that node 2 can start on time more often, it needs less time to compensate for delays of predecessors and therefore the planned start time of node 1 is met more often as well. Finally both cost functions have the same probability that the final deadline is met, $P(A_0 = 0) = 0.900$, a necessary requirement for an optimal plan for both cost functions.

In general, under the 'pay as planned' cost function the intermediate deadlines are more often met. On the other hand, this also means that the total planned leadtime ($t_0^s - \min_{i \in \mathcal{R}} t_i^s$) increases compared to the 'pay as realized' cost function. Thus, the 'pay as planned' costing scheme should be used when intermediate deadlines are important, rather than a short total planned leadtime. The 'pay as realized costing scheme' should be used when intermediate deadlines are flexible and a short total planned leadtime is preferred.

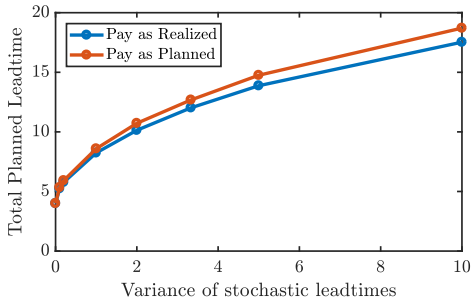
4.6.3 Variance of Stochastic Leadtimes

In the final numerical experiment, we study the effect of the variance of the leadtime distributions on the optimal production plan. Instead of using exponential distributions we now use Gamma distributions. The reason to do this is that by using the gamma distributions we can change the variance while keeping the mean constant. Gamma distributions have two parameters, a and b, which together define the mean μ and variance σ^2 as follows:

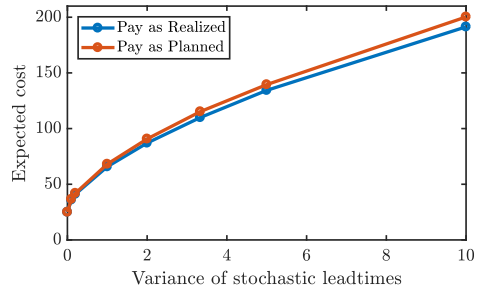
$$\begin{aligned} \mu &= ab \\ \sigma^2 &= ab^2 \end{aligned}$$

We choose the values of a and b such that the mean is kept constant at $\mu = 1$ and the variance is changed: $0.1 \leq \sigma^2 \leq 10$. The results are shown in Figure 4.8.

Figure 4.8a shows the total planned leadtime, defined as $0 - t_8^s$. For both cost functions the total planned leadtime increases when the variance of stochastic leadtimes increases. In line with Proposition 2.3 the total planned leadtime under the 'pay as planned' cost function is always higher than under the 'pay as realized' cost function. The only exception is the case when the variance $\sigma^2 = 0$. This is a deterministic leadtime of 1 for each node. This results in a total leadtime of 4 for the system, since the longest paths consists of 4 nodes.



(a) Total planned leadtime



(b) Expected Cost

Figure 4.8: Expected cost and planned leadtime when increasing the variance of leadtime distributions.

In Figure 4.8b the expected cost are shown for both cost functions. Besides the stochastic leadtimes, we also add the case of deterministic leadtimes with $\sigma^2 = 0$. The cost of such a system are computed via (4.10) and equal 25. The cost of the system are increasing in a similar way as the planned leadtimes. When the variance reaches 10, the cost are roughly 8 times as high compared to the deterministic case. We also see that the 'pay as planned' costing scheme always leads to slightly higher cost. This is in line with Proposition 2.2.

4.7. Conclusions & Future Research Directions

In this paper, we performed an exact analysis of the planned leadtime problem for a general class of networks. This problem is present during the production of capital intensive, customer specific and complicated products. Production is split up in multiple activities that each add value to the final product and have a stochastic duration. We model the network as a directed acyclic graph with one end node. Compared to previous work, this is the most general class of networks, most papers available in literature study only a sub-class. We develop production plans that consist of planned start times for each node. We don't prescribe a finish time, except for the final deadline. Using planned start times allows us to formulate the expected cost of the system in a tractable way. Furthermore, we introduce a concept called critical tardy paths, which gives insight which activities in the network cause lateness. For this general class of systems, we formulate optimality

equations and show that at optimality, the probability of a critical path from a certain node is proportional to the value it adds to the final product. Furthermore, we prove properties of the optimal solution that can be used to develop efficient approximation algorithms to solve this problem.

4.A. Proofs

Proof of Lemma 4.1

Since G is a directed acyclic graph we can unfold the recursive definitions of (4.1), (4.2) and (4.3). Then $A_i(\omega)$ can be written as

$$A_i(\omega) = \max\left\{ t_i^s, \max_{l \in \mathcal{Y}(i)} \left\{ t_l^s + \sum_{m \in w_{l,i} \setminus \{i\}} T_m(\omega) \right\} \right\}.$$

Since leadtimes are independent, it follows directly that A_i only depends on $T_j,\ j \in \mathcal{Y}(i)$ and $t_j^s, j \in \mathcal{Y}(i) \cup \{i\}$.

Proof of Lemma 4.2

Since G is a directed acyclic graph we can unfold the recursive definitions of (4.1), (4.2) and (4.3). Then $B_i(\omega)$ can be written as

$$B_i(\omega) = \max_{l \in \mathcal{Y}(i) \cup \{i\}} \left\{ t_l^s + \sum_{m \in w_{l,i}} T_m(\omega) \right\}.$$

Since leadtimes are independent, it follows directly that B_i only depends on $T_j,\ j \in \mathcal{Y}(i) \cup \{i\}$ and $t_j^s, j \in \mathcal{Y}(i) \cup \{i\}$.

Proof of Lemma 4.3

According to Definition 4.2, the event $\theta_{i,j}^{(i)}$ depends on $A_k^{(i)},\ k \in w_{i,j} \cup \{s(j)\}$ and $B_k^{(i)},\ k \in w_{i,j}$. Applying Lemmas 4.1 and 4.2 on the subgraph and using the

assumption that leadtimes are independent, it follows that $\theta_{i,j}^{(i)}$ only depends on $T_k,\ k \in \mathcal{Y}(s(j)) \cap \mathcal{V}^{(i)}$.

Proof of Lemma 4.4

Suppose $\omega \in \theta_{i,j}^{(i)}$. Consider a node k on the path $w_{i,j}$. From Definition 4.2 it then follows that $A_i^{(i)} = t_i^s, A_{s(l)}^{(i)}(\omega) = B_l^{(i)}(\omega), l \in w_{i,k}$. Hence, $\omega \in \theta_{i,k}^{(i)}$ as well.

Proof of Lemma 4.5

Suppose we have a solution $\mathbf{t^s}$ containing a reversed pair $t_i^s > t_{s(i)}^s$. In such a solution, node i will always start after $t_{s(i)}^s$ and hence (4.3) reduces to

$$A_{s(i)}(\omega) = \max_{j \in \mathcal{P}(s(i))} \{B_j(\omega)\}, \quad \omega \in \Omega. \tag{4.15}$$

Due to the recursive definition of actual start times, actual start times of nodes on the path $w_{s(i),1}$ also do not depend on $t_{s(i)}^s$. We now define the solution $\mathbf{t^{s\prime}}$:

$$\begin{aligned} t_{s(i)}^{s\prime} &= t_i^s \\ t_k^{s\prime} &= t_k^s \quad k \in \mathcal{V} \backslash \{s(i)\} \end{aligned}$$

Note that $\mathbf{t^{s\prime}} \in \mathbb{D}$. By definition $T_i(\omega) > 0$ and hence it follows that $A'_{s(i)} = A_{s(i)}$. Because the actual start times do not change, the cost also do not change and hence we find $C(\mathbf{t^s}) = C(\mathbf{t^{s\prime}})$.

Proof of Lemma 4.6

The proof is a standard Newsvendor proof. We start by taking the partial derivative of (4.8) w.r.t. t_0^s. From Lemma 4.1 it follows that besides t_0^s, A_0 is the only term in this cost function that depends on t_0^s and can be written as $A_0 = \max\{B_1, t_1^s\}$. This term has the following derivatives:

$$\frac{\partial}{\partial t_0^s} A_0 = \begin{cases} 1 & \text{if } B_1 < t_0^s \\ 0 & \text{otherwise.} \end{cases}$$

Then, the derivative of the cost function becomes

$$\frac{\partial}{\partial t_0^s} C(\mathbf{t^s}) = (h_1^c + p)\mathbf{1}_{\{B_1 < t_0^s\}} - p.$$

Taking the expectation and equating the derivative to 0 yields

$$(h_1^c + p)P(B_1 < t_0^s) - p = 0.$$

Since we defined $t_0 = 0$ this equation can be rewritten to

$$P(A_0 > 0) = \frac{h_1^c}{h_1^c + p}.$$

This concludes the proof.

Proof of Lemma 4.7

From Definition 4.2 it follows that for an event $\theta_{i,1}^{(i)}$ it is required that $A_{s(k)}^{(i)}(\omega) = B_k^{(i)}(\omega)$ for all nodes on the path $w_{i,1}$. This also implies that $A_{s(k)}^{(i)}(\omega) > B_l^{(i)}(\omega)$ for all $l \in \mathcal{P}(s(k))\backslash\{k\}$. This violates the definition of $\theta_{m,1}^{(i)}$ for all $m \in \mathcal{V}^{(i)}\backslash\{i\}$.

Proof of Theorem 4.1

We prove Theorem 4.1 by taking the partial derivative of $\mathbb{E}[C(\mathbf{t^s})]$ w.r.t. planned start time t_i^s. At optimality this derivative should equal zero. To find the derivative we first show that

$$\frac{\partial}{\partial t_i^s}\mathbb{E}[C(\mathbf{t^s})] = \mathbb{E}\left[\frac{\partial}{\partial t_i^s} C(\mathbf{t^s})\right]$$

or equivalently, that for every sequence $g_1, g_2, \ldots$ converging to 0,

$$\lim_{n\to\infty} \frac{\mathbb{E}[C(\mathbf{t^s} + g_n\mathbf{e_i})] - \mathbb{E}[C(\mathbf{t^s})]}{g_n} = \mathbb{E}\left[\frac{\partial C(\mathbf{t^s})}{\partial t_i^s}\right] \tag{4.16}$$

where $\mathbf{e_i}$ is the unit vector with 1 at position i. To find the derivative, we rewrite (4.8) as follows:

$$C(\mathbf{t^s}) = \sum_{k \in \mathcal{V}} h_k(A_0 - A_k) + pA_0. \tag{4.17}$$

Now consider the stochastic terms $A_k,\ k = 0, 1, ..., N$ which are all functions of $\mathbf{t^s}$. For every realization $\omega = (t_1, \ldots, t_N)$ we have

$$A_k(\omega) = \max\left\{ t_k^s, \max_{l \in \mathcal{Y}(k)} \left\{ t_l^s + \sum_{m \in w_{l,k} \setminus \{k\}} T_m(\omega) \right\} \right\}. \tag{4.18}$$

Hence for $k \in w_{i,1} \setminus \{i\}$ the derivative $\frac{\partial A_k(\mathbf{t^s})(\omega)}{\partial t_i^s}$ exists for almost all ω and

$$\frac{\partial A_k(\mathbf{t^s})(\omega)}{\partial t_i^s} = \begin{cases} 1, & \text{if } \omega \in \theta_{i,\mathcal{P}(k) \cap \{w_{i,k}\}} \\ 0, & \text{otherwise} \end{cases}$$

If $k = i$, then

$$\frac{\partial A_i(\mathbf{t^s})(\omega)}{\partial t_i^s} = \begin{cases} 1, & \text{if } A_i(\omega) = t_i^s, \\ 0, & \text{otherwise.} \end{cases}$$

From Lemma 4.1 it follows that A_k does not depend t_i^s if $k \notin w_{i,1}$. Hence $\frac{\partial}{\partial t_i^s} A_k(\omega) = 0$ for all $k \notin w_{i,1}$. We have now determined the partial derivatives w.r.t. t_i^s of all actual start times A_k in (4.17). In this equation we multiply these start times by the cost parameters h_i and p. Hence, for almost all ω,

$$\left| \frac{C(\mathbf{t^s} + g_n \mathbf{e_i})(\omega) - C(\mathbf{t^s})(\omega)}{g_n} \right| \leq h_1^c + p,$$

so by bounded convergence we can conclude that (4.16) holds. Combining the results we find that

$$\frac{\partial \mathbb{E}[C(\mathbf{t^s})]}{\partial t_i^s} = \mathbb{E}\left[\frac{\partial C(\mathbf{t^s})}{\partial t_i^s} \right] = (h_1^c + p) P(\theta_{i,1}) - h_i P(A_i = t_i^s) - \sum_{k \in w_{i,1} \setminus \{i\}} h_k P(\theta_{i,\mathcal{P}(k) \cap w_{i,k}})$$

At optimality, $\frac{\partial \mathbb{E}[C(\mathbf{t^s})]}{\partial t_i^s} = 0$. Also, we can use (4.7) to rewrite $P(\theta_{i,1})$ as $P(\theta_{i,1}^{(i)})P(A_i = t_i^s)$. This yields

$$(h_1^c + p)P(\theta_{i,1}^{(i)})P(A_i = t_i^s) - h_i P(A_i = t_i^s) - \sum_{k \in w_{i,1} \setminus \{1\}} h_{s(k)} P(\theta_{i,k}^{(i)}) P(A_i = t_i^s) = 0. \tag{4.19}$$

Note that we changed the summation range from $w_{i,1}\setminus\{i\}$ to $w_{i,1}\setminus\{1\}$. Replacing k by $s(k)$ and $\mathcal{P}(k) \cap w_{i,k}$ by k then yields the same term. In the domain $\mathbb{D}$ we know that $t^s_{s(i)} > t_i^s$ and thus it holds that $P(A_i = t_i^s) > 0$. Thus, we can divide by $P(A_i = t_i^s)$ and find

$$(h_1^c + p)P(\theta_{i,1}^{(i)}) - h_i - \sum_{k \in w_{i,1} \setminus \{1\}} h_{s(k)} P(\theta_{i,j}^{(i)}) = 0.$$

Finally, we rewrite the equation to

$$P(\theta_{i,1}^{(i)}) = \frac{h_i}{h_1^c + p} + \sum_{k \in w_{i,1} \setminus \{1\}} \frac{h_{s(k)} P(\theta_{i,j}^{(i)})}{(h_1^c + p)}, \quad i \in \mathcal{V}.$$

This concludes the proof.

Proof of Lemma 4.8

1. The event $\theta_{i,1}^{(i)}$ is defined on the subgraph $V^{(i)}$, which excludes the nodes in $\mathcal{Y}(i)$. Hence, the probability $P(\theta_{i,1}^{(i)})$ does not depend on t_k^s. By definition, $\alpha_i \subseteq \theta_{i,1}^{(i)}$ and thus $P(\alpha_i)$ does not depend on t_k^s either.

2. $P(\theta_{i,1}^{(i)})$ is strictly increasing in t_i^s, because in the subsystem i is a root node. In the case of an event $\theta_{i,1}^{(i)}$, from Definition 4.3 it follows that there is exactly one critical tardy path, starting at a node on the path $w_{i,1}$. In other words, $\theta_{i,1}^{(i)} \subseteq \bigcup_{j \in w_{i,1}} \alpha_j$. However in case 1 of the lemma we showed that the probabilities $P(\alpha_j)$, $j \in w_{i,1}\setminus\{i\}$ do not depend on t_i^s because $i \notin V^{(j)}$. Thus, $P(\alpha_i)$ is strictly increasing in t_i^s.

3. When $t_j^s < \max_{l \in \mathcal{P}(j)}\{t_l^s\}$, t_j^s does not have an effect on $\theta_{i,1}^{(i)}$. When $t_j^s > \max_{l \in \mathcal{P}(j)}\{t_l^s\}$ t_j^s there will be some ω that were first in $\theta_{i,1}^{(i)}$ that now become

part of $\theta_{j,1}^{(j)}$.

Proof of Lemma 4.10

Consider an alternative solution $\mathbf{t}^{\mathbf{s}\prime} \in \mathbb{R}^N$, $\mathbf{t}^{\mathbf{s}\prime} \neq \mathbf{t}^{\mathbf{s}*}$. We define a set that contains all nodes for which the planned start time in the alternative solution is increased compared to their planned start time in the solution $\mathbf{t}^{\mathbf{s}*}$:

$$\mathcal{J} = \{j : t_j^{s\prime} > t_j^{s*}\}.$$

Similarly, we define a set for all nodes that have a decreased planned start time: start time:

$$\mathcal{K} = \{k : t_k^{s\prime} < t_k^{s*}\} :$$

Since $\mathbf{t}^{\mathbf{s}\prime} \neq \mathbf{t}^{\mathbf{s}*}$, $\mathcal{J}$ and/or $\mathcal{K}$ is/are nonempty. Suppose J is nonempty. Then, from Lemma 4.9 it follows that

$$\sum_{j \in J} P(\alpha_j') > \sum_{j \in J} P(\alpha_j^*). \tag{4.20}$$

This is due to the fact that all planned start times of nodes in the summation have increased, leading to an increase of the summed probability. All planned start times of nodes not in the sum remained unchanged or decreased. The nodes that decreased can cause another increase in the sum of tardy path probabilities. As $\mathbf{t}^{\mathbf{s}*}$ satisfied Theorem 4.2, it holds that

$$\sum_{j \in \mathcal{J}} P(\alpha_j^*) = \sum_{j \in \mathcal{J}} \frac{h_j}{h_1^c + p}. \tag{4.21}$$

Combining (4.20) and (4.21) yields

$$\sum_{j \in \mathcal{J}} P(\alpha_j') > \sum_{j \in \mathcal{J}} \frac{h_j}{h_1^c + p}.$$

Similarly, if $\mathcal{K}$ is nonempty, it follows that

$$\sum_{k \in \mathcal{K}} P(\alpha_K') < \sum_{k \in \mathcal{K}} \frac{h_k}{h_1^c + p}.$$

We conclude that $\mathbf{t}^{\mathbf{s}\prime}$ does not satisfy the system of Newsvendor equations and that $\mathbf{t}^{\mathbf{s}*}$ is a unique solution.

Proof of Theorem 4.2

By induction. We have to show for all $i \in \mathcal{V}$ that

$$P(\alpha_i) = \frac{h_i}{h_1^c + p}.$$

In the case $i = 1$ in (4.13) the summation term vanishes and it directly follows that

$$P(\alpha_1) = \frac{h_1}{h_1^c + p}$$

Now suppose we have shown for all nodes $k < i$ that

$$P(\alpha_k) = \frac{h_k}{h_1^c + p}.$$

By inserting the induction hypothesis, we rewrite (4.13) for node i to

$$P(\theta_{i,1}^{(i)}) = \frac{h_i}{h_1^c + p} + \sum_{j \in w_{i,1} \setminus \{1\}} P(\alpha_{s(j)}) P(\theta_{i,j}^{(i)}).$$

By definition $\alpha_{s(j)}$ is a subset of $\theta_{s(j),1}^{(i)}$. From Lemma 4.4 it follows $\theta_{s(j),1}^{(i)} \cap \theta_{i,j}^{(i)} = \emptyset$. Hence $\alpha_{s(j)} \cap \theta_{i,j}^{(i)} = \emptyset$ as well. Using the assumption that stochastic leadtimes of different nodes are independent it follows that $P(\alpha_{s(j)}) P(\theta_{i,j}^{(i)}) = P(\alpha_{s(j)}, \theta_{i,j}^{(i)})$. Hence we can write

$$P(\theta_{i,1}^{(i)}) = \frac{h_i}{h_1^c + p} + \sum_{j \in w_{i,1} \setminus \{1\}} P(\alpha_{s(j)}, \theta_{i,j}^{(i)}).$$

Next, using Definition 4.3 and Corollary 4.1 we can separate the term $P(\theta_{i,1}^{(i)})$ on the left-hand side into a sum of mutual exclusive probabilities. Furthermore, on the right-hand side we change the notation of the summation over nodes on the path $w_{i,1}$. Instead of having a summation over the set of nodes $w_{i,1} \setminus \{1\}$, we sum over the set of nodes $w_{i,1} \setminus \{i\}$. By also changing $\alpha_{s(j)}$ to α_j, we still have a summation

over the same nodes.

$$P(\alpha_i) + \sum_{j \in w_{i,1} \setminus \{i\}} P(\alpha_j, \theta_{i,1}^{(i)}) = \frac{h_i}{h_1^c + p} + \sum_{j \in w_{i,1} \setminus \{i\}} P(\alpha_j, \theta_{i,1}^{(i)})$$

Finally the summation terms cancel out and we find the following Newsvendor equation for node i:

$$P(\alpha_i) = \frac{h_i}{h_1^c + p}.$$

By induction we now conclude that

$$P(\alpha_i) = \frac{h_i}{h_1^c + p}, \quad i \in \mathcal{V}.$$

This completes the proof.

5

Capacity-constrained Project Portfolio Selection

5.1. Introduction

Innovation in healthcare is crucial for ensuring a healthy population in the future (Dai and Tayur, 2018). The development of new drugs is such an innovation. Pharmaceutical companies, universities and governments have the task to discover new compounds and develop them into drugs that are both efficacious and safe to use for humans. Drug development is a long and costly process with an uncertain outcome. The cost of drug development has been increasing in recent years (DiMasi et al., 2016). However, only a few drugs are developed successfully, resulting in market approval. The vast majority of drug development projects is terminated early in the process, due to test results indicating insufficient efficacy or serious side effects. In recent years, the percentage of compounds reaching market approval has decreased (Pammolli et al., 2011).

Pharmaceutical companies aim for a portfolio of projects that maintains long term profits of the company. Managing the drug development portfolio is a core task. A constraint that affects the portfolio management decisions is the pharmaceutical companies' research capacity. Research capacity can, for example, be lab facilities,

scientists, doctors or test patients. Capacity decisions are long term decisions, which are made under uncertainty. Given its importance and uncertainty surrounding it, researchers have developed several models to assist the pharmaceutical companies with their capacity decisions (Huchzermeier and Loch, 2001; Girotra et al., 2007).

To be able to deal with the uncertain nature of the drug development process, outsourcing is an option that is widely used. Instead of carrying out the complete process in house, pharmaceutical companies use Contract Research Organizations (CROs) to execute parts of the drug development (Mirowski and Van Horn, 2005). Examples of services provided by CROs include the recruitment of test patients, setting up clinical trials and dealing with regulatory bodies. CROs execute clinical trials on a contract basis. In general, they get paid for the work they do, regardless of the outcome of it. To cope with increasing uncertainties and to reduce costs, pharmaceutical companies increasingly rely on outsourcing. Howells et al. (2008) observe that outsourcing is especially useful for standardized non-core activities. The execution of clinical trials is an example of such an activity. This results in CRO market growth (Zion Market Research, 2015).

CROs have a large world-wide network of resources, such as patients, doctors, nurses and medical facilities. These resources are used to offer services to pharmaceutical companies. The resources together form the capacity for a CRO to do research. Since CROs work with many different pharmaceutical companies, they try to utilize this capacity as best as possible.

Although there are many types of collaborations between pharmaceutical companies and CROs, CROs typically work on a *fee-for-service contract*. According to this contract a CRO receives a reward for the service it delivers. The reward is agreed on beforehand and does not depend on the results of the service. If a CRO conducts a study that results in a the drug development process being terminated, the CRO still receives its rewards. The fee-for-service type of contracts imply different tradeoffs for CROs compared to pharmaceutical companies. First of all, for a pharmaceutical company the cost of developing a drug includes the payments for a CRO, while for a CRO this is the main source of income. A failed drug development project, i.e., a project that does not result in market approval, is still a source of income for CRO. Secondly, the revenue for the pharmaceutical company depends on the potential sales an approved drug can generate. Therefore, this is an important factor to be considered when making development decisions. For a CRO potential sales are

irrelevant, i.e., the payment received from the pharmaceutical company does not depend on the sales. Finally, the pharmaceutical company is the project owner and, hence, takes go/no go decisions, which for a CRO are exogenous decisions.

Several studies consider the capacity problem of pharmaceutical companies and develop models to plan the drug development process (Colvin and Maravelias, 2008, 2010). These models do not apply to the CROs' capacity problems since CROs face different tradeoffs. In this chapter, we develop models from the CROs' point of view. We formulate mathematical models that assist CROs with two capacity related decisions. The first decision considers *the amount of research capacity* a CRO should have. Research capacity is very specific and cannot easily be scaled up or down. On one hand it is key to have sufficient capacity to be able to execute large research projects, but on the other hand, if capacity is not utilized it directly affects the CROs profits. The second decision we aim to model is *the project acceptance decision*. Pharmaceutical companies can have projects with different characteristics in terms of capacity requirements, success probabilities and rewards. In order to best utilize its research capacity, a CRO selects projects that are profitable and fit within the available capacity.

The CROs decisions on the amount of research capacity and project selection are not straightforward due to uncertainties around the drug development process. Projects consist of multiple phases. When a CRO accepts a project, it aims to execute all the phases. However, the project might be terminated by a pharmaceutical company. This outcome is not desired by either of the CRO or the pharmaceutical company. For the CRO it is useful to complete the project since this can establish a better relationship with the pharmaceutical company. For the pharmaceutical company, it is useful to complete the project since switching between CROs for different phases of the project is costly and results in delays in development. Although not desired, there is a possibility that the project is terminated after a certain number of phases. This is an uncertainty that the CROs should take into consideration when making decisions on research capacity and project selection.

Projects and their phases can have different characteristics. Each phase might have a different capacity requirement, which depends on the duration and other aspects of the project. In addition, phases might have different success likelihoods. These differences imply another uncertainty for the CRO, since the decision on project selection should take the capacity requirements and success probabilities

into account. The CRO has to make this decision without knowing what project offers it will receive in the future.

In this chapter, we consider a CRO and propose a Markov Decision Process (MDP) formulation to assist the CRO with its research capacity and project selection decisions. Our model aims to maximize the CRO's average periodic profit by optimizing its research capacity and project acceptance/rejection decision. The state of the MDP is determined by the projects that are currently executed and the available projects that can be selected by the CRO. The optimal policy prescribes for each state whether a new project should be accepted or not.

We analyze the structure of the MDP and solve it for different parameters. We consider a simplified setting with homogeneous project types and determined the set of potential optimal policies for the acceptance decisions. We identify the parameter settings under which each potential acceptance policy is optimal. For a more complex setup with nonhomogeneous project types, we show that instead of focusing on one project type, the CRO should apply a policy where it accepts project types in such a way that projects complement each other and that capacity is highly utilized, but not over-utilized. For the capacity decision, we show that it is not always optimal to choose a capacity such that all arriving projects can be accepted.

The remainder of this chapter is organized as follows. In Section 5.2 we review the relevant literature. In Section 5.3 we develop the general model to analyze the CRO's capacity management problem. To get an insight in the dynamics of the model, we analyze a simplified version in Section 5.4, where we consider a CRO executes only one type of projects. In Section 5.5 we extend these results to a CRO that execute multiple project types. In both these sections, we assume a fixed capacity and analyze the project selection decision. In Section 5.6 we include the capacity as a decision variable. Finally, conclusions and recommendations can be found in Section 5.7.

5.2. Literature Review

The CRO's capacity problem fits into the wide literature stream of new drug development. The viewpoint of most papers in this stream is the viewpoint of

a pharmaceutical company. Although our viewpoint fundamentally differs from the drug development literature, we benefit from it by defining similar modeling assumptions and using similar modeling techniques.

Most of the drug development literature acknowledges the uncertain outcome of a project and the possibility of an early termination as a key issue. It is vital to quantify these uncertainties as they are an important driver for the value of a project. Jacob and Kwak (2003) discuss existing techniques and develop a new method to determine the value of a drug development project using a real options approach. Girotra et al. (2007) take a different approach and analyze the value a drug has within the drug portfolio of a pharmaceutical company. Hence, they consider the fact that the value can depend on the status of the other drugs in the portfolio.

Multiple studies address the project selection decision. Huchzermeier and Loch (2001) focus on mitigating risks in the portfolio by postponing or expediting critical activities in a project. Yu and Gittins (2008) study the optimization of long-term profitability of the project selection decisions. In a follow up paper Charalambous and Gittins (2008) model the same problem as an MDP and derive optimal policies. Their main focus is on the pre-clinical trial stage, where multiple variants of drug compounds are investigated simultaneously. The value of one variant might depend on whether other variants are also under development and this can affect the optimal selection decision.

In the clinical trial phase the drug is tested in large scale experiments involving human test subjects. Especially in this phase, a large of amount of resources is needed, while these resources are usually scarce. This limited capacity is an important aspect in the project selection problem. Gatica et al. (2003) study the project selection process by taking this capacity constraint into account. In order to cope with the failure probabilities of projects, they generate different scenarios and then formulate a mixed integer linear program to find the optimal selection. Colvin and Maravelias (2008) and Colvin and Maravelias (2010) develop stochastic programming formulations to model the failure probabilities at different project phases. In Christian and Cremaschi (2015) heuristics are developed for the planning of clinical trials. All of these studies consider the discounted potential revenue of a drug and this results in a planning that gives priority to the most profitable drugs.

While all of these studies consider the selection problem from a pharmaceutical

company's perspective, Kouvelis et al. (2017) consider the problem from a CRO's point of view. This study focuses on the planning of clinical trials for a single drug. It addresses operational aspects such as as the number of facilities to open and the number of patients to enroll.

Besides literature that focuses on the application area of new drug development, there are also studies on project scheduling considering failure probabilities of activities. Henig and Simchi-Levi (1990) and De Reyck and Leus (2008) consider the option of executing activities in parallel to reduce the duration of a project. Since discounted rewards are only collected at the end, this can be a profitable option (assuming capacity is available). On the other hand, this also expedites costs of development, which can lead to higher losses in case of a failure.

We contribute to the literature on drug development by developing models from the CROs' point of view. We formulate mathematical models to assist CROs with their research capacity and project acceptance decisions.

5.3. Problem Formulation

In this section, we define the system parameters (Section 5.3.1) and introduce our Markov Decision Process formulation (Section 5.3.2).

5.3.1 Problem Definition

We consider a CRO that receives requests for drug development projects with different characteristics. We use index $n \in \{1, 2, \ldots, N\}$ to represent a project type. There exist N different project types. We consider an infinite horizon discrete time model, i.e., time is divided into periods and use index t to represent a period. The project arrival process is random. In each time period t either no project or one project arrives. The probability that a project of type n arrives is q_n. Since, in a period, at most one project arrives it holds that $\sum_{n=1}^{N} q_n + q_0 = 1$, where q_0 denotes the probability that no project arrives. We assume that this arrival process is stationary over time.

Projects consist of a certain number of phases. These phases are executed sequentially. We use m to represent the phase number. Each project consists of

M phases, i.e., $m \in \{1, 2, \ldots, M\}$. A phase denotes a part of the project that takes exactly one period. At the end of a phase a go/no go decision is taken. This decision is taken by the customer, i.e., pharmaceutical company, based on the results of the already executed phases. Besides the test results, the customer also takes other factors into account such as other drugs in the portfolio of the customer, market potential and cost of future development when making this go/no go decision. For the CRO this is an exogenous decision. We model the customer's go/no go decision by defining the parameter $p_{n,m}$, which denotes the probability that the go decision is taken after completing phase m of a project type n. This is the probability that the project continues to the next phase, i.e., phase $m+1$. With probability $1 - p_{n,m}$ the project is terminated after phase m. By definition $p_{n,M}$ is 0, since phase M is the last phase of each project.

If the customer decides to continue to the next phase, we assume this phase should be executed immediately in the next period. The motivation for this is that drug development projects are time critical, i.e., it is essential to get FDA approval as soon as possible, such that the drug can enter the market.

Each phase of the project requires a certain capacity. With capacity, we denote the resources that CROs posses. These resources can for example be laboratory facilities, hospitals, patients, nurses and doctors. For this high level model we assume that this capacity is homogeneous and can be used for all phases of all project types. We use $c_{n,m}$ to represent the capacity requirement of phase m of project type n. The CRO has a limited capacity of C per period. In each period, this capacity costs α per unit, regardless of whether it is utilized. As this is a long term decision, we assume C cannot be changed from period to period. Due to the uncertainty of go/no go decisions and arrival of new projects, the total capacity requirement for future periods of all projects in the system is uncertain. Due to this uncertainty, it is possible that in some periods the CRO needs more capacity than C. For these periods, the CRO can rely on external options and decide to buy extra capacity, for example from another CRO at cost β per unit. We have $\alpha < \beta$, otherwise it would be optimal to always use the external option. In fact the value of β is relatively high, due the specific expertise that is needed at high flexibility.

For each completed phase of a project a CRO receives a reward. $r_{n,m}$ represents the reward that the CRO receives after completion of phase m of project type n. This reward is received for the service the CRO delivers in a period and is independent

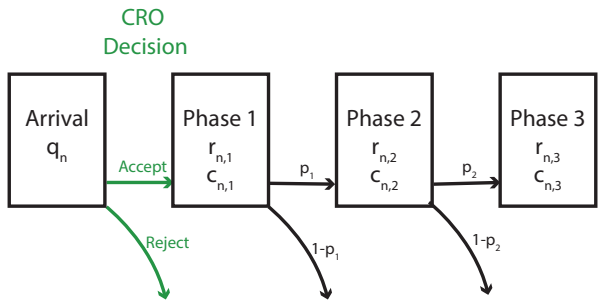


Figure 5.1: Steps in a drug development of 3 phases executed by a CRO.

of the go/no go decision that follows at the end of the period.

An overview of the phase characteristics and the CRO's acceptance decision is given in Figure 5.1.

5.3.2 Markov Decision Process Formulation

We model the behavior of the system as a Markov Decision Process. The state of the system at the beginning of a period is denoted by the vector $\vec{x} = [x_0, x_1, x_2, ..., x_M]$. The first element of the vector gives information on the project arrival process at *the beginning of the period*. This element indicates whether a project arrives and if it arrives, the type of it. If $x_0 = 0$, there is no project arrival and if $x_0 = n$, a project of type $n \in \{1, 2, \ldots, N\}$ arrives. If a project arrives and it is accepted, its first phase starts in the next period. Note that there is at most one project arrival at the beginning of every period and each phase takes exactly one period. These assumptions imply that there can be at most one project in each phase. The element x_m indicates the project type in phase m. If $x_m = 0$, there is no project in phase m. As an example, assume that the system is in state $\vec{x} = [1, 2, 0, 1]$ at the beginning of a period. This state indicates that a project of type 1 has arrived, a project of type 2 is in phase 1, there is no project in phase 2 and a project of type 1 is in phase 3.

In each period the CRO collects rewards for the project phases it executes. The total reward in a given period is given by

$$\sum_{m=1}^{M} \sum_{n=1}^{N} r_{n,m} \mathbf{1}_{\{x_m = n\}}.$$

Here, $\mathbf{1}_{\{x_m=n\}}$ is an indicator function and denotes whether a project type n is present in phase m. Given that there can be at most one project per phase, the sum $\sum_{n=1} \mathbf{1}_{\{x_m=n\}}$ equals either 0 or 1. If the total capacity requirement is above C, extra capacity is needed. The required extra capacity is given by:

$$\left(\sum_{m=1}^{M} \sum_{n=1}^{N} c_{n,m} \mathbf{1}_{\{x_m=n\}} - C \right)^+,$$

where $(x)^+ = \max\{x, 0\}$.

Combining the rewards for executing projects, the capacity usage and the per unit capacity costs α and β, we can write the expression for the net reward when the system is in state $\vec{x} = [x_0, x_1, x_2, ..., x_M]$ at the beginning of the period. We represent this reward as $R(\vec{x})$ and express it as

$$R(\vec{x}) = \sum_{m=1}^{M} \sum_{n=1}^{N} r_{n,m} \mathbf{1}_{\{x_m=n\}} - \alpha C - \beta \left(\sum_{m=1}^{M} \sum_{n=1}^{N} c_{n,m} \mathbf{1}_{\{x_m=n\}} - C \right)^+. \tag{5.1}$$

Note that $R(\vec{x})$ does not depend on the first element of the state $\vec{x}$ since the acceptance decision for the arriving project affects the state of the system starting from the beginning of the next period. Depending on the parameters and the state of the system $R(\vec{x})$ can be negative. For example, when there is no projects executed, the net reward is $-\alpha C$.

We define $\vec{R}$ as the vector containing the rewards for all states. By accepting and rejecting projects at the right moments, the CRO can affect the states that can be reached. In this way, the CRO can maximize its average net reward per period. We describe this decision of accepting and rejecting the projects by a policy $a \in A$. For each state, the policy a prescribes whether to accept or reject the project that arrives at the beginning of the period. A is the set of all possible policies. We define $\Pi(a)$ as the vector of steady state probabilities when policy a is executed. By multiplying the steady state vector, we can write the average net reward per period as

$$V(a) = \Pi(a)^T \vec{R}, \quad a \in A. \tag{5.2}$$

$\Pi(a)^T$ is the transpose of the vector $\Pi(a)$. In order to maximize its profit, the CRO needs to find the optimal total capacity C^* and the corresponding optimal policy

a^*, which maximizes the average value per period. The optimal policy for a given total capacity C is given by

$$a^* = \underset{a \in A}{\operatorname{argmax}} \{V(a)\}. \tag{5.3}$$

5.4. Homogeneous Projects

The state space of the MDP model and the number of policies increase exponentially with the number of project types and the number of phases per project. To get insights on the optimal decisions, we initially solve our model with several simplifying assumptions. First of all, we consider a deterministic arrival process with only one project type with 3 phases. In each period a project arrives with probability $q_1 = 1$. Since this is the only project type, we omit the index n for the project characteristics. Our simplifying assumptions results in a state space with $2^3 = 8$ phases. We have 3 phases and each phase has either 1 project or no project. The state vector $\vec{x}$ has 4 entries with $x(1) = 1$ for all states, since $q_1 = 1$. For example the state $\vec{x} = [1, 0, 0, 0]$ is the state where a new project arrived, but no projects are executed. As an example, this state space is shown in Figure 5.2. The transition probabilities in this figure correspond to a policy we define as *always accept*. When the CRO uses this policy it always accepts the arriving projects. Since arrivals are deterministic, transition probabilities only depend on the success probabilities of phase 1 and 2. Phase 3 is the last phase of a project and hence transition probabilities do not depend on p_3. For example, when the system is in state $[1, 0, 0, 1]$ there is only one project in the system and it is in phase 3. The next state is $[1, 1, 0, 0]$ with probability 1 since a new project is accepted, the project accepted in the previous period moves to phase 1 and the project in phase 3 project leaves the system.

In each state of the model, we can either accept or reject a new customer. With a total of 8 states, this leads to $2^8 = 256$ different policies. Before determining the optimal policy, we show that in order to find the optimal policy it suffices to consider only a subset of these 256 policies.

Lemma 5.1 *A policy $a^* \in A$ that maximizes* (5.3) *is a policy that has the same action for each pair of states $[1, i, j, 0]$ and $[1, i, j, 1]$ with $i, j \in \{0, 1\}$.*

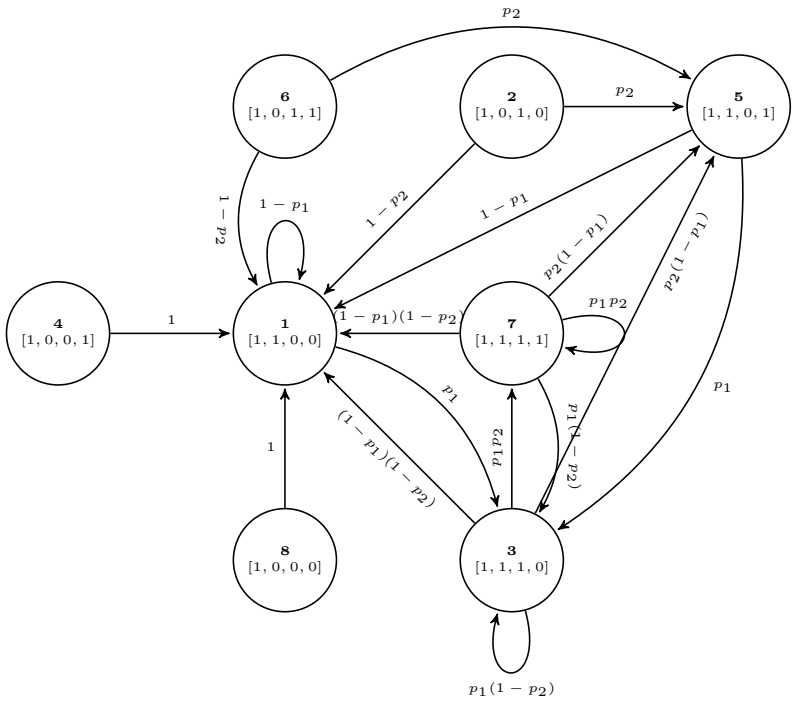


Figure 5.2: State space of homogeneous customer model under 'always accept' policy.

This lemma states that, for the optimal action, it does not matter whether there is currently a project in phase 3. The lemma can be easily proved by verifying that each pair of states have the same outgoing transition probabilities to the same states. As a result of Lemma 5.1, to characterize the optimal policy, we only need to consider $2^4 = 16$ different policies. Out of these 16 policies, we can limit ourselves to the 6 policies shown in Table 5.1. The reason for this is that the other 10 policies lead to a steady state distribution that is identical to one of the policies in Table 5.1.

Lemma 5.2 *The optimal policy to* (5.3) *is one of the policies in Table 5.1.*

Lemma 5.2 does not require any restrictions on cost and capacity parameters. It provides an important result since we know that, without any information on cost parameters, capacity parameters and success rates, a policy in Table 5.1 is optimal. Furthermore, the result helps to derive useful analytic bounds. The first bound

State	$a=1$	$a=2$	$a=3$	$a=4$	$a=5$	$a=6$
[1,0,0,0],[1,0,0,1]	0	1	1	1	1	1
[1,1,0,0],[1,1,0,1]	0	0	1	0	1	1
[1,0,1,0],[1,0,1,1]	0	0	0	1	1	1
[1,1,1,0],[1,1,1,1]	0	0	0	0	0	1

Table 5.1: Candidate optimal policies. Each column represents a policy with id a. A 0 represents a reject and 1 an accept action.

relates to the required parameter setting for a CRO to be profitable. For the CRO it should be possible to attain an average profit, otherwise it is better to quit the business. In order to achieve this it should be more profitable to accept 1 project at the time instead of accepting no projects at all. In Table 5.1, $a = 1$ is the policy that accepts no projects at all and $a = 2$ is the policy that ensures there is at most 1 project in the system. To achieve this, it only accepts when the system is empty or when there is only a project in phase 3. It is straightforward to verify that $V(1) = -\alpha C$, which is a loss for the CRO. For the CRO to be profitable it should hold that $V(2) > 0$. In Lemma 5.3 we provide the condition that needs to be satisfied to ensure a positive net reward.

Lemma 5.3 *The CRO can attain a positive net reward only if*

$$\frac{r_1 - \beta(c_1 - C)^+ + p_1(r_2 - \beta(c_2 - C)^+) + p_1 p_2(r_3 - \beta(c_3 - C)^+)}{p_1 + 2} - \alpha C \geq 0.$$

Another simple policy is to always accept all projects. This is policy $a = 6$ in Table 5.1. Under this policy, the CRO on average receives a reward of $r_1 + p_2 r_2 + r_3 p_1 p_2$ per period. It is easily verified that when $c_1 + c_2 + c_3 \leq C$ it is optimal to always accept. However, it is possible to find a bound that is tighter. To find this bound we compare policy $a = 6$ with policy $a = 5$. Under this policy, the CRO almost always accepts, except for when it is in state $[1, 1, 1, 0]$. By doing this, it is ensured that the state $[1, 1, 1, 1]$ is never reached. This state contains 3 projects, which can lead to high cost of outsourcing when there is not enough fixed capacity. When $V(6) > V(5)$ it is optimal to always accept new projects. Rewriting this inequality leads to the following bound.

Lemma 5.4 *For a CRO it is optimal to accept all projects when*

$$R(5)(-p_2p_1^2+p_2p_1)+R(3)(-p_2p_1^2+p_1)-R(1)(p_1+p_1p_2-p_1^2p_2-1)+p_1^2p_2R(7)+ \frac{R(1)(p_2p_1^2-1)}{2p_1+1}-\frac{R(4)(-p_2p_1^2+p_2p_1)}{2p_1+1}-\frac{p_1R(3)}{2*p1+1}-\frac{p_1^2p_2R(5)}{2p_1+1}- \frac{p_1^2p_2R(6)}{2p_1+1}+\frac{p_1^2R(2)(p_2-1)}{2p_1+1}-\alpha C\geq 0 \quad (5.4)$$

Here, $\vec{R} = [R(i)]_{i=1}^{6}$ denotes the net reward (rewards minus outsourcing cost) in each state:

$$\vec{R} = \begin{pmatrix} r_1-\beta(c_1-C)^+ \\ r_2-\beta(c_2-C)^+ \\ r_1+r_2-\beta(c_1+c_2-C)^+ \\ r_3-\beta(c_3-C)^+ \\ r_1+r_3-\beta(c_1+c_3-C)^+ \\ r_2+r_3-\beta(c_2+c_3-C)^+ \\ r_1+r_2+r_3-\beta(c_1+c_2+c_3-C)^+ \\ 0 \end{pmatrix}$$

The two remaining policies are $a = 3$ and $a = 4$. Policy $a = 3$ accepts only new projects when there is no project in phase 2. As a result, the state where there are are projects in both phase 1 and phase 3 can not be reached. This policy is typically useful when $c_1 + c_3 >> C$. Policy $a = 4$ accepts only new projects when phase 1 is empty. As a result, the states where there are projects in phases 1 and 2 and the states where with projects in phases 2 and 3 cannot be reached. This is typically useful when phase 2 requires a lot of capacity.

The main factor that prevents the CRO from accepting all projects is the usage of external capacity. Whether this external capacity is needed heavily depends on the go/no go probabilities of phase 1 and 2. To show the effect of these probabilities, we consider an example of a CRO that executes clinical trials for a pharmaceutical company but has limited capacity. A clinical trial usually consists of 3 phases. In each phase the required capacity increases. The capacity parameters are $c_1 = 3$, $c_2 = 12$, $c_3 = 35$, $C = 35$. Rewards are equal to the required capacity: $r_1 = c_1$,

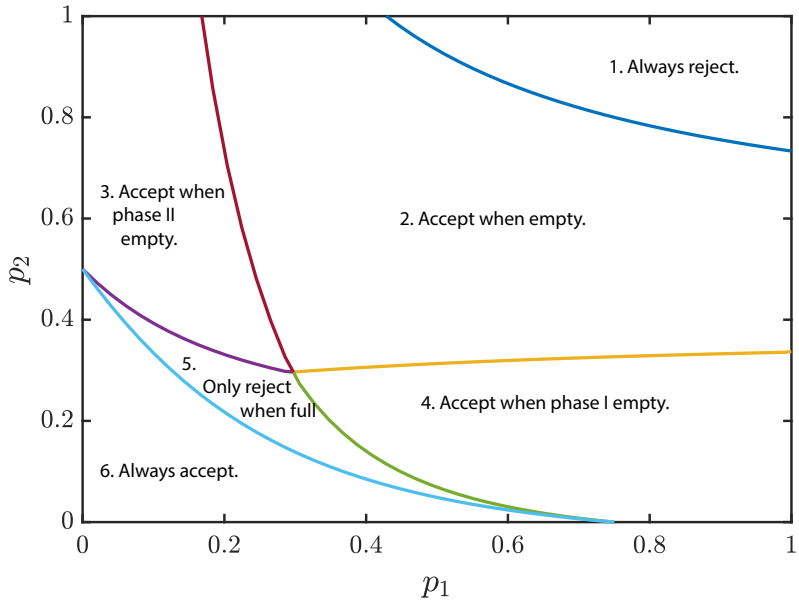


Figure 5.3: Optimal acceptance policies for project arrivals with different go/no-go probabilities.

$r_2 = c_2$, $r_3 = c_3$. The available capacity $C = 15$ is only enough to execute phases 1 and 2 simultaneously. Execution of phase 3 always requires extra capacity and a unit of extra capacity costs $\beta = 3$. Since in this section we only consider instances with the same initial capacity, the term αC is a constant. Therefore, we set $\alpha = 0$. Figure 5.3 shows the optimal policy for any combination of p_1 and p_2.

Policy 1 (always reject) is optimal in the right top corner, when both success probabilities are high. When accepting a project, the probability that it reaches phase 3 is high, which results in a loss, due to high penalty costs and low fixed capacity.

The area where policy 1 is optimal is bounded by the area where policy 2 is optimal. When success probabilities decrease, it becomes profitable to accept projects, but in such a way that there is at most 1 project at the time in the system. Note that the area of policy 1 is completely bounded by policy 2 which is the bound in Lemma 5.3.

When reducing the success probabilities further, one can either reach the region where policy 3 is optimal or the region where policy 4 is optimal. Policy 3 (accept

when phase 2 is empty) is typically optimal when p_2 is high and p_1 is low. A high p_2 results in a high probability of reaching phase 3 when a project is already in phase 2. By accepting only when phase 2 is empty, the policy avoids having multiple projects in the system when there is a project in phase 3. Accepting a project in phase 1 is not a big problem here, since p_1 is low and thus the project will most likely fail after phase 1.

Policy 4 (accept when phase 1 empty) is optimal when p_1 is high and p_2 medium. This policy reduces the chance of having 2 projects in phase 1 and 2 or projects in phase 2 and 3 simultaneously. Having projects in phases 2 and 3 is very costly and hence undesired. Having projects both in phase 1 and 2 with a high p_1 and a medium p_2 results in a high probability of having phase 2 and 3 occupied in the next period.

Policy 5 only rejects when both phase 1 and phase 2 are occupied and accepts everywhere else. This is optimal for low p_2, since low p_2 ensures that a state where both phases 2 and 3 are occupied is rarely reached.

Policy 6 (always accept) is optimal for low values of both p_1 and p_2. When p_2 is low, it is still optimal to always accept even when p_1 is high. Only states with an occupied phase 3 are costly, but due to the low p_2 these states are never reached. Note that the region where policy 6 is optimal is completely bounded by the region where policy 5 is optimal and the corresponding expression for the bound is in Lemma 5.4.

Figure 5.3 shows the optimal policy for each combination of success probabilities. It does not tell what the expected reward is under the optimal policy. In Figure 5.4 we show the expected reward under the optimal policy. We show the reward as function of p_1. The consider very low success probability and a very high success probability for phase 2, i.e., $p_2 = 0.1$ and $p_2 = 0.8$, respectively. Note that both lines start at $V = 3$ when $p_1 = 0$, since the value of p_2 does not matter when all projects fail after phase 1.

When increasing p_1, the expected reward decreases when p_2 is high, but increases when p_2 is low. The CRO benefits from the increase in p_1, since it means it can also execute phase 2. When p_2 is low, it does not need to execute phase 3, which is very costly. However, when it has to execute phase 3, the profit decreases. Here, we see a clear difference in incentives for the pharmaceutical company and the CRO.

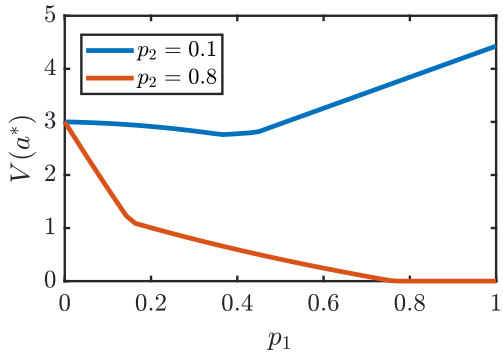


Figure 5.4: Expected reward under optimal policy for different go/no go probabilities

The pharmaceutical company prefers high p_1 and p_2 since these imply a successful outcome of the project, while a CRO, due to its limited capacity, prefers projects with a high p_1 and a low p_2. The CRO could increase its capacity C bu this is not easily done and it increases the risk of under-utilization.

5.5. Nonhomogeneous Projects

In the previous section, we considered a model with only one project type, which arrives with probability 1 in each period. In this section, we extend the model by allowing two project types and uncertain project arrivals. To keep the state space tractable, we consider projects that have at most two phases. For the MDP, we can do a similar analysis as in Section 5.4. The MDP now consists of $3^3 = 27$ states.

One of the policies is *always reject*. This policy's state space is shown in Figure 5.5. For this policy, transition probabilities are a function of arrival probabilities q_1 and q_2 only.

In each state, we have the possibility to accept or reject a project, if it is available. When there is no project, there is no action possible. In exactly $1/3$ of the states, there is no project, which means that in theory we have 2^{18} different policies. For similar reasons as in Section 5.4, the number of policies can be reduced to 37. The policies are similar to those in Table 5.1, but now also differentiated in terms of project types. For example, a CRO might focus on one project type and always reject the other type.

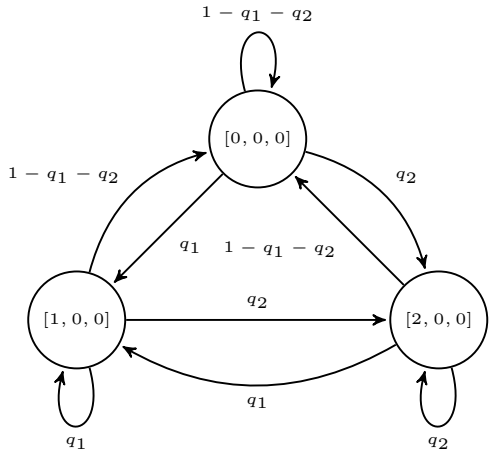


Figure 5.5: State space under 'always reject' policy for a model with two customer types.

	$c_{n,1}$	$c_{n,2}$	$r_{n,1}$	$r_{n,2}$	$p_{n,1}$	$\bar{r}_n$	$\bar{c}_n$
Type 1	3	3	3	3	1	6	6
Type 2	2	5	4	2.5	0.8	6	6

Table 5.2: Two project types with different characteristics but equal expected reward and capacity

To compare the project types, we introduce the expected reward and capacity of a project. These variables are defined as $\bar{r}_n = r_{n,1} + \sum_{m=2}^{M} p_{n,m-1} r_{n,m}$ and $\bar{c}_n = c_{n,1} + \sum_{m=2}^{M} p_{n,m-1} c_{n,m}$ and denote the total capacity and reward of a project, respectively. For example, when for two projects it holds that $q_1 \frac{\bar{r}_1}{\bar{c}_1} >> q_2 \frac{\bar{c}_1}{\bar{c}_1} >>$ the CRO might focus on type 1 projects as the average reward per unit of capacity is higher. By rejecting all type 2 projects, the CRO spares capacity for type 1 projects which are likely to arrive in the future periods.

When two projects have equal expected capacities and rewards, it is not obvious what a good policy for the CRO looks like. Consider the two project types in Table 5.2. Both projects have equal expected rewards and capacities. Type 2 projects have more uncertainty than type 1 projects. Type 2 requires more capacity in phase 2 then in phase 1, but in phase 1 it pays a higher reward per unit of capacity. Type 1 projects require equal capacity over all phases and phase 2 is always executed since $p_1 = 1$. The other parameters in the experiment are fixed at $C = 5$ and $\beta = 3$.

What is changed during the experiment is the mix of project types. More

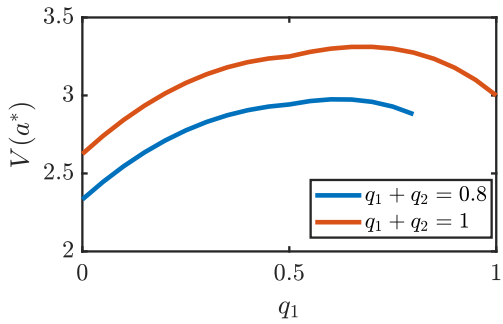


Figure 5.6: Optimal expected reward per period for different mixes of arriving projects

specifically, we change q_1 and q_2 but keep the sum $q_1 + q_2$ constant. In this way, the total value of incoming projects stays the same and we can see the effect of the customer types on the expected reward of the CRO. The results are shown in Figure 5.6. The lines indicate the expected reward under the optimal policy for two arrival processes. Blue line is for the process with a probability of a new project arrival of 0.8, while the red line denotes experiments where this probability is 1. Hence, when $q_1 = 0$, we have $q_2 = 0.8$ for the blue line and $q_2 = 1$ for the red line. As in the latter case, more projects arrive, the expected reward is higher for all combinations of projects. Although the total incoming value of projects remains unchanged for both lines, the actual reward obtained by the CRO changes based on the customer mix. First, we observe that the blue line at $q_1 = 0.8$ has a higher reward than at $q_1 = 0$. The same is valid for the red line; the reward at $q_1 = 1$ is higher than the reward at $q_1 = 0$. This implies that the CRO prefers Type 1 projects over only Type 2 projects. However, the maximum reward is attained when the arrival process is a mix of both project types. At the point where the maximum reward is attained two project types complement each other. If the CRO has a project of type 1 in phase 2 and a project of type 2 in phase 1 it exactly fills its capacity since $c_{1,2} + c_{2,1} = 3 + 2 = 5 = C$. The maximum reward is attained when $q_1 = 0.6$ (blue line) and $q_1 = 0.7$ (red line). The optimal policy for both points is to always accept projects when available, except for when there is a type 2 project in phase 1. In this way, it is avoided that there is another project in phase 1 when there is type 2 project in phase 2. These states are very costly because extra capacity is required.

5.6. Capacity Decision

In the previous section, we focused on project selection decisions at each period. We assumed that the available capacity in each period, C could not be changed. The capacity of a CRO can be changed and it is a long term decision. In this section, we consider C as a decision variable. We have two decisions to optimize; the long term capacity decision and the project acceptance decision. We write the average net reward per period as

$$V(a, C) = \Pi(a)^T \vec{R}(C). \tag{5.5}$$

To maximize $V(a, C)$, we can choose a policy $a \in A$ and a capacity C such that

$$0 < C \leq \max_{\vec{x}} \left\{ \sum_{m=1}^{M} \sum_{n=1}^{N} c_{n,m} \mathbf{1}_{\{x_m = n\}} \cdot \right\}. \tag{5.6}$$

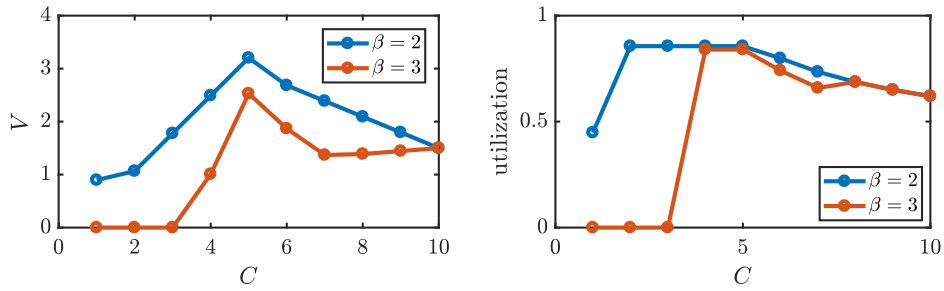
The upper bound for capacity follows from the state $\vec{x}$ that requires the most capacity. We then know that the reward in each state as defined in (5.1) is decreasing in C.

To give an idea of the effect of the capacity parameter, we consider an example similar to the one in Section 5.5; an example with two project types with two phases. The parameters for the project types are shown in Table 5.3. The cost per unit of fixed capacity is set to $\alpha = 1$. In this example, both project types have equal capacity requirements for phases 1 and 2. The projects differ in the go/no go probabilities of phase 1 and the rewards of phase 2. Type 1 has a very high probability of reaching phase 2, but the reward is the same as phase 1. A type 2 customer has a very low probability of reaching phase 2, but if it does, the reward is very high. For such an arrival process, it is on beforehand unclear whether the CRO should focus on a specific customer type or not. Furthermore, it is unclear how much capacity the CRO should install. The capacity question is answered in Figure 5.7.

Figure 5.7a shows the expected net reward under the optimal policy for a given value of C. According to (5.6), we have $0 < C \leq 10$. Two lines indicate the results for different cost for external capacity β. Similarly, Figure 5.7b shows the average utilization of the fixed capacity. When $C = 10$, the CRO can simply accept all projects because external capacity is never needed. Hence the lines for $\beta = 2$

	$c_{n,1}$	$c_{n,2}$	$r_{n,1}$	$r_{n,2}$	$p_{n,1}$	$\bar{r}_n$	$\bar{c}_n$	q_n
Type 1	5	5	7.5	7.5	0.9	14.25	9.5	0.4
Type 2	5	5	7.5	35	0.2	14.5	6	0.4

Table 5.3: Two project types with different characteristics but equal expected reward and capacity



(a) Value *(b)* Utilization

Figure 5.7: Expected net reward and utilization per period for different values of fixed capacity

and $\beta = 3$ congregate at this point. Due to the uncertainty in arrival ($q < 1$) and the go/no go probabilities, the utilization at this point is around 0.7. From Figure 5.7a we can conclude that the highest profit for the CRO is attained at $C = 5$, for both values of β. For $\beta = 3$, to achieve this result, a policy with the following characteristics is applied:

Always accept new projects, except for when there is a project of type 1 in phase I.

When there is a type 1 project in phase 1 with probability $p_{1,1} = 0.9$ it continues to phase 2 in the next period and consumes all the available capacity. A type 2 project in phase 1 most likely does not go to phase 2 ($p_{2,1} = 0.2$) and therefore it is safe to accept new projects. There is a risk that external capacity is needed, but not accepting new projects when there is a type 2 project in phase 1 results in the risk of an empty system if the pharmaceutical company decides not to continue the project.

For $\beta = 2$, the maximum reward at $C = 5$ is higher. This due to the lower cost for external capacity, but also due to a change in the policy. For these parameters, the optimal policy is

Always accept new projects, except for when there is a project of type 1 in phase I. Then, only accept projects of type 2 and reject projects of type 1.

This policy differentiates between both project types. The reason for this is that project type 2 has a higher expected reward per unit of capacity: $\frac{\bar{r}_2}{\bar{c}_2} > \frac{\bar{r}_1}{\bar{c}_1}$. As cost for external capacity are lower, the CRO can accept more projects, but only the most profitable ones.

Although in this example it is optimal to use a capacity of $C = 5$ and not accept all projects, there can be other motivations for a CRO to accept all projects. For example, to establish and maintain relations with pharmaceutical companies. Therefore, it is interesting to know at which capacity this becomes an optimal policy. For the case of homogeneous project types, this bound is stated in Lemma 5.4. However, for the nonhomogeneous projects obtaining such a bound is difficult if not impossible. From our numerical experiment it follows that for $\beta = 2$ it is optimal to always accept when $C \geq 6$. For $\beta = 3$ this is the case for $C \geq 8$. Under the always accept policy, the net reward is in this example linear in C on the domain $5 \leq C \leq 10$ and can be expressed as

$$V = 11.5 + (0.352\beta - 1)C - 3.52\beta.$$

which is decreasing in C.

The numerical results in this section indicate that finding the best combination of the capacity decision and the acceptance decision is not a straightforward task even when we consider a small number of projects with a few phases. The difficulty increases when the number of project types and the number of phases per project type increase.

5.7. Conclusions

In this chapter we studied the CRO's problem of managing research capacity for multiple drug development projects. The difficulty of drug development projects is that after each project phase a go/no go decision is taken. If the project is continued, research capacity is needed immediately, to reduce the time to market. To achieve this, one can already reserve research capacity, but if the project is discontinued, the

capacity is not used.

While most studies approach this problem from the perspective of a pharmaceutical company, we addressed the problem from the perspective of a contract research organization (CRO). Although in essence the problem is the same, the profitability of a CRO depends on different factors than the pharmaceutical company. For the pharmaceutical company, the result of the research project is essential for profitability, while for a CRO the contract with the pharmaceutical company and the utilization of its capacity are the most important factors that determine profitability. To manage its capacity, a key decision for a CRO is to the accept the right projects at the right time. We modeled this problem using an MDP. For the problem of homogeneous projects we determined a set of 6 policies that can be optimal. For each policy, we describe under which parameters it is optimal.

We also consider the problem of multiple project types. Projects differ in go/no go probabilities between phases, capacity requirements and rewards received per unit of capacity. We show that instead of focusing on one project type, the CRO should apply a policy where it accepts project types in such a way that projects complement each other and that capacity is highly utilized, but not over-utilized.

We also investigate how much capacity a CRO should install given the uncertain project arrivals. We show that it is not always optimal to choose a capacity such that all arriving projects can be accepted. The optimal policy becomes more complicated with an increasing number of customers and an increasing number of phases. As a further research we will focus on the development of heuristics. These heuristics will provide easy-to-understand and -implement guidelines for the capacity and project acceptance decisions.

6
Conclusion

In this thesis we develop quantitative models for stochastic project planning. We focus on two model types. In Chapters 2, 3 and 4 we develop models for planning individual projects that face uncertainty in task duration. We develop project plans that minimize total expected cost and guarantee on time completion with a certain probability. In Chapter 5 we propose a model for planning of multiple projects under a capacity constraint. Using a Markov Decision Process, we determine the optimal capacity levels and the corresponding optimal project selection policies.

In Section 6.1 we present the main results of this thesis and discuss their practical implications. In Section 6.2 we provide recommendations for future research.

6.1. Main Results

In Chapter 2 our research objective is to analyze the 'pay as planned' cost structure and compare it with the 'pay as realized' cost structure. We study an assembly system consisting of multiple parallel sub-assembly activities and one final activity. We minimize the pay as planned cost function and derive an optimal project plan. In this plan, each node in the network gets assigned an optimal planned leadtime. This result is obtained by taking into account the uncertain activity duration, the

location of the node in the network and the value it adds to the total project. We introduce a blame policy which identifies for each late completion exactly one node that caused it. We show that under the optimal solution the probability that a node is blamed for the lateness of the system equals a Newsvendor fractile. Accordingly, the more value a node adds to the final product, the more it can cause lateness of the final product.

The system that we study in Chapter 2 is rather simple with a single assembly activity. The real-world systems are much more complex but our results and insights continue to hold and hence, practitioners can rely on them to make decisions on which accounting scheme to use. We show that the pay as planned cost accounting scheme has several advantages compared to the pay as realized cost accounting scheme. First of all, the optimal solution under the former one leads to a higher probability that intermediate deadlines in the project are met compared to the latter one. Hence, this scheme should be preferred by companies where intermediate deadlines are important and costly to exceed. Companies that produce capital-intensive products are the ones that can benefit the most. These companies allocate expensive material to production stages and these are ready at the activity planned start times. If an activity cannot start at the planned start time, the costs related to keeping the material idle are incurred. Hence, for such a company relying on the pay as planned accounting scheme is beneficial since the likelihood of having expensive material and labor waiting for an activity to start is smaller. Companies in high-tech industry and construction industry are advised to use the pay as planned accounting scheme.

On the other hand, for companies that can easily allocate their material and labor to other projects and activities, having intermediate deadlines in addition to a deadline for delivering the final product is not usually the case. These companies are advised to use the pay as realized cost accounting scheme. We note that the optimal solution under the pay as realized cost scheme can advise negative leadtimes. This can be seen as one of the drawbacks of the pas as realized cost scheme since it might be difficult to interpret the negative planned leadtimes in practice. Further in the thesis, we show how to obtain an equivalent solution with non-negative leadtimes and we advise practitioners to rely on them.

Research objective 2 concerns the extension of structural results obtained for the assembly system with the pay as planned cost structure. To this end, in Chapter 3

we formulate the problem in terms of planned start and finish times instead of planned leadtimes. We use an Activity on Node (AoN) graph to describe the general network structure. We show that at optimality, for each leaf node, the probability of exceeding its deadline satisfies a Newsvendor equation. In the case of an exceeded deadline, it is possible to follow a path of back to back executed activities that caused the delay. We call this path a tardy path. For specific nodes in the network, the probability that the path is tardy satisfies a Newsvendor equation. We show that the Newsvendor equations also hold for when random activity leadtimes are dependent. This implies the pay as planned cost function can be applied to a wider range of problems than the pay as realized costing scheme. Although the optimality equations cannot be solved recursively, we show that the optimization problem is convex. This result allows for standardized optimization techniques to be used instead of tailored heuristics. We show that simulation based optimization provides a quick and accurate solution to the problem.

In Chapter 4 our objective is to provide an exact analysis for the pay as realized cost function. This cost function is especially useful in environments where intermediate deadlines are flexible, i.e. the costs of exceeding them are low compared to the costs incurred when actually starting an activity. The main result of our analysis is a system of optimality equations which is derived for the class of strictly converging networks. To interpret these equations we again use the concept of tardy paths. Instead of focusing on the complete system, we now introduce the concept of subsystems. Except for some special cases, it is possible to rewrite the equations into a set of Newsvendor equations which relate the relative added value of an activity to the probability of a critical tardy path. Using the subsystems, we show the dependencies among the optimality equations and show that they can be partly solved with a recursive procedure.

In Chapter 2, we compare the pay as planned and pay as realized accounting schemes for a rather simple network. In Chapter 3 and Chapter 4, we consider more realistic assemble-to-order systems with multiple end products and multiple assembly operations, respectively. In these chapters, we provide multiple important theoretical results which help us in characterizing the optimal solution. Our theoretical results generalize, extend and provide a good conclusion to the literature on planned lead times. Companies with complex network structures can rely on these results to evaluate and optimize their planned lead times.

The numerical results show that the optimal planned duration of an activity depends on 3 factors. First, it depends on the probability distribution of the stochastic leadtime. The higher the expected leadtime and the higher the variance, the more time is assigned. Secondly the location in the network is important. The closer an activity is to the final deadline, the more time is assigned. This time is also partially needed to mitigate delays caused by preceding activities. Finally the value a node adds to the final product is important. The more value a node adds, the more it causes a tardy path. One could say that cheap activities should always be ready and waiting for expensive activities, while cheap activities should never cause expensive activities to start later than planned.

In Chapter 5 we consider a capacity planning problem. This chapter concerns a Contract Research Organization (CRO) that selects project offers from pharmaceutical companies. Selecting the right offers is a decision under uncertainty as it is uncertain whether better project offers will arrive in the future. Our research objective is develop a mathematical model that supports this selection decision. Different from the previous chapters, we now assume that projects can be terminated immediately after completing an activity. This is typical practice when developing a new drug, for example, when test results demonstrate insufficient efficacy of the drug. We develop a Markov Decision Process Model and characterize a set of optimal policies. We determine the parameter combination for which each policy is optimal. Besides selecting the optimal policy for a given capacity level, we investigate the effect of the capacity constraint and the option of outsourcing additional capacity.

The key characteristics of a project that affect the CRO's decision are the success probability of each phase and the required capacity per phase. High differences in the capacity requirements for different phases and low success rates results in a high variance of the capacity utilization. Over and under-utilizing the capacity is costly and should be avoided. The cost of under-utilizing (the cost for keeping idle capacity) and the the cost for over-utilizing (the cost of sourcing capacity externally) are the most important parameters for determining the optimal capacity.

6.2. Directions for Future Research

For the pay as planned cost function in Chapter 3 we consider networks with fork-join structures and multiple end nodes. For the pay as realized cost function in Chapter 4 we assume strictly converging networks. It is interesting to investigate whether the results in Chapter 4 can be generalized to the networks in Chapter 3. However, to derive the optimality equations for the pay as realized cost function, the concept of subsystems plays a key role. Since we assume strictly converging networks and independent activity leadtimes, start and finish times in the subsystems could easily be determined. When the network contains fork-join structures the analysis of subsystems can be complicated. Any recursive properties of the optimality equations are lost if the assumption of independent leadtimes is relaxed. Hence, this extension would require an approach different than the subsystems analysis.

Another possible future direction is the generalization of results to different cost functions. A logical extension is a cost function that is a combination of both pay as planned and pay as realized costs. Then, each node can incur part of its holding cost from the planned start time and part of its cost from the actual start time. For such a cost accounting scheme, it can be easily shown that the general Newsvendor equation always holds (the proof of Lemma 4.6 remains unchanged). However, deriving the Newsvendor equations for individual nodes is challenging. Both cost accounting schemes assume holding costs and penalty costs that increase linearly over time. It could be useful to investigate whether the obtained results for linear costs can be generalized to nonlinear cost structures. For example, assuming quadratic penalty cost might reduce the occurrence of excessive delays, while maintaining the same service level.

The numerical experiments presented in this thesis provide an illustration to the structural results. To obtain the numerical results, we use standard optimization tools. These tools give a accurate result in a reasonable amount of time, but do not exploit specific characteristics of the problem. Future research could focus on developing heuristics that solve the problem for large networks faster and more accurate. For example, solving subsystems sequentially could be a candidate heuristic. Furthermore, for larger networks, one could group activities and compute a planned start time for the group.

In this thesis we focus on the theoretical analysis of both pay as realized and the pay as planned cost accounting schemes. For the pay as realized scheme, Atan et al. (2016) study a real life example of a lithography machine manufacturer. A future research direction might be to analyze the pay as planned scheme for the same industry, but also test it for other industries.

For the capacity planning model in Chapter 5 we derive the optimal project selection policies. For homogeneous customers and at most 3 phases per project, these policies are in a form that can be easily communicated to practitioners. When the number of customer types and the number of phases increase, the model becomes applicable to more practical cases. However, the number of different policies increases dramatically and it might be complicated to compute the optimal policy. Therefore, further research should use the optimal results of the homogeneous model to develop easy to understand heuristics with small optimality gaps.

Another useful extension is the relaxation of the assumption of a stationary arrival process. In this thesis we assume an uncertain arrival process, which is identical for every period. As the drug development market is continuously changing, the project arrival process can also change. In such an extended model, the arrival process could be a function of the CRO's selection decision. An example of such a dependency could be that accepting a small loss-making project could increase the probability of a large profitable project offer in the future. In this way, one can model the close collaboration between pharmaceutical companies and CROs.

Finally, a future research direction could be to combine features of the project planning model and the capacity constrained model. For example, assuming uncertain phase durations in the capacity planning model or adding a capacity constraint to the project planning model would improve the richness and applicability of the models. Although these are very useful extensions, they would require more complex models.

Bibliography

K. J. Arrow, T. Harris, and J. Marschak. Optimal inventory policy. *Econometrica: Journal of the Econometric Society*, pages 250–272, 1951.

Z. Atan, T. de Kok, N. P. Dellaert, R. van Boxel, and F. Janssen. Setting Planned Leadtimes in Customer-Order-Driven Assembly Systems. *Manufacturing & Service Operations Management*, 18(1):122–140, 2016.

Z. Atan, T. Ahmadi, C. Stegehuis, T. de Kok, and I. Adan. Assemble-to-Order Systems: A Review. *European Journal of Operational Research*, 261(3):866–879, 2017.

R. Atkinson. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International Journal of Project Management*, 17(6):337 – 342, 1999. ISSN 0263-7863.

S. Axsäter. Planning order releases for an assembly system with random operation times. *OR Spectrum*, 27(2-3):459–470, 2005. ISSN 0171-6468.

K. R. Baker and D. Trietsch. *Principles of Sequencing and Scheduling*. Wiley, 2019.

O. Ben-Ammar, A. Dolgui, and D. D. Wu. Planned lead times optimization for multi-level assembly systems under uncertainties. *Omega*, 78:39 – 56, 2018. ISSN 0305-0483.

P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European journal of operational research*, 112(1):3–41, 1999.

J. Buzacott and J. Shanthikumar. Safety stock versus safety time in MRP controlled production systems. *Management Science*, 40(5):1678–1689, 1994.

R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical programming*, 89(1): 149–185, 2000.

C. Charalambous and J. Gittins. Optimal selection policies for a sequence of candidate drugs. *Advances in Applied Probability*, 40(2):359–376, 2008.

S. Chauhan, A. Dolgui, and J.-M. Proth. A continuous model for supply planning of assembly systems with stochastic component procurement times. *International Journal of Production Economics*, 120(2):411–417, 2009.

S. Chopra, G. Reinhardt, and M. Dada. The effect of lead time uncertainty on safety stocks. *Decision Sciences*, 35(1):1–24, 2004.

B. Christian and S. Cremaschi. Heuristic solution approaches to the pharmaceutical R&D pipeline management problem. *Computers & Chemical Engineering*, 74:34 – 47, 2015. ISSN 0098-1354.

A. J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, 6(4):475–490, 1960.

M. Colvin and C. T. Maravelias. A stochastic programming approach for clinical trial planning in new drug development. *Computers & Chemical Engineering*, 32 (11):2626–2642, 2008.

M. Colvin and C. T. Maravelias. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. *European Journal of Operational Research*, 203(1):205–215, 2010.

M. Colvin and C. T. Maravelias. R&D pipeline management: Task interdependencies and risk management. *European Journal of Operational Research*, 215(3):616 – 628, 2011. ISSN 0377-2217.

J. F. Cox and J. H. Blackstone. *APICS dictionary*. American Production & Inventory Control Society, 2002.

T. Dai and S. R. Tayur. Healthcare operations management: A snapshot of emerging research. *Manufacturing & Service Operations Management, Forthcoming*, 2018.

T. de Kok and J. C. Fransoo. Planning supply chain operations: definition and comparison of planning concepts. *Handbooks in operations research and management science*, 11:597–675, 2003.

B. De Reyck and R. Leus. R&D project scheduling when activities may fail. *IIE transactions*, 40(4):367–384, 2008.

J. A. DiMasi, H. G. Grabowski, and R. W. Hansen. Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics*, 47:20 – 33, 2016. ISSN 0167-6296.

A. Dolgui, O. B. Ammar, F. Hnaien, M.-A. Louly, et al. A state of the art on supply planning and inventory control under lead time uncertainty. *Studies in Informatics and Control*, 22(3):255–268, 2013.

F. Y. Edgeworth. The mathematical theory of banking. *Journal of the Royal Statistical Society*, 51(1):113–127, 1888.

M. Elhafsi. Optimal leadtimes planning in serial production systems with earliness and tardiness costs. *IIE Transactions*, 34(3):233–243, 2002. ISSN 1573-9724.

M. Ferguson, V. Jayaraman, and G. C. Souza. Note: An application of the eoq model with nonlinear holding cost to inventory management of perishables. *European Journal of Operational Research*, 180(1):485 – 490, 2007. ISSN 0377-2217.

G. Gatica, L. Papageorgiou, and N. Shah. Capacity planning under uncertainty for the pharmaceutical industry. *Chemical Engineering Research and Design*, 81(6): 665–678, 2003.

K. Girotra, C. Terwiesch, and K. T. Ulrich. Valuing R&D projects in a portfolio: Evidence from the pharmaceutical industry. *Management Science*, 53(9):1452–1466, 2007.

E. M. Goldratt. Critical chain. Technical report, North River Press,, 1997.

E. M. Goldratt and J. Cox. *The goal: a process of ongoing improvement*. Routledge, 2016.

L. Gong, T. de Kok, and J. Ding. Optimal leadtimes planning in a serial production

system. *Management Science*, 40(5):629–632, 1996.

M. I. Henig and D. Simchi-Levi. Scheduling tasks with failure probabilities to minimize expected cost. *Naval Research Logistics (NRL)*, 37(1):99–109, 1990.

W. Herroelen and R. Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289 – 306, 2005. ISSN 0377-2217. Project Management and Scheduling.

W. Hopp and M. Spearman. Setting safety leadtimes for purchased components in assembly systems. *IIE Transactions*, 25(2):2–11, 1993.

J. Howells, D. Gagliardi, and K. Malik. The growth and management of R&D outsourcing: evidence from UK pharmaceuticals. *R&D Management*, 38(2):205–219, 2008.

A. Huchzermeier and C. H. Loch. Project management under risk: Using the real options approach to evaluate flexibility in R&D. *Management Science*, 47(1):85–101, 2001.

W. F. Jacob and Y. H. Kwak. In search of innovative techniques to evaluate pharmaceutical r&d projects. *Technovation*, 23(4):291–296, 2003.

S. Jansen, Z. Atan, I. Adan, and T. de Kok. Newsvendor equations for production networks. *Operations Research Letters*, 46(6):599 – 604, 2018. ISSN 0167-6377.

S. Jansen, Z. Atan, I. Adan, and T. de Kok. Setting optimal planned leadtimes in configure-to-order assembly systems. *European Journal of Operational Research*, 273 (2):585 – 595, 2019. ISSN 0377-2217.

D. L. Keefer and W. A. Verdini. Better estimation of PERT activity time parameters. *Management science*, 39(9):1086–1091, 1993.

J. E. Kelley Jr and M. R. Walker. Critical-path planning and scheduling. In *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, pages 160–173. ACM, 1959.

P. Kouvelis, J. Milner, and Z. Tian. Clinical trials for new drug development: Optimal investment and application. *Manufacturing & Service Operations Management*, 19(3):437–452, 2017.

D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar. Application of a

technique for research and development program evaluation. *Operations research*, 7(5):646–669, 1959.

A. Márkus, J. Váncza, T. Kis, and A. Kovács. Project scheduling approach to production planning. *CIRP Annals-Manufacturing Technology*, 52(1):359–362, 2003.

H. Matsuura and H. Tsubone. Setting planned lead times in capacity requirements planning. *Journal of the Operational Research Society*, 44(8):809–816, 1993.

H. Matsuura, H. Tsubone, and M. Kanezashi. Setting planned lead times for multi-operation jobs. *European Journal of Operational Research*, 88(2):287–303, 1996.

J. R. Meredith and S. J. Mantel Jr. *Project management: a managerial approach*. John Wiley & Sons, 2011.

P. Mirowski and R. Van Horn. The contract research organization and the commercialization of scientific research. *Social studies of science*, 35(4):503–548, 2005.

J. Mula, R. Poler, J. P. García-Sabater, and F. C. Lario. Models for production planning under uncertainty: A review. *International journal of production economics*, 103(1):271–285, 2006.

F. Pammolli, L. Magazzini, and M. Riccaboni. The productivity crisis in pharmaceutical R&D. *Nature reviews Drug discovery*, 10(6):428, 2011.

B. Ronen and D. Trietsch. A decision support system for purchasing management of large projects: Special focus article. *Operations Research*, 36(6):882–890, 1988.

K. Rosling. Optimal inventory policies for assembly systems under random demands. *Operations Research*, 37(4):565–579, 1989.

L. San-José, J. Sicilia, and J. Garca-Laguna. Analysis of an eoq inventory model with partial backordering and non-linear unit holding cost. *Omega*, 54:147 – 157, 2015. ISSN 0305-0483.

H. Shore. Setting safety lead-times for purchased components in assembly systems: a general solution procedure. *IIE Transactions*, 27(5):638–645, 1995.

D. Song, C. Earl, and C. Hicks. Stage due date planning for multistage assembly systems. *International Journal of Production Research*, 39(9):1943–1954, 2001.

J. S. J. Song, G. J. van Houtum, and J. A. Van Mieghem. Capacity and inventory

management: Review, trends, and projections. *Manufacturing & Service Operations Management, Forthcoming*, 2019.

D. Trietsch. Optimal feeding buffers for projects or batch supply chains by an exact generalization of the newsvendor result. *International Journal of Production Research*, 44(4):627–637, 2006.

D. Trietsch and K. R. Baker. {PERT} 21: Fitting pert/cpm for use in the 21st century. *International Journal of Project Management*, 30(4):490 – 502, 2012. ISSN 0263-7863.

J. Weeks. Optimizing planned lead times and delivery dates. 21^{th} *Annual Conference Proceedings, American Production and Inventory Control Society*, pages 177–188, 1981.

D. C. Whybark and J. G. Williams. Material requirements planning under uncertainty. *Decision sciences*, 7(4):595–606, 1976.

C. A. Yano. Setting Planned Leadtimes in Serial Production Systems with Tardiness Costs. *Management Science*, 33(1):95–106, 1987a. ISSN 0025-1909.

C. A. Yano. Stochastic leadtimes in two-level assembly systems. *IIE Transactions*, 19 (4):371–378, 1987b.

J.-Y. Yu and J. Gittins. Models and software for improving the profitability of pharmaceutical research. *European Journal of Operational Research*, 189(2):459–475, 2008.

Zion Market Research. CRO Market for Early-Stage Development Services and Last-Stage Development Services: Global Industry Perspective, Comprehensive Analysis and Forecast 2014 - 2020, October 2015.

Summary

Quantitative Models for Stochastic Project Planning

In this thesis, we focus on projects. We see projects as a collection of tasks that need to be executed to achieve a specific goal, for example manufacturing a product. In order to achieve this goal, resources such as capital, workforce and equipment are required. These resources are typically constrained because they are costly and other projects also make use of them. In order to achieve the project goal the project needs to be managed.

The execution of projects is subject to uncertainties. Costs can be higher than expected, durations can be longer than expected and so on. We can not mitigate these uncertainties, projects are uncertain in nature. However, in order to help decision making we can develop mathematical models that take into account these uncertainties. In this thesis we focus on the development of stochastic models that take into account several types of uncertainties in project planning.

The thesis is split up in two parts. In the first part we focus on project planning with uncertain durations of individual tasks. Although there are many application areas, in these chapters we mainly consider manufacturing systems where low volume, capital intensive and customer specific products are produced. Examples are airplanes or lithography machines. Because of these characteristics we consider

each product as a separate project instead of using a traditional production control approach. The goal of these project is to deliver the machine on time to the customer.

We model the projects via activity on nodes (AoN) networks. For each activity we have information about the uncertainty in durations either from estimations or from data from previous projects. The goal of the model is to assign each activity a planned start time in such a way that total cost are minimized and an on time completion is achieved. Since we take into account the complete network, safety time is assigned, there where it is most effective.

Since we face stochastic leadtimes, it is possible that a product is completed late. In order to find the activity that caused this lateness, we introduce the concept of tardy paths. This set of rules identifies exactly one activity in that is held responsible for this late completion.

In the minimization problem, we balance holding cost which can be interpreted as cost for being early for a deadline and penalty cost, the cost for being late at a deadline. For different cost structures, we derive optimality equations. These optimality equations have a Newsvendor shape. The equations relate the probability that an activity is causing a tardy path to a Newsvendor fractile, which denotes the relative value added by an activity to the final product. We show that the more value an activity adds, the higher the probability that it causes a tardy path. Besides the optimality equations, we derive structural results of the optimal solutions and compare the optimal solutions of different cost functions.

In the second part of the thesis, we consider the uncertainty of go/ no go decisions in projects. Instead of assuming that the complete project is executed, we assume that projects can be terminated early. A typical area where this is a relevant problem is new drug development in the pharmaceutical industry, where development projects can be terminated when test results show limited efficacy or serious side effects.

In recent years pharma companies more and more outsourced these development projects to contract research organizations (CROs). These companies have highly specialized resources, such as lab facilities, nurses, doctors and test patients. These resources can be assigned to projects from possibly different pharma companies. Projects can differ in capacity requirements, go/no go decisions and revenue. For

a CRO it is key to utilize the available research capacity by accepting the right projects at the right moment. The difficulty here is that if a project is terminated by the pharma company, the CROs capacity utilization drops. Furthermore, there is uncertainty in new projects that arrive. Once a project arrives the CRO needs to decide whether to accept or reject the project. When this decision is made, it is uncertain if projects arrive in the future and what characteristics they will have.

We model this acceptance decision as Markov Decision Process (MDP). The state describes the projects that are currently executed by the CRO. Each go/no go decision and each new project arrival are modeled by a state transition. A policy describes for each possible state whether to accept or reject new projects. For a given system we can characterize optimal policies for different customer arrival processes.

Furthermore we assist the long term capacity decision of a CRO. For given arrival processes we determine how much research capacity a CRO should build and what projects it should accept. For systems with many different projects, optimal policies can be quite complicated. Therefore we develop heuristics that are easy to understand and implement but also result in close to optimal performance.

Acknowledgments

After graduating from my masters, I spend a year on deciding on whether or not to do a PhD. A PhD is difficult, not only in academically, but also personally. Now, looking back at 4 years PhD it turns out to be a great decision. Of course you need some capabilities to start a PhD, but in my opinion, the key factor in completing a PhD is the people around you. I'm very thankful to the people around me in the last four years.

A first word of gratitude goes to my first promotor Ton de Kok. Your broad knowledge of both academia and industry helped me a lot in finding my way in this field of research. Your enthusiasm about my work is admirable and pushed me a lot to solve complicated mathematical problems. In the first weeks of my PhD, in the summer of 2015 we had a lot of 'quality time', as you call it. In this period we had hours and hours of brainstorming sessions. I think in this period, the foundation was built for a great collaboration. Later in my PhD your time was more precious. However, your door was always open, especially early in the morning.

Then my second promotor Ivo Adan. We already know each-other since the start of my master Mechanical Engineering in the summer of 2011. During all those years, I really appreciated your personal approach. Quite often, only half of the meeting time we talked about research, the rest was about everything else. In my

PhD, you really helped me in the theoretical parts of my work. I learned developing mathematical proofs and above all, I really enjoy it. I think your standard comment on my papers was: 'I think I know what you want to say here but that is not what you wrote down.' Although your comments are sometimes frustrating, they were always correct, valuable and constructive which always resulted in a better paper.

The person who helped me the most in these 4 years is of course my co-promotor Zümbül Atan. Having you as a daily supervisor was a real privilege. You know every little detail of this thesis almost as good as I know it. But for you, my thesis was only one of many things that were on your plate. Still you find time to read every single file I sent you and have a meeting with me every week. You were the one that kept me on the right track, as I tend to deviate a lot from it. That's the fun of research, each answer leads to new questions, but sometimes it is the time to write things up and meet deadlines. Thank you as well for all the fun moments, all the morning coffees and also the dinners we had. Your hospitality, generosity and care are incredible.

I'm grateful to Feryal Erhun for our collaboration in the past year. Thank you for hosting me in Cambridge. My stay in Cambridge really broadened my view on research and academia. You constantly pushed me to show the impact of my models in practice rather than the result of the model itself. You did this by always asking the right question; questions that were unexpected and really made think.

Furthermore I would like to thank Erik Demeulemeester, Dan Trietsch and Nico Dellaert for being involved in my committee. I thank you for spending time on reading my thesis, the valuable comments I received and for traveling to Eindhoven for my defense ceremony.

Besides the people directly involved in my thesis there are many more colleagues in the OPAC group that contributed to an exciting working environment. Thank you Claudine, Christel, Jolanda and José for all the support. Thank you to all my fellow PhDs for all the discussions about research and the social activities we had. In particular, I would like to thank Bram, Laura, Volkan, Joni, Simon, Kay, Joost, Taher, Dalia, Afonso, Loe, Mirjam, Ipek and Denise.

Sometimes it is good to step away from PhD and have some free time. Many thanks to all the friends that made the last years superb. In particular, I would like to thank Evert, you were the one helping me writing a motivation letter for this PhD during

our travels. Jos, thank you for being a great friend for such a long time already. Stijn, thank you for funding this PhD for an infinitesimal small part. Ruud, thank you for being my running mate in the last years. Kees, thanks for all the beers we drank together and the great chats we had while doing so. Thanks to the 'Acti Vento' board for all the great city trips, bbqs and parties we had. Thanks as well to the Telegram group 'Eten' for all the dinners (obviously).

This thesis would also not have been there without the help, love and support of my family. Mom, Dad, thanks for being great parents. You never pushed me in a specific direction but always let me make my own decisions. You always supported me in every decision I made. Twan and Ivo, thanks for always being there for me. Having you two as brothers makes the choice for paranymphs very easy. Thanks for the awesome moments we had together, and the awesome moments that will come. Wherever on this globe it might be. Contrary to the proverb, to me a good neighbor is as good as a far friend.

The final word is for you, Yeşim Koca. Thanks for being a great colleague and especially thank you for loving me in the last few months. I'm extremely delighted to address you as girlfriend instead of colleague.

About the author

Sjors Jansen was born on January 3rd, 1989 in Angeren, The Netherlands. In 2007 he started a Bachelor in Mechanical Engineering at Eindhoven University of Technology, where he obtained his degree in the summer of 2011. In this period, he also served one year full time as treasurer of the board of study association Simon Stevin.

He continued for a Master in Mechanical Engineering, also at Eindhoven University of Technology. He specialized in Manufacturing Networks and Systems Engineering. In the fall of 2012 he did a semester abroad at Auburn University, Alabama, USA where he worked with Prof. Kevin Gue on a sequencing algorithm for parcel sorting in warehouses. In 2013, he started his graduation project and graduated at Vanderlande. There he developed and implemented control software for baggage handling systems using model based design. In the summer of 2014 he obtained his degree "with great appreciation".

Before starting his PhD he spent a year backpacking in several countries in Asia and Africa. In the summer of 2015 he started his PhD within the Operations Planning Accounting and Control group at the school of Industrial Engineering. Under supervision of Ton de Kok, Zümbül Atan and Ivo Adan he worked on optimal planning of stochastic activities in production networks. In the spring of 2018 he

spent 4 months at Judge Business School, Cambridge University, United Kingdom. Under supervision of Feryal Erhun he worked on a capacity planning model to be used in in the pharmaceutical industry.