Order acceptance under uncertainty in batch process industries

Order acceptance under uncertainty in batch process industries

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr. R.A. van Santen, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op maandag 4 oktober 2004 om 16.00 uur

door

Virginia Cristina Ivănescu

geboren te Boekarest (Roemenië)

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J.W.M. Bertrand

en

prof.dr. J.P.C. Kleijnen

Order acceptance under uncertainty in batch process industries

Virginia Cristina Ivănescu

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Ivănescu, Virginia Cristina

Order acceptance under uncertainty in batch process industries / by Virginia Cristina Ivănescu. – Eindhoven : Technische Universiteit Eindhoven, 2004. – Proefschrift.

ISBN 90-386-2097-7

NUR 804

Keywords: Production control / Order acceptance / Batch process industries / Regression analysis

Printed by Universiteitsdrukkerij Technische Universiteit Eindhoven

Cover design by Paul Verspaget

Acknowledgements

A publication such as this would not be possible without the assistance, guidance, and kind words provided by many people. It is my pleasure to express now my gratitude to all of them.

First of all, I would like to thank prof. dr. ir. J.W.M. Bertrand and prof. dr. ir. J.C. Fransoo for initiating this project. My daily advisor, prof. dr. ir. J.C. Fransoo, has continuously guided and helped me to grow as a researcher. His optimism and sense of humor were invaluable in helping me to put things in their proper perspective. I also thank my first advisor, prof. dr. ir. J.W.M. Bertrand, for his valuable ideas, his constant interest in my research progress and his sharp and quick feedback in all matters. I am very indebted to my second advisor, prof. dr. J.P.C. Kleijnen, for his prompt feedback and corrections regarding the elaboration of this thesis. I also thank prof. dr. J. Wijngaard, who took effort in reading my work and providing me with valuable comments on earlier versions of this thesis. The other members of my promotion committee prof. dr. A.G. de Kok, dr. A. Di Bucchianico, prof. dr. A. van Harten, and dr. ir. Z. Verwater - Lukszo are kindly acknowledged for their acceptance and interest in this defence.

I want to thank all my colleagues in the OPAC department for providing a motivating and enthusiastic atmosphere in our group. I am deeply indebted to my office mate Bogdana Drăguţ for being a great source of practical information, as well as being happy to be the first to hear my outrage or glee at the day's current events. Special thanks go to Gudrun Kiesmuller for listening and offering her support in many occasions, to Judith Spitter for taking care that I will not skip the coffee, lunch, or the tea break, to Mustafa Dogru for cheering me up during hard times, and to Nessim Erkip for taking the time to listen to me. I enjoyed the company and support of many others whom I fail to mention here, and I'm thankful for their kindness.

The four years I've spent in Eindhoven have been made infinitely more pleasant on account of many friends I've made while living here. I am very grateful to my compatriots (in alphabetical order!) Călin Ciordaş, Bogdana Drăguţ, Mihaela Iftimi-Ilie, family Maxim, Laura Măruşter, Anca Molnoş, Simona Vlad, and Marton Zelina, who helped me to feel almost at home. I also want to mention the members of the HocHabet fencing club who substantially contributed to the improvement of my Dutch and took care that I will not work too hard and skip the nice activities they organize.

Finally, I wish to express my sincere thanks to my family who have always been there for me, offering their unconditional love. Last but not least, I thank Terry for his continuing emotional support during hard times and practical support during busy times, especially during the last phase of my project.

Cristina Ivănescu,
October 2004

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research topic and motivation

In this thesis we study the order acceptance function in a multi-resource production system with overlapping processing steps, no-wait restrictions among processing steps and stochastic processing times. The motivation for studying such a system originated from process industries.

In the process industries one finds a large variety of businesses "that add value to materials by mixing, separating, forming, or chemical reactions" (Wallace, 1984). The American Production and Inventory Control Society (APICS)'s definition of the process industry distinguishes between two types of process industries, stating "...Processes may be either continuous or batch..." (Wallace, 1984). Flow process industries are characterized by one (or very few) processing steps, the same processing routings for all products, a divergent material flow, and a low added value. Examples of flow process industries include the oil and steel industry, glass manufacturing, and paper production. Batch process industries, in contrary, are characterized by a large number of processing steps, different product routings, a convergent material flow, and a high added value. Examples of batch process industries include the food industry, specialty chemicals, and the pharmaceutical industry. In this thesis, we consider only *batch process industries*.

Process industries generally use a capacity-oriented planning and scheduling framework, which - in contrast to fabrication and assembly industries - schedules capacity before raw materials. One reason is the small variety of raw materials that process industries use in large quantities. The raw materials have low values by themselves, but the cost of adding value to them is high. The capacity-oriented planning and scheduling framework consists of resource requirements planning, production planning, and scheduling (Taylor *et al.*, 1981). Stated briefly, the resource requirements planning develops the long term strategic plans to acquire the resources necessary for future operations. Production planning is concerned with adapting the capacity requirements over time to the available capacity, and allocating sufficient resources to

deal with the incoming demand in the medium term. Finally, scheduling determines the short term detailed specifications of what is required, when it should be produced, and where it is produced.

Order acceptance is the decision function which is responsible for the coordination of capacity requirements due to customer orders and the available capacity over time. Basically, it deals with accepting or rejecting customer orders. It may also deal with related decisions, such as due dates setting and price determination.

The order acceptance process differs significantly for various manufacturing environments. In a production-to-stock environment, customer orders are generally accepted with a short delivery time if there is sufficient inventory of the demanded product. If not, a later delivery date is agreed with the customer, based upon the expected stock fill rate. In production-to-order environments, orders should be accepted in such a way that capacity utilization and delivery reliability are maximized.

Markets in which batch process industries operate are characterized by an increasing demand uncertainty, both in product mix and in product volume. The markets demand an increasing diversity of products (product proliferation). Due to this increased product variety and an increased demand for customer specific products, batch process industries show a tendency towards increasing production-to-order, as opposed to forecast driven production (Ten Kate, 1994).

Given this increasing competitive pressure in the marketplace, the key to success for a production-to-order company lies in consistently meeting delivery promises to win and keep satisfied customers. When demand exceeds available capacity, it might appear at first glance that the company should expand capacity or schedule overtime to obtain dependable delivery performance. Although this could be a viable long-term option, it is not a feasible option in the short-term. This is due to the high capital investment in resources and the need for a highly skilled labor force. Hence, with on-time delivery as an important aspect of the company's competitive strategy, it becomes necessary to balance the capacity requirements (or demand) and the available capacity over time. This may be done by carefully assessing which customer orders to accept and how to allocate the available capacity to the accepted orders over time.

Customer orders arrive irregularly over time and the acceptance decision can be made upon arrival of each order or upon a number of orders that have arrived in a specific decision period. There are several aspects to be considered for the acceptance or rejection of customer orders; most important are the following two:

- the availability of resources and materials, and
- the cost and revenues that result from accepting the order.

In this thesis, we develop models that evaluate on-line for each arriving order the implications of accepting the order for production. The acceptance decision is based on the availability of capacity to complete the order and the already accepted orders before their requested due dates.

Order acceptance has a large influence on the performance of a company. On one

hand, rejecting too many orders leads to low capacity utilization. In addition, rejecting a customer order may have further repercussions for future customer relations. On the other hand, accepting too many orders leads to an over-loaded production environment, where lead times increase and orders are increasingly delivered late. And promising to deliver an order to a customer by a certain date and failing to do so may again result in lost of goodwill. A good method to support order acceptance decisions is, therefore, essential to solve these problems. In developing an adequate order acceptance method for batch process industries, their specific characteristics should be considered explicitly. In the following section, we discuss the typical characteristics of batch process industries (see Section 1.2.1) and their consequences for planning and scheduling (see Section 1.2.2). We also present a brief overview of the available literature on planning and scheduling in batch process industries (see Section 1.2.3). In Section 1.3, we evaluate the available literature and we further elaborate on the motivation for this research. In the last two sections of this chapter (sections 1.4 and 1.5), we address our research questions, we discuss the methods we use to answer these questions, and we introduce the subject of each chapter of this thesis.

1.2 Planning and scheduling in batch process industries

According to Kallrath (2003), planning and scheduling is part of company-wide logistics and supply chain management. He further states that planning in process industries is used to "create production, distribution, sales and inventory plans based on customer and market information while observing all relevant constraints". Planning is typically associated with a longer term horizon and involves less detail, whereas scheduling defines the precise timing and sequencing of individual operations as well as the assignment of the required resources over a short period of time. Planning and scheduling are closely related as the decisions made at the planning level have a strong influence on scheduling.

1.2.1 Distinctive features of batch process industries

Batch process industry can be considered as a specific domain within planning and scheduling. This is due to a number of specific characteristics, which have been well documented (see e.g., Hayes & Wheelwright, 1979; Taylor *et al.*, 1981; Fransoo & Rutten, 1994) and will be discussed in this section.

Production characteristics The process equipment for batch process industries consists of tanks, mixers or reactor vessels which are linked by a network of pipelines and transfer units. We can distinguish two basic types of process equipment: multi-product which is used to produce different variants of a basic product type following the same routing (cf. a flow shop), and multipurpose which allows diverse processing tasks to be carried out by a particular equipment unit (cf. a job shop) (Reklaitis,

1996). In this thesis, we focus on batch process industries using *multipurpose equipment*.

Production in multipurpose batch process industries is described by the following dominant characteristics (Raaymakers, 1999):

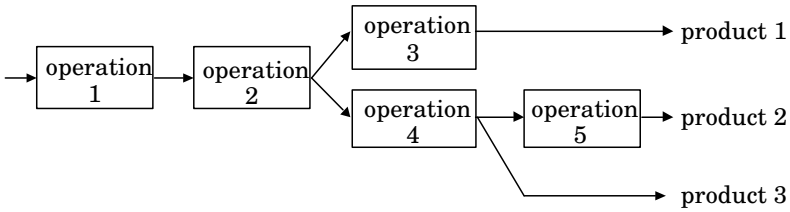
- multipurpose resources,
- high product variety,
- long, divergent routings,
- overlapping processing steps, and
- waiting time restrictions.

Production is carried out by multipurpose resources that may be used to perform a variety of different processing steps. As a result, for each processing step a number of similar resources are available, which may be used as alternatives. The availability of multipurpose resources is a necessary condition to be flexible to produce a large variety of products.

Production occurs in discrete batches. The batch size is limited by the equipment unit that can handle the smallest batch. Batch processing times are not highly correlated with the size of the batch in a particular unit. The chemical reaction time is often independent of the amount of chemical in the unit. Thus, it will often take the same amount of time to process a half filled or a completely filled unit.

A number of subsequent operations has to be carried out to produce a product. Each operation results in a stable intermediate product, which can be stored. The number of operations required may vary for each final product, but total production lead times are generally long, up to one year. Routings may be divergent, which means that intermediate products resulting from the same operation may be used to produce different finished products. An example of a routing is given in Figure 1.1.

Figure 1.1: Routing example



Each operation consists of one or more processing steps. By definition, intermediate products are not stable during an operation and, therefore, must be further processed without delay. This results in no-wait restrictions between processing steps of an operation. A processing step is defined such that for each processing step exactly one resource is required. Processing steps of an operation may be overlapping in time.

The product being processed is generally a fluid or powder that needs to be kept by a resource at any time during production. Therefore, processing steps are directly succeeding or overlapping in time. An example of an operation is given in Figure 2.3.

Demand characteristics Demand in multipurpose batch process industries is best described by the following dominant characteristics:

- low demand volume per product,
- small number of customers per product, and
- high variability and dynamics of demand.

The demand volume for each individual product is relatively small. The turnover of a company may still be considerable due to the large range of high added-value products, for example ingredients for pharmaceutics or flavors & fragrances. Each product is produced for a small number of customers. Demand for individual products is often lumpy because of the low order frequency in combination with minimum order quantities used by customers.

Demand is usually highly variable and dynamic. The product assortment changes relatively quickly, and the rate at which new products are introduced increases steadily (Wera, 2002). It is therefore difficult to provide accurate and reliable demand forecasts. In the pharmaceutical industry, acquisition and expiration of patents and the introduction to new markets influence the demand volumes for products over time. In other types of industry, seasonal effects may have an important additional influence on demand volumes.

1.2.2 Consequences for planning and scheduling

The large product variety in combination with variable demand results in capacity requirements that are variable over time. This variation applies to both the total capacity requirements, and to the capacity requirements for different resources. Often bottlenecks are not stable over time. Production planning is, therefore, mainly concerned with the coordination of capacity requirements and the available capacity over time and with allocating sufficient resources to deal with the incoming demand. For multipurpose process industries, the level of available capacity is relatively inflexible due to high capital investments in resources and highly skilled labor force. Consequently, production planning aims at smoothing the capacity requirements. Order acceptance decisions are made to smooth capacity requirements over time.

The no-wait restrictions between processing steps have a considerable impact on production planning and scheduling. Production orders - called jobs - are related to a single operation in the routing of a product. Consequently, several subsequent jobs are required to produce a specific product. Jobs consist of several processing steps, each requiring exactly one resource. The number, type, and sequence in which these resources are required may differ among jobs. In order to meet the no-wait restrictions

between processing steps, the scheduler has to make sure that all required resources are available at the right time before the processing of a job can start.

1.2.3 Literature review

To position our research, we now discuss related work on production planning and scheduling in multipurpose batch process industries. Multipurpose batch process industries are characterized by (i) their flexibility and ability to produce a large number of different products in different qualities so as to satisfy the customer's demand, and (ii) the possibility of considering alternate production paths for the same product in order to reduce costs. This high degree of flexibility is one of the major sources of complexity in production planning and scheduling problems. In the last decades, a considerable amount of research has been carried out on production planning and scheduling issues arising in process industries. Comprehensive reviews on this subject are provided by Reklaitis (1996) and Kallrath (2003).

Most of the publications on multipurpose batch process industries have focused on developing better models for scheduling alone, without involving planning. It is beyond the scope of this thesis to give even a selected survey on the existing literature on scheduling in multipurpose batch process industries. We only outline here the most common solution methods and asses the approaches with respect to the modelling effort. Among the numerous models and solution approaches proposed in the literature (see Kallrath (2003) for an extensive overview), we found that mathematical optimization techniques and stochastic search techniques are the most common ones.

Most of the approaches that make use of mathematical optimization techniques may be classified on the basis of the time representation followed. Early research (see e.g. Kondili *et al.*, 1993; Shah *et al.*, 1993) discretized the time horizon into a number of intervals of equal durations and assumed that events only happen at the boundaries of these time intervals. The problem is formulated as a mixed-integer linear programming (MILP) model using the state-task network (STN) representation. The main limitations of the time discretization models are that (i) they correspond to an approximation of the time horizon and (ii) they result in an unnecessary increase of the number of binary variables in particular and of the overall size of the mathematical model (Ierapetritou & Floudas, 1998). Therefore, in more recent work, both the state-task network (STN) and the resource-task network (RTN) representations are considered in continuous time (see e.g. Pinto & Grossmann, 1995; Dimitriadis *et al.*, 1997; Bok & Park, 1998; Ierapetritou & Floudas, 1998; Schilling & Pantelides, 1999; Kim *et al.*, 2000). However, despite the improved formulations, and the recent improvements in computer hardware and optimization software, the short-term scheduling of STN or RTN multipurpose batch plants in continuous time remains a difficult problem to solve (Maravelias & Grossmann, 2003).

Given that most scheduling problems belong to the class of NP-complete problems - even when simplifications in comparison to real life problems are introduced - it is often argued that only the use of problem specific heuristics can lead to efficient solution procedures. Among the stochastic search techniques, simulated annealing (SA) and

genetic algorithms (GA) are the most used techniques to solve the scheduling problem of multipurpose batch process industries under various storage policies (see e.g. Ku & Karimi, 1991; van Bael, 1999; Raaymakers & Hoogeveen, 2000; Bernal-Haro *et al.*, 2002). These techniques have the disadvantage that no measure of quality can be given to judge their solutions (i.e. they usually lack the proof of optimality). The main advantage is that these algorithms incorporate problem specific knowledge which often leads to good solutions, which are obtained in an acceptable amount of time.

A significant amount of work has also been dedicated to developing production planning models for multipurpose batch process industries. The most common approach is the campaign planning. A campaign mode of operation dedicates all plant resources to a single product or a small subset of products with similar processing requirements over a period of time (the "campaign") and thus reduces the cost of changeovers. Two different classes of approach can be further distinguished within the campaign planning literature: hierarchical (also called "sequential") (see e.g. Mauderli & Rippin, 1979; Wellons & Reklaitis, 1991; Papageorgiu & Pantelides, 1993, 1996; Grunow *et al.*, 2002) and simultaneous (see e.g. Shah & Pantelides, 1991; Voudouris & Grossmann, 1993). A general mathematical formulation of campaign planning problem and a comprehensive literature review of both approaches are given by Papageorgiu & Pantelides (1996). However, a noteworthy observation of Papageorgiu & Pantelides is that campaign planning is particularly appropriate when stable demand patterns over a long planning horizon are observed.

A different approach to the production planning problem in multipurpose batch process industries is proposed by Raaymakers (1999). Within a hierarchical production planning framework, she addresses the capacity estimation problem. The complex capacity structure in batch process industries in combination with variable and dynamic demand result in variable bottlenecks over time. This makes it difficult to determine which part of the available capacity can be effectively used for production. Raaymakers uses regression analysis to estimate the output that can be realized by the production system. She demonstrated that in static and deterministic situations, the makespan of a given set of jobs (i.e. the completion time of the last job) can be estimated reliably with a regression model including only a small number of parameters (explanatory variables). This regression model is further used to support the order acceptance decisions in a setting with random order arrivals and deterministic processing times.

In some publications, the production planning and scheduling problems are integrated and defined as sub-problems in the design of batch chemical plants. Subrahmanyan *et al.* (1995) and Subrahmanyan *et al.* (1996) consider a Design Super Problem (DSP) which handles aggregate decisions and a number of detailed scheduling problems. These problems are all formulated as MILPs. A solution for DSP does not necessarily result in feasible solutions for the scheduling problems. Therefore, the DSP and the scheduling problems are solved iteratively, until feasible solutions are found. This is done prior to deciding on the design of the plant. The decomposition of the overall problem into sub-problems is done for computational reasons. Demand uncertainty is considered in the DSP as a set of scenarios. Each scenario provides a given demand over a relatively long horizon.

Zhu & Majozi (2001) proposes a novel procedure for the integration of production planning and scheduling in multiplant operations. This procedure entails decomposition of the overall planning and scheduling problem into two levels. At the first level the planning model is formulated and solved for the optimal allocation of raw materials to individual processes. At the second level, the raw material targets obtained from the planning model are incorporated in the scheduling models for individual processes. These models are then solved independently.

1.3 Contributions of this thesis

The literature review in the previous section reveals that there are many research contributions to production planning and scheduling problems in multipurpose batch process industries. However, we saw that most of the contributions focus on the scheduling problem. With respect to the production planning problem in batch process industries, two shortcomings may be identified. First, the methods proposed in the literature assume that detailed and accurate information on demand and production is available. However, multipurpose batch plants often operate in a dynamic stochastic environment and the assumption that all orders are known in advance is not very realistic. Furthermore, production disturbances may occur, which affect the future status of the production system. We may distinguish two types of production disturbances. The first type is caused by uncertainty of the resource availability. In industrial practice, breakdowns of resources may occur. However, in process industries such breakdowns do not occur frequently. Therefore it seems reasonable to assume uninterrupted resource availability. The second type of production uncertainty is caused by uncertainty in the production process. Processing times may vary as well as the product quality. Quality variations may result in rework, for which additional capacity is required. In batch process industries, fluctuations in the quality of raw materials, catalyst, or fluctuation of cleanness of equipment result in high degree of processing time variation (Ishii & Muraki, 1996). Therefore, processing time uncertainty should be taken into consideration.

Second, except for the contribution of Raaymakers (1999), no attention has been payed to the order acceptance function in the planning and scheduling literature in batch process industries. Due to the dynamics and variability of demand, the capacity requirements may vary considerably over time. Given that the available capacity is generally fixed in the medium term in this type of industry, smoothing the capacity requirements to the available capacity is becoming essential. This may be done by carefully selecting which customer orders to accept, based on the availability of capacity to complete the orders before their requested due date.

Different policies may be used to evaluate whether sufficient capacity is available to produce an order before the due date requested by the customer. Generally, the order acceptance policies used in industry are workload based, in the case where the capacity complexity is low or sufficient slack exists in the system, or schedule based in the case of less slack in the system or increased complexity (Raaymakers *et al.*, 2000b). Due to the high scheduling complexity and many interrelations between processing steps

in batch process industries, schedule based evaluations are very time consuming. An alternative is to use aggregate information. However, the main problem resulting from neglecting detailed scheduling information is that many of the accepted orders cannot be completed in the planned period. This results in low delivery reliability and many replanning activities. The idea is to capture the complexity of the scheduling task through an aggregate model which can accurately estimate the production output that can be realized with the available capacity. Previous research (Raaymakers, 1999) has yielded new insights with respect to the possibility of using statistical methods for this purpose. This thesis is an extension of her work. However, contrary to the environment studied by Raaymakers, we consider settings with stochastic processing times.

In some cases, it may be fairly straightforward to estimate whether sufficient capacity is available to produce an order before the due date requested by the customer. This is for instance the case in flow process industries, where a single resource is dominant in determining the available capacity (Fransoo, 1993). If the level of interaction between jobs and resources and between jobs themselves is high - as in the case of batch process industries - the estimation process becomes very complex. In traditional job shops, capacity loading decisions - which are closely related to order acceptance decisions because the availability of sufficient capacity is the main criterion for accepting an order - are generally addressed using queueing theory (Buzacott & Shantikumar, 1993). A shortcoming of queueing analysis is that it is hostage to its assumptions: practical production problems seldom satisfy the assumptions needed to obtain analytical results. While these results can often be used for simple production systems, the need for alternative approaches increases as production complexity increases. Such an alternative may be based on regression analysis.

Statistical methods have a long tradition in the field of Operations Research (OR). Empirical studies by Ford *et al.* (1987) and Lane *et al.* (1993) indicate that statistical methods - especially regression analysis - is one of the top-three OR techniques that OR educators teach and practitioners find most useful. Besides its major role in sensitivity analysis, regression analysis is also used in forecasting and prediction problems in many areas, including manufacturing planning and control, projecting workforce requirements, and development of project costs and cash flows. For example, simple and multiple linear regression analyse are common approaches in the due-date assignment literature. Researchers used a variety of job-related and shop-related factors as independent variables to predict the flowtime (i.e. the total throughput time of a job in the production system, which consists of processing time and waiting time) for arriving jobs, and for setting the due date accordingly (see e.g. Ragatz & Mabert, 1984; Cheng & Gupta, 1989; Vig & Dooley, 1991, 1993; Gee & Smith, 1993). However, when compared to other approaches such as mathematical programming or queueing networks, the use of regression analysis to support planning decisions is limited.

1.4 Problem statement and research questions

The research described in this thesis is meant to contribute to the development of models to support order acceptance decisions and to provide insight into the benefits of using regression techniques to support these decisions in stochastic settings in multipurpose batch process industries. In order to meet these goals, the following research questions are used to direct the research project:

1. How do aggregate regression models perform compared with detailed scheduling models, when used to support order acceptance decisions in settings with random order arrivals and processing times?
2. Can the strengths of both aggregate regression-based policies and detailed scheduling-based policies be combined in an improved order acceptance policy?
3. To what extent can regression analysis be used if only limited historical data is available?

The research methodology that is used to answer these questions is discussed in the next section.

Raaymakers (1999) showed that regression modelling provides the decision makers with a powerful and relatively straightforward tool for supporting customer order acceptance decisions in multipurpose batch process industries under deterministic problem settings. Although the performance of her aggregate regression-based policy is surprisingly good in a deterministic situation, there is still considerable difference in performance between the aggregate regression-based policy and a detailed scheduling-based order acceptance policy. Multipurpose batch plants often operate in a dynamic stochastic environment where demand may be both variable and uncertain, and production disturbances may occur. We may expect that the performance of deterministic order acceptance policies will deteriorate under stochastic production conditions. For example, in a stochastic environment, we may expect the performance of the detailed scheduling-based policy to be affected by the uncertainty in the processing times, since the ex ante schedule constructed upon order acceptance is not anymore an exact representation of the future status of the production system. On the other hand, an aggregate regression-based policy may be less sensitive to uncertainty.

These considerations triggered the first research question. It makes sense therefore to develop order acceptance policies that account for processing time uncertainty and to investigate the performance that can be obtained by using such policies under stochastic conditions. Besides measuring their performance, we also aim at understanding how these policies contribute to this performance. Only such an understanding can lead to constructive design of a new, improved policy to answer the second research question. By "improved" we refer not only to improved system performance, but also to the robustness of such a policy, i.e. making it less sensitive to specific production conditions.

While the first two research questions relate to the scientific relevance of this research, the third research question address its practical relevance. Application of

regression-based order acceptance policies in real life requires that sufficient historical data (regarding customer orders and production system) are available to estimate the coefficients of the regression models with acceptable accuracy. In real life, however, there may be only a limited amount of historical data available. It is therefore worthwhile to investigate what is the impact of limited historical data on the performance of the policies, and how this limited data problem can be solved.

1.5 Thesis outline

This section describes the steps that will be followed to answer the research questions stated in the previous section. Six steps are identified, namely:

1. Developing aggregate regression models that account for processing time uncertainty to estimate the production output that can be realized with the available capacity.
2. Evaluating the performance of these models as methods to support order acceptance decisions.
3. Identifying the characteristics that affect this performance.
4. Developing improved models by combining the strengths of both detailed scheduling- and aggregate regression-based policies.
5. Developing a procedure to solve the problem of limited data.

The first two steps are related to research question 1 in the previous section. Research question 2 is addressed in steps 3 and 4. Research question 3 is addressed in step 5.

When developing order acceptance policies, the main issue is to accurately estimate the production output that can be realized with the available capacity. In the first step, we develop aggregate regression-based models that estimate the completion time (or makespan) of a given set of jobs that has to be completed on a given resource configuration. These regression-based estimation models are based on regression analysis and use a few aggregate characteristics of the job set. We conduct controlled computer simulation experiments to determine the relationship between the makespan of a set of jobs and the characteristics of this job set. Computer experiments are used because a large number of different job sets can be evaluated in this way. The parameters in the simulations are varied such that the job sets generated resemble the characteristics of the job sets in industry, as identified in the empirical study of Raaymakers (1999). We use multiple linear regression analysis to determine the relation between the job set characteristics and the makespan of a job set. The development of aggregate regression-based models is presented in Chapter 3.

In the second step, we examine the performance that can be obtained by using such aggregate makespan estimation models, as compared with detailed scheduling-based models, when used to support customer order acceptance decisions. We develop two

order acceptance policies: the regression policy and the scheduling policy. The regression policy uses the aggregate makespan estimation models developed in Chapter 3 to estimate the actual makespan of an order set. The scheduling policy, uses simulated annealing techniques and a statistically determined slack to estimate the actual makespan of an order set. This slack is added in order to cope with the effects of processing time uncertainty. We conduct a simulation study to collect and analyze the performance of these two order acceptance policies. These experiments simulate a hierarchical planning situation with the upper control level deciding about the composition of job sets, and the lower control level scheduling these job sets within the available time frame. The performance evaluation is carried out under a realistic setting, i.e., random order arrivals and processing times. The performance is evaluated based on (i) the ability to control the delivery performance and (ii) the realized capacity utilization. These results are presented in Chapter 4.

In the third step, we perform a detailed analysis of those aspects that influence the performance of both the regression policy and the scheduling policy. Order acceptance procedures have a considerable impact on the mix of jobs that need to be scheduled, by refusing specific jobs from the total demand. However, by refusing jobs with specific characteristics in order to, for example, maximize the resource utilization, an important and often unforeseen side-effect occurs, namely that the mix of orders changes in such a way that the expected delivery reliability is no longer met. We investigate this selectivity property and its impact on the system performance for both policies in Chapter 5.

The insight gained at the third step is used in the fourth step to develop a new order acceptance policy, called the hybrid policy. The hybrid policy combines the strengths of both the scheduling policy and the regression policy. We discuss the development of this policy in Chapter 6. Both simulated annealing techniques and regression analysis are used to develop the hybrid policy. We also evaluate the performance of the hybrid policy and we compare it with the performance of both the regression policy and the scheduling policy. A simulation study is again performed for this evaluation. These experiments are similar to the experiments performed at the second step.

In the fifth step, we investigate to what extent regression analysis can still be used to support customer order acceptance decisions if limited historical data are available. Application in real life of the order acceptance policies we developed assumes that sufficient historical data are available in order to estimate the parameters of the models. However, in real life there may be a limited amount of historical data available, or not all the available data may be relevant. And limited data may not be sufficient to estimate the parameters of the models with acceptable accuracy. We refer to this problem as the limited data problem. For solving the problem of limited data, we use bootstrapping (Monte Carlo re-sampling with replacement of the available data). The performance of this bootstrap procedure is evaluated by simulation. These results are presented in Chapter 7.

In Chapter 8, we conclude this thesis with an overview of our results and directions for further research.

Chapter 2

Production environment

The objective of this chapter is to describe the production environment assumed in this thesis, and to justify the assumptions we made to limit the size - but not the scope - of the research project. This chapter is organized as follows. Section 2.1 considers the production planning framework we assume in this thesis. Section 2.2.1 introduces the order stream assumptions. Section 2.2.2 considers the shop layout. Section 2.2.3 describes the job set scheduling and execution process.

2.1 Planning and scheduling framework

The objective of this thesis is to develop models to support order acceptance decisions in multipurpose batch process industries with random order arrival and processing times. In this section we provide the context in which the order acceptance decisions are made.

We consider the hierarchical planning and scheduling framework developed by Raaymakers (1999) for batch process industries. She distinguishes the following decision functions: capacity adaptation, order acceptance and capacity loading, and resource allocation. Capacity adaptation is a medium- to long-term decision function and deals with adapting the levels of both machine capacity and operator capacity in multipurpose batch process industries. Order acceptance and capacity loading is a medium-term decision function and decides upon accepting or rejecting customer orders and - in case of acceptance - allocating the resulting job to a specific planning period. Resource allocation is a short-term decision function and deals with the operational use of resources. This type of hierarchical framework is quite common, both in industrial and theoretical settings (see, e.g. Hax *et al.*, 1980; Bertrand *et al.*, 1990; Schneeweiß, 1995).

We argued in Chapter 1 that batch process industries are characterized by very complex processes and process structures. In addition, many sequencing constraints dictate the operational scheduling of jobs, such as sequences with no-wait restrictions

among them. These complex characteristics make the overall planning and scheduling problem over the entire planning horizon computationally intractable and lead to a necessity to decompose it. Another reason for decomposing the overall planning and scheduling problem into an aggregate and a detailed problem is that at the time the aggregate decisions are made, detailed decisions for the same period cannot or need not yet be made. For example, order acceptance decisions may need to be made at a moment when not all information is available on the future status of the production system. Decisions on the actual execution of a job are made later, when detailed information is available. Moreover, several production and demand disturbances may occur between the aggregate and detailed decision making times. These disturbances influence the detailed decision.

Figure 2.1: Planning and scheduling framework

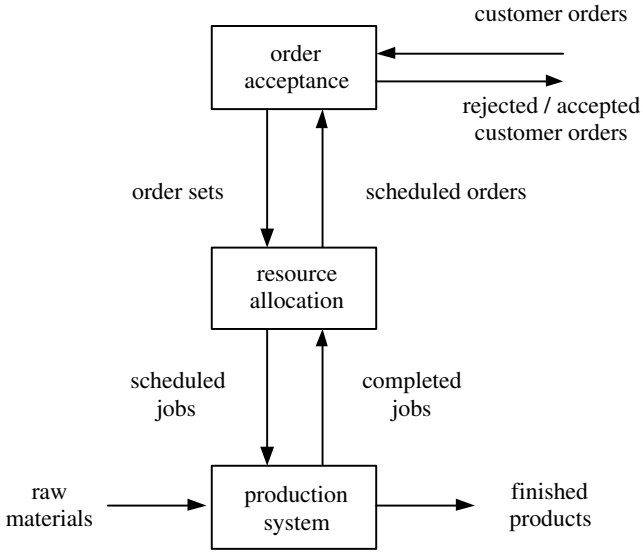


Figure 2.1 displays the hierarchical planning and scheduling framework we consider in this thesis. At the first level (the order acceptance level) there is the planner who accepts or rejects orders that are requested by the market for delivery at the end of a specified planning period. The order acceptance decision is based on the availability of sufficient capacity to complete the order before its requested due date. At the second level (the resource allocation level) there is the scheduler who allocates the processing steps of a job (the accepted order) to specific resources and determines the exact sequence and timing of the (planned) execution of the processing steps on the resources.

An essential characteristic of this production planning framework is the use of planning periods. Planning periods are introduced in order to cope with the distinctive features of multipurpose batch process industries. As we mentioned in Section 1.2.1, no-wait restrictions exists among the processing steps of a job. Due to this feature,

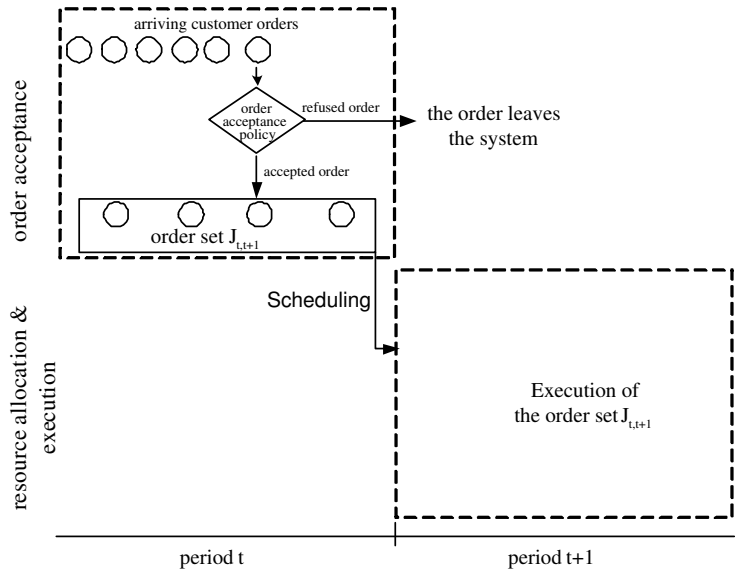
the resource allocation is done by scheduling the jobs. Other methods for allocating resources, such as dispatching, are not suitable because this will not guarantee that all the no-wait restrictions are met. In the literature, job shops are often considered as queuing networks (see e.g. Buzacott & Shantikumar, 1993). Jobs that need to be processed on a specific resource are waiting in a queue until the resource becomes available. Then one job in the queue is dispatched to the available resource based on some priority rule. Although multipurpose batch process industries have much in common with job shops, the latter type of resource allocation is not possible here. In order to meet all the no-wait restrictions, we have to ensure that - prior to the start of the first processing step of a job - all the requested resources will be available at the requested time. This can be best realized by scheduling a number of jobs in the same time. Therefore, the planning horizon (say) H is divided into n_H planning periods or time buckets. A planning period t ($t = 1, ..., n_H$) always starts with an empty system; at the end of each period the system must be empty again. This is a reasonable assumption if there is no production during the weekends. If production is performed round-the-clock, this will lead to some start-up and shut-down losses in each period. If the length of the planning period is sufficiently large, then these losses are expected to be small.

The overall planning and scheduling objective is to maximize the resource utilization, while maintaining a minimum service level. In a hierarchical structure this overall objective is decomposed into separate objectives for each hierarchical level. The objective of the order acceptance level is to determine job sets for each planning period that are achievable and realize high capacity utilization. By "achievable" we mean that the total workload and job mix in the job set is such that at the lower decision level a resource allocation can be made that completes all jobs before their due dates. The delivery reliability and capacity utilization realized are determined by this decision function. The objective of the resource allocation level is to construct a schedule for the released job sets, such that all jobs are completed in that period. Therefore, the criterion used is the minimization of the job set makespan. Jobs are executed by the production system according to this schedule.

2.2 System characteristics

We assume a single production department that produces to customer order. Production to order seems to be a logical choice because of low demand frequency per product and the variability and dynamics in demand in batch process industries. As explained in the previous section, we consider a setting with independent planning periods. During a planning period t, customer orders arrive at the production department at random points in time. We assume that the requested due date is non-negotiable and is equal to the end of the next period, $t + 1$. Each customer order is evaluated on line, immediately upon its arrival. An order is accepted only if sufficient capacity is expected to be available to complete both the new order and the orders already accepted before their requested due date. Orders that fail this test are rejected and leave the system (see Figure 2.2). Let $J_{t,t+1}$ denote the set of orders that are accepted in period t and are due at the end of period $t + 1$.

Figure 2.2: Decision moments in the system



2.2.1 Order characteristics

We assume that each customer order consists of exactly one job ($j, j = 1, ..., |J_{t,t+1}|$). Thus, throughout the remainder of this thesis, *order* and *job* are interchangeable terms. We also assume no precedence relations among the jobs. In industrial practice, however, a number of subsequent operations has to be carried out to produce a product. This means that a customer order may result in several consecutive jobs. However, the empirical study of Raaymakers (1999) indicates that it is common to generate a single job for every operation in the routing of a product. Furthermore, subsequent operations of a product often are performed in different production departments. If we develop a method that supports order acceptance for single jobs, then this can be extended relatively easily to multiple jobs per customer order as long as each job is allocated to a different job set. Therefore, it seems reasonable to assume that each order results in a single job. We also assume that customer orders are not combined, which is reasonable due to the low demand frequency per product.

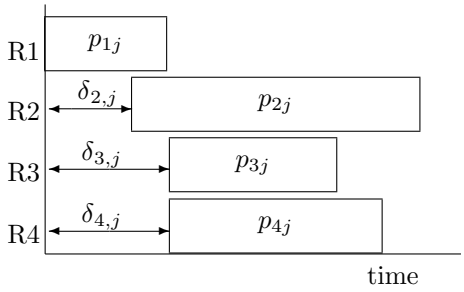
For each job j, a specific number of no-wait processing steps (s_j) is required. No-wait restrictions are generally modelled as finish to start relationships, i.e. each processing step of a job has to start exactly at the time its immediate predecessor is completed (Pinedo, 1995). In this thesis, we use a more general definition of no-wait restrictions, namely the start of the first processing step determines the exact start times of all the other processing steps of the same job. We need this extended definition in order to allow for overlapping processing steps.

In multipurpose batch process industries, the processing steps of a job may have an overlap in time. The overlap is due to the fact that two resources are needed simultaneously. The product being processed is generally a fluid that needs containerization

in order to be stored. If there are two consecutive processing steps, these could be decoupled by storing the product in a silo or bin, but this is often not possible due to the lack of stability of the product. As a consequence, the product is transferred from one resource to the next (e.g., from a reactor to a distillation unit). The product will then occupy both units during some time. Figure 2.3 gives an example of a job that consists of four processing steps.

Each processing step i requires p_{ij} time units on a resource of a specific type. The no-wait restrictions between the processing steps are given by the fixed time delay ($\delta_{i,j}$) between the start time of the processing step i ($i = 2, ..., s_j$) relative to the start time of the first processing step.

Figure 2.3: Job example



We assume that, upon arrival of an order j, the number of processing steps, and the timing and the sequencing constraints on the processing steps are known. Let us now turn to the modelling issues of the processing time's distribution and the planner/scheduler's knowledge of processing times. We assume that, upon order arrival, the planner/scheduler knows for each processing step i of each order j the expected processing time $E[P_{ij}]$ and the probability density function (p.d.f.) of the processing time P_{ij} (we use capital letters to denote random variables, and lower case letters to denote their realizations, e.g. p_{ij}). We further assume that each processing step may be different and may have a different expected processing time $E[P_{ij}]$. We modelled this by considering $E[P_{ij}]$ as a random variable uniformly distributed. Actual production data obtained from a batch chemical processing department in industry showed that the distribution of processing times is close to being Erlang-distributed. Thus, in this thesis, the processing times P_{ij} are assumed to be Erlang distributed with mean $E[P_{ij}]$ and shape parameter k. We also assume the same Erlang shape parameter k for each processing step i of every job j.

Set-up times are assumed to be sequence independent and are therefore included in the processing times. This is a reasonable assumption because of the low production frequency per product.

2.2.2 Shop layout

The shop environment consists of a stable number of resources of some specified type. This is called the *resource configuration*. A resource configuration is determined by the number of resource types M and the number of resources per type $n_m, m = 1, ..., M$. In this thesis, jobs are scheduled on a fix resource configuration. This is a reasonable assumption since the configuration of resources in a production department remains unchanged in the medium term.

The following assumptions regarding the resources are made:

- resources are available from the start of the planning period;
- each processing step has to be performed without preemption on exactly one resource of a specific resource type;
- more than one processing step of a job may require a resource of the same resource type. These processing steps have to be performed on different resources of that type if they overlap; that is, resources of the same type are identical. The empirical study of Raaymakers (1999) indicates that in industrial practice resources are generally not identical, though similar resources can be used as alternatives. Therefore, it seems reasonable to assume that resources of the same type are identical.

2.2.3 Scheduling and execution of job sets

At the start of each period $t+1$, a job set $J_{t,t+1}$ is released to be scheduled and executed by the production system. The scheduler constructs a schedule ($S_{J_{t,t+1}}$) based on the expected processing times, i.e. by assuming deterministic processing times equal to their expectation. The no-wait job shop scheduling problem considered here is NP-hard in the strong sense (Lenstra *et al.*, 1977). We can therefore not expect to find optimal solutions to realistic instances within reasonable time. Heuristic methods are used to obtain near-optimal solutions. We chose for simulated annealing (SA) because it has proven to be an effective local search procedure that can be easily applied to different types of problems (van Laarhoven & Aarts, 1987; Raaymakers & Hoogeveen, 2000). SA is a randomized neighborhood search algorithm that accepts worse-cost solutions with a certain probability (Aarts & Lenstra, 1997). As a result, this provides the opportunity to escape from local optima. A drawback of SA is that considerable computation time is generally required. An overview of the SA algorithm is given in Appendix A and we refer to Raaymakers & Hoogeveen (2000) for further details on the algorithm. The makespan associated with the schedule $S_{J_{t,t+1}}$ is denoted by $C_{\max}^{\text{ex ante}}(S_{J_{t,t+1}})$ and is referred as the ex ante makespan in the remainder of this thesis.

Jobs are released to be executed in the sequence of this schedule. Since the actual processing times are not known until realized, non-feasibility problems may occur during execution - so rescheduling may be needed. The rescheduling procedure used in this thesis is a "right-shift" control policy (Leon *et al.*, 1994) that entails a right-shifting of the schedule in order to restore the feasibility on the resources while always maintaining the original sequence, and can be used for locally revising the schedule in real time. In the remainder of this section, we present our rescheduling procedure.

At the end of the planning period $t+1$, we have the realized schedule and its corresponding actual makespan (or ex post makespan), which is denoted by $C_{\max}(S_{J_{t,t+1}})$. Since the jobs that arrive in a planning period are always due at the end of the next planning period, for simplicity we suppress the subscripts that refer to the planning periods. So, we refer to a job set $J_{t,t+1}$ by J and to the schedule $S_{J_{t,t+1}}$ by S_J. Note that $C_{\max}(S_J)$ is a random variable determined by the schedule and by the individual processing times P_{ij}.

Rescheduling procedure The scheduler monitors the schedule progress and reviews the schedule such that any new information on the actual processing time is taken into consideration. If at a review moment τ, the actual processing time of a particular processing step is larger than the expected processing time, then infeasibility may occur. The right-shift rescheduling procedure delays processing if necessary, while maintaining the processing order of the initial schedule. Therefore, all the processing steps that have a scheduled starting time larger than or equal to τ, are delayed by one time unit - until full information about the actual processing time of that particular processing step becomes available to the scheduler. On the other hand, if the actual processing time is smaller than the expected processing time, then the starting time of all remaining jobs may be decreased while maintaining the no-wait restrictions.

One of the difficulties in no-wait job shop scheduling is that changes in the job sequence on one resource are likely to affect several other resources. If the actual processing times are not exactly known at the time the schedule is determined, it is impossible to satisfy all the no-wait restrictions throughout the job set execution, because the actual processing times only become known over time. Violations of the no-wait restrictions may cause product quality problems, therefore violations are undesirable. This is measured by the feasibility performance, defined as one minus the fraction of processing steps that violate the no-wait restrictions.

To illustrate the proposed rescheduling procedure, an example problem with four jobs to be processed on five machines is shown in Table 2.1.

Table 2.1: Job parameters for the example problem

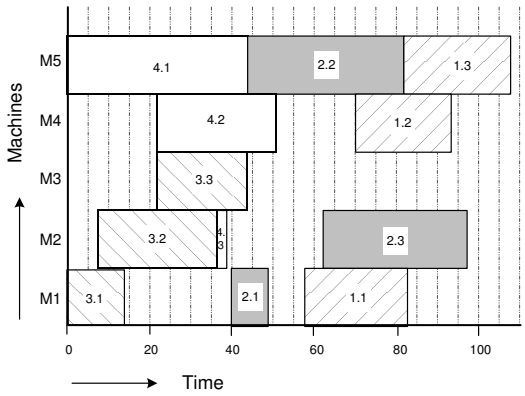
Job	*Processing step*	*Expected processing time*	*Actual processing time*	*Time delay*
1	1	25	36	0
	2	23	24	12
	3	26	28	24
2	1	9	2	0
	2	38	34	4
	3	35	36	23
3	1	14	6	0
	2	29	32	7
	3	22	8	22
4	1	44	38	0
	2	29	25	22
	3	2	2	36

The Gantt chart of the schedule obtained by using simulated annealing algorithm is shown in Figure 2.4(a). Numbers inside the blocks represent the job number and the associated processing steps (e.g., 4.1 means the first processing step of job 4).

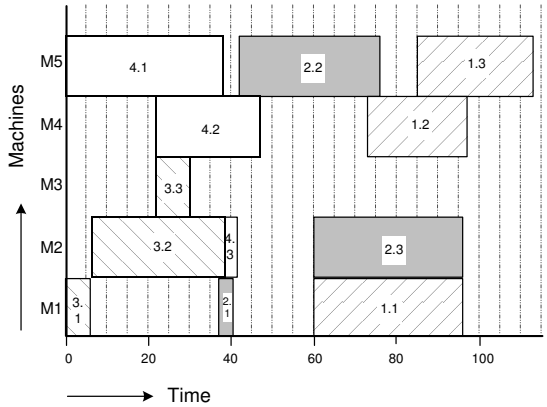
The right-shift procedure was applied to the example problem; the result is shown

in Figure 2.4(b). The first non-feasibility problem occurs when on machine 2, the actual processing time of processing step 3.2 is longer than the expected processing time. Because the completion time of processing step 3.2 was changed, at time $\tau = 36$ the starting time of all the other processing steps that had a scheduled starting time larger than or equal to 36 were delayed - until complete information about the actual processing time of processing step 3.2 is known. Next, at time $\tau = 38$, the actual processing time of processing step 4.1 is less than the expected processing time, so, the starting time of job 2 is decreased (from 40 to 38) such that the no-wait restrictions are not violated. Finally, at time $\tau = 39$, the processing step 3.2 is completed, and the processing step 4.3 can start. But delaying the starting time of the processing step 4.3 (from 36 to 39) violates the no-wait restrictions within job 4.

Figure 2.4: Gantt chart for the example in Table 2.1



(a) Ex ante schedule



(b) Ex post schedule

Chapter 3

Predicting the makespan of a job set

3.1 Introduction

In the hierarchical planning and scheduling framework described in Chapter 2, planners are responsible for balancing the capacity requirements (due to customer orders) and the available capacity on medium-term. They plan the various jobs over time and load them into time buckets and onto processing departments. Therefore, the planners are expected to make an adequate assessment of the feasibility of the job set - which requires a fairly detailed insight into the constraints on the shop floor in the processing departments. In addition, the planners have to be able to estimate if the jobs in the job set can be delivered on time. In a situation with discrete planning periods, this means being able to predict the makespan of a job set.

Raaymakers (1999) showed that, in settings with deterministic processing times, accurate estimates for the makespan of a job set may be obtained by using a regression model with a limited number of aggregate job set and resource characteristics, in addition to the workload of the job set. In this chapter [1], we extend this work and investigate whether similar aggregate job set characteristics may be used for predicting the makespan of a job set in settings with stochastic processing times.

The structure of this chapter is as follows. In Section 3.2 we provide an overview of the literature on makespan estimation. In Section 3.3 we discuss the formulation of the prediction model. In Section 3.5 we present the regressors of the prediction model. In Section 3.4 we address the data generation process. In Section 3.6 we elaborate on the building and evaluation of the regression models. In Section 3.7 we present our conclusions.

[1] Earlier versions of the content of this chapter and Chapter 4 are joint work with J.C. Fransoo and J.M.W. Bertrand, and have been published in Ivanescu *et al.* (2002) and Ivanescu *et al.* (2003a).

3.2 Review of literature on makespan estimation

The literature on makespan estimation is limited. Fransoo *et al.* (1994) made a first attempt to estimate the job set makespan for a flexible manufacturing system (FMS) by using queuing models. For each period a makespan estimate is provided for the job set to be processed, based on the number of jobs, the average throughput time per job, and the number of product carriers in the system. Simulation experiments showed a low correlation between estimated and realized makespan ($R^2 = 0.27$). This may be caused by the fact that queueing models, that describe long term stationary behavior, are used to estimate short term behavior that may not be stationary. The interesting part of their research, however, is the idea to use makespan estimation - based on aggregate models - in order to estimate the likeliness of completing a job set within a given period.

Raaymakers (1999) addressed the makespan estimation problem in multipurpose batch process industries. She developed aggregate models for predicting the makespan of a job set assuming deterministic processing times, in a setting with discrete planning periods. She identified five aggregate characteristics of the job set and resources that influence the completion time of a job set: (1) the average number of processing steps per job, (2) the average overlap of processing steps, (3) the standard deviation of the processing time, (4) the average number of identical resources per type, and (5) the workload balance per resource type. She used these characteristics, in addition to the workload of a job set, to accurately predict the makespan of a job set, by means of multiple linear regression analysis.

Related to the job set makespan estimation is the problem of estimating the time required to perform a given set of activities (i.e. a project), which has been investigated in the context of project management and scheduling since the early papers proposing the Program Evaluation and Review Technique (PERT) (Battersby, 1967). A PERT network, also known as a stochastic activity network, is based on the concept that a project is divided into a number of activities that are carried out concurrently subject to precedence constraints and limitations on some critical resources (such as skilled labor, equipment, utilities). Two major problems may be distinguished: modelling the duration of each activity, and studying the distribution of the total duration of a project under uncertainty conditions. Determination of the exact distribution of the total duration of a project is complicated by the fact that different paths in the network are correlated, and also because of the need to find the maximum of a set of random variables (Dodin, 1985). Three major approaches have been developed to overcome this problem: (1) approximate estimation of the distribution of the total duration; (2) computing lower or upper bounds for this distribution; and (3) estimation of bounds for the expected project total duration (Tavares, 1999). Exact methods are not easy to find, therefore simulation methods are use to obtain desired statistics for networks with specified distributions for the activities. Tavares *et al.* (1999) studied the statistical distribution of the total duration of a project for small networks under specific assumptions regarding the distribution of the duration of each activity. By using six indicators of the morphology of the network, they developed a model to predict the statistical parameters of the total duration of a project.

A companion of the problem of makespan estimation is the problem of estimating the flowtime of jobs consisting of a linear structure of processing steps, which has received considerable attention in the literature. The flowtime is the total throughput time of a job in a production system, which consists of processing time and waiting time. Because flowtime estimation is used to assign order due dates, the problem has been mostly studied in the context of due date assignment. Most due date literature concentrates on establishing flow allowances for arriving jobs, in order to accurately reflect their actual completion times and setting the due date accordingly. For an overview of the results published up to the end of the 1980s we refer to Cheng & Gupta (1989), while we refer to Gordon *et al.* (2002) for a review of more recent results.

Two basic approaches have been used to establish the relationships needed to predict flowtimes: the analytical approach and the simulation approach. There are advantages and disadvantages associated with each approach. The analytical approach, usually based on queuing theory, proposes an exact way of determining mean and variances of flowtime estimates. However, the dynamic and stochastic nature of production systems makes it difficult to develop realistic analytical models. On the other hand, a simulation approach may need many computer runs to obtain accurate and precise estimates. Since these two approaches are complementary in nature, the literature has been developed in both directions. The approach used in this thesis is simulation. Thus, we next discuss in more detail the simulation related literature. For the analytical studies, we refer to Miyazaki (1981), Baker & Bertrand (1981b), Baker & Bertrand (1981a), Bertrand (1983a), Bertrand (1983b), Cheng (1985), Cheng (1986), Adam *et al.* (1993), Lawrence (1995), and Bertrand & van de Wakker (2002).

Research using the simulation approach is due to Conway (1965), Eilon & Chowdhury (1976), Weeks (1979), Ragatz & Mabert (1984), Fry *et al.* (1989), Enns (1993, 1994, 1995), Gee & Smith (1993), and Vig & Dooley (1991, 1993). Most of these studies used simulation to generate innitial data required to develop the flowtime relationships used in further experimentation. Conway (1965) compares four flowtime estimation methods: total work content (TWK), number of operations (NOP), constant (CON), and random (RDM). The results of this study indicate that the methods that utilize job information perform better than the others. Eilon & Chowdhury (1976) use shop congestion information to estimate flowtimes. In their work, TWK is compared with three other methods: jobs in queue (JIQ), delay in queue (DIQ), and modified total work content (MTWK). They used regression analysis to establish due-date rule parameters in order to minimize the deviations between the completion times and due-dates. Their results indicate that JIQ, which employs shop congestion information, outperforms other methods. Weeks (1979) proposes a method that combines job and shop information. This method performs very well for performance metrics such as mean lateness, mean earliness, and number of tardy jobs.

Ragatz & Mabert (1984) tested a number of due-date setting procedures to determine which one predicts most accurately the flowtimes. Regression analysis was again used. They concluded that using shop load data related to individual job's routings was more beneficial than using overall shop load data.

Fry *et al.* (1989) also investigate the job and shop characteristics that affect a job's

flowtime in a multistage job shop. They construct two linear and two multiplicative nonlinear models to estimate the regression coefficients of the factors. This study shows that (i) models using product structure and shop conditions estimate flowtimes better than the others, (ii) their linear models are superior to their multiplicative models, and (iii) the predictive ability of the models improves as the utilization increases.

Enns (1993) and Vig & Dooley (1991) examine dynamic models of varying complexity to predict flowtimes. Vig & Dooley (1993) note that due date setting with dynamic shop information tends to be more sensitive to changes in the environmental conditions, resulting in nervousness (over- or under-compensation). They proposed a weighted approach that combines static information - which enhances robustness of due dates - with dynamic information - which lead to improved accuracy. Enns (1995) proposes a forecasting model based on dynamic shop load information. In this model, due dates are set by monitoring the distribution of the flowtime estimation errors and by providing safety allowances for target levels of delivery performance. His results suggest that for a desired service level, due date based dispatching minimizes lead time.

Gee & Smith (1993) propose an iterative procedure for estimating flowtimes when due date dependent dispatching rules are used. Two flow time estimation methods are employed, one based on job related information and one based on both job and shop related information. Their results indicate that the second method yields better results.

From this literature review on flowtime estimation and due date setting we conclude that both job and shop information need to be considered. A common characteristic of the papers we mentioned above is that flowtime estimation rules are developed for a job shop system with queues in front of resources. In such a situation, the jobs are immediately released to the shop floor and processed according to some dispatching rules. In our situation, however, queueing is not possible due to no-wait restrictions between the processing steps of a job. As we explained in Chapter 2, we assume that a set of jobs is periodically scheduled and released to the shop floor. Consequently, we do not concentrate on the completion time of individual jobs but on the makespan of a set of jobs. In this respect, our approach to makespan estimation is related to the approach of Fransoo *et al.* (1994), and builds on the work of Raaymakers (1999).

3.3 Prediction model formulation

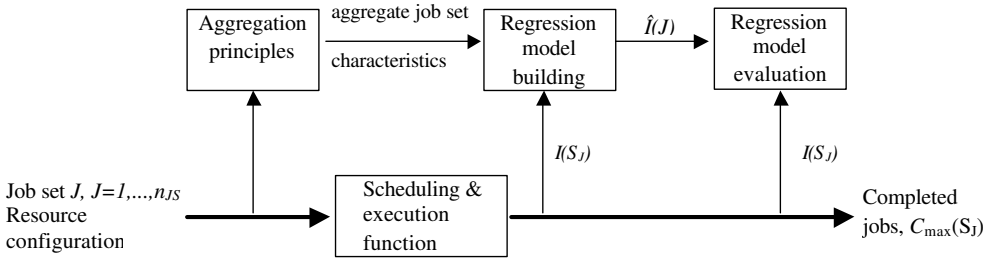
The makespan of a job set is clearly influenced by its workload. The workload of a job set is the quantity of capacity needed to complete the job set. More specifically, the workload on the bottleneck resource type puts a lower bound on the makespan. The bottleneck resource type is the resource type that has the highest utilization. This lower bound - which we denote by $LB(J)$ - is a single resource lower bound on the makespan which is computed for each job set by dividing the workload on the bottleneck resource type by the number of resources of that type (Carlier, 1987). We may round this value upwards, because all processing times are integer values and no pre-emption is allowed.

However, because of job interactions (timing and no-wait sequencing constraints) at the scheduling level, the minimal makespan for which a feasible schedule is realized will often exceed this lower bound. Job interaction results from relations between capacity requirements on different resources and from scarcity of capacity. The capacity requirements for different resources have a fixed offset in time for each job, due to the fixed time delay for each processing step (see Figure 2.3). To obtain a feasible schedule, some idle time on the resources can generally not be avoided. The job interaction is measured by the *interaction margin*. For a given schedule S_J associated to a job set J, the interaction margin - denoted by $I(S_J)$ - is defined as the relative difference between the realized makespan and the lower bound (Raaymakers, 1999):

$$I(S_J) = \frac{C_{\max}(S_J) - LB(J)}{LB(J)} \tag{3.1}$$

Given a job set we are interested - without actually constructing the schedule for the job set - in estimating the amount of job interaction that is expected to be realized if the schedule would have been constructed. Obviously, given a job set J, the exact amount of job interaction cannot be determined without the actual construction of the schedule. Therefore, following the work of Raaymakers (1999), we determine an estimate for the average interaction margin. We use a multiple linear regression model to predict the interaction margin - the response variable - based on a limited number of aggregate job set characteristics - the predictor variables or regressors. Figure 3.1 schematically shows the development of our prediction model.

Figure 3.1: Development of the prediction model



We assume the existence of a number of job sets, say n_{JS}, that have been scheduled and executed on a given resource configuration. These may be real (historical) data of the production department or simulated data, obtained from a pilot simulation study. Although the problem we address is inspired by real life, we develop models based on simulated data. Job set generation is discussed in Section 3.4. For the scheduling and execution of these job sets, we use the simulated annealing algorithm and the rescheduling procedure described in Section 2.2.3.

For each job set J, $J = 1, ..., n_{JS}$, we determine the aggregate job set characteristics, i.e. the regressors in our regression model. Furthermore, the result of scheduling and execution of the job set is the ex post makespan $C_{max}(S_J)$ which allows us to determine the interaction margin (see equation 3.1) - our response variable. We

determine the regression model using a two-step process that can be found in most statistics textbooks (e.g. Montgomery & Peck, 1992): model building and model evaluation. In the model building step, a regression equation is determined that provides the best fit to the available data. In the remainder of this thesis, these data will be called the *construction data set*.

In the model evaluation step we distinguish between model adequacy checking and model validation. Model adequacy checking investigates the fit of the regression model to the construction data set using residual analysis. However, there is no assurance that the equation that provides the best fit to these data will be a successful predictor. Therefore, the predictive performance of the model has to be tested. The cross validation method is employed (or data splitting according to Montgomery & Peck, 1992) for evaluating the model. In the cross validation method, a sample of approximately 80% is taken from the available data for obtaining the construction data set, whereas the rest (20%) is used to test the model. The latter set of data will be called the *testing data set* in the remainder of this thesis.

Through the estimated regression equation we can predict the average interaction margin for a given job set J. This estimate may be used to predict the expected makespan of the job set. Formally, this may be expressed as follows:

$$\widehat{C}_{\max}(J) = (1 + \hat{I}(J))LB(J) \tag{3.2}$$

where the hat denotes an estimate of a variable. Note that both $\widehat{C}_{\max}(J)$ and $\hat{I}(J)$ depend only on the job set, since a schedule S is not yet constructed.

3.4 Data generation

In this section we discuss the data generation process. The setting for our research is a hypothetical production department in the batch process industry. The simulated shop consists of five resource types, with two identical resources per type. Two reasons determined the choice of this particular resource configuration. First, regarding the size (10 resources), this is a realistic size of a production department (Raaymakers, 1999). Second, the size of the scheduling problem remains reasonable with respect to the computational time for the simulation experiments.

The following factors are considered for generating the job sets: the number of jobs in the job set n_J, the number of processing steps per job s_j, the overlap of the processing steps g_j, the probability that the processing steps are executed on a particular resource type p_m $(m = 1, 2..., 5)$, and the probability distribution function which gives the expected processing times $F_{E[p]}$. The number of jobs in the job set n_J is obtained by drawing a number from the uniform distribution on the interval $[25, 65]$, rounded upwards to an integer value. The values for the lower and upper bounds of the uniform distribution are chosen such that the job sets consist of a realistic number of jobs (see Raaymakers *et al.*, 2000a).

The rest of the experimental factors are varied at two levels, as presented in Table

3.1. A full factorial design is used to generate the job sets, so eight combinations of different factor levels are possible. For each combination, 50 job sets are generated. This results in a total of 400(= n_{JS}) job sets. These job sets form the core of the data set.

Table 3.1: Experimental factors levels for generating job sets

Factors	*L*	*H*
s_j	$U(4,7)$ *	$U(1,10)$
p_m	$0.3, 0.25, 0.20, 0.15, 0.10$	0.2 for $m = 1, ..., 5$
$F_{E[p]}$	$U(15,35)$	$U(1,49)$

* $U(a,b)$ denotes the uniform probability distribution on the interval $[a,b]$

Each job j ($j = 1, ..., n_J$) requires s_j processing steps determined according to an uniform probability distribution (see Table 3.1). The overlap of the processing steps is obtained by determining the time delay ($\delta_{i,j}$) between the start time of processing step i relative to the start time of job j. The time delay is determined as follows. A number r between 0 and 1 is randomly generated. Then, the time delay of the current processing step is the time delay of the previous processing step plus r times the processing time of the previous processing step. The time delay is then rounded upwards to an integer value.

The processing time P_{ij} for each processing step i of a job j is generated by a two-step sampling procedure. First, a value is generated from the distribution $F_{E[p]}$. This value represents the expected processing time of the processing step i of the job j ($E[P_{ij}]$). Given that a deterministic schedule is constructed by using these values, we round them up to an integer for computational easiness. Second, another value is generated by sampling from an Erlang distribution with the mean equal to $E[P_{ij}]$ and the shape parameter k. The result is made integer to give the actual number of time units required for the processing step. The processing times P_{ij} are independent identically distributed random variables.

We model different levels of uncertainty in the processing times by considering nine levels for the Erlang shape parameter, from 2 to 10. Each job set J ($J = 1, ..., 400$) is scheduled and executed by the production department. All the processing times in J are stochastic (e.g., Erlang distributed with the same shape parameter k). A single simulated execution of the job set would not be sufficient to properly capture the effect of stochastic processing times when developing the regression models. Thus, we repeat the execution of each job set several times. Additional experiments showed that, for each Erlang shape parameter k, 250(= n_{repl}) replications are necessary to reduce the variability in the results. This would result in a $400 \times 9 \times 250 = 900\ 000$ observations. In order to keep the size of the data set at a manageable level, we consider only one uncertainty level for each job set. Moreover, it is realistic to assume that different job sets may experience different levels of uncertainty at the shop floor. Consequently, we randomly allocate an Erlang shape parameter to each job set. This resulted in a data set with a total of $400 \times 250 = 100\ 000$ observations.

We randomly split the core data set (i.e. the 400 job sets) into two parts: about

80% of the data - namely 319(= n_{JS}^{cons}) job sets - form the construction data set whereas the remaining 81(= n_{JS}^{test}) job sets form the testing data set. We chose to split the core of the data set and not the whole data set, because the latter contains replicates of the same job set characteristics. If these replicates were not eliminated, the construction and testing data sets would be quite similar and this would not necessarily test the models severely enough. In conclusion, the construction data set contains $319 \times 250 = 79\ 750$ observations, whereas the testing data set contains $81 \times 250 = 20\ 250$ observations.

3.5 Regressors in the prediction model

Raaymakers (1999) identified the following job set characteristics that proved to be responsible for the difference between the minimal makespan and the lower bound in a setting with deterministic processing times: the number of identical resources per resource type, the workload balance of resource types, the average number of processing steps per job in a job set, the average overlap of processing steps in a job set, and the standard deviation of processing times. These aggregate job set characteristics - referred as *interaction margin variables* in the remainder of this thesis - will play the role of regressors in the linear regression model. Note that since we consider a fixed resource configuration, the number of identical resources per resource type interaction margin variable will have a constant value, so it is not further considered in our study. In addition, we consider the number of jobs in the job set to be one of the regressors.

Contrary to the deterministic setting of Raaymakers, two types of variation should be consider with respect to the processing times. The first type is due to the fact that each generated job may be different, so each processing step within the job set may be different. We measure this variation by the expected processing times variation in the job set $cv^2_{E[p]}$, which indicates the dissimilarity degree of the processing steps with respect to the expected processing times. The second type of variation is due to the randomness of the processing times (i.e. the actual (realized) processing times may differ from their expected value). Therefore, we include the squared coefficient of variation of the actual processing times cv_p^2 to be one of the regressors.

For reasons of self-containedness, we briefly discuss each of the interaction margin variables, and we refer to Raaymakers (1999) for details. Note that, although these variables are determined for a given job set J, we suppress the dependence of these variables on J for notational simplicity.

Workload balance of resource types Scarcity of capacity is a cause for job interactions. If excess capacity exists for some resource types, it is expected that there will be less job interactions, because a feasible schedule for these resources may be easily constructed. However, if capacity requirements are equally high for all resource types, many job interactions are expected to occur, because each of the resources may become bottleneck. Therefore, the balance of the workload distribution over different resource types is expected to influence the interaction margin. As a measure

of workload balance we use the maximum utilization if the makespan is equal to the lower bound. The maximum utilization ρ_{max} of a job set gives the utilization realized if a feasible schedule is constructed with a makespan equal to the lower bound $LB(J)$:

$$\rho_{\max} = \frac{\bar{L}}{LB(J)} \tag{3.3}$$

where

$$\bar{L} = \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{n_m} L_n \tag{3.4}$$

and N is the total number of resources:

$$N = \sum_{m=1}^{M} n_m \tag{3.5}$$

The number of jobs in the job set The flexibility at the scheduling level increases with a higher number of jobs in the job set. This may influence the interaction margin. Therefore, we consider the number of jobs in the job set n_J to be one of the regressors in the prediction model.

Average number of processing steps per job in a job set Job interaction arises because several resources are required simultaneously or successively for each job. It may be expected that jobs with a larger number of processing steps cause a higher interaction margin, since they require more resources. The average number of processing steps μ_s is defined as follows:

$$\mu_s = \frac{1}{n_J} \sum_{j=1}^{n_J} s_j \tag{3.6}$$

Average overlap of processing steps in a job set In process industries the processing steps may have an overlap in time (see Section 2.2.1). Jobs may differ in the amount of overlap between processing steps. The amount of overlap between processing steps influences the time between the start and the completion of a job. For each job j the overlap g_j is computed as follows:

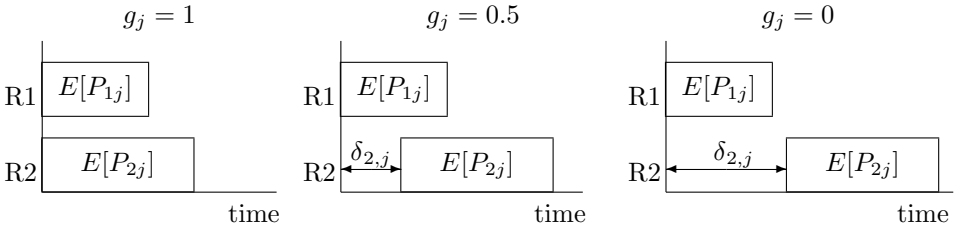
$$g_j = \frac{1}{s_j - 1} \sum_{i=1}^{s_j} (1 - \frac{\delta_{i,j} - \delta_{i-1,j}}{E[P_{i-1j}]}) \tag{3.7}$$

Figure 3.2 illustrates two jobs with the same number of processing steps and processing times, but different overlap.

Next, the average overlap of processing steps in a job set is obtained as follows:

$$\mu_g = \frac{1}{n_J} \sum_{j=1}^{n_J} g_j \tag{3.8}$$

Figure 3.2: Job overlap example



Expected processing times variation in a job set The lower bound on the makespan is based on the workload of the resource types, but does not take into account the variation of the expected processing times in the job set. This variation arises as each job may be different, so each processing step may also be different and may have a different expected processing time. This variation indicates the dissimilarity of the processing steps in the job set and may influence the interaction margin. We measure this variation by the expected processing times variation in a job set $cv^2_{E[p]}$, defined as follows:

$$cv^2_{E[p]} = \frac{\sigma^2_{E[p]}}{\mu^2_{E[p]}} \tag{3.9}$$

where $\mu_{E[p]}$ denotes the average of the expected processing time over all processing steps in the job set:

$$\mu_{E[p]} = \frac{1}{S} \sum_{j=1}^{n_J} \sum_{i=1}^{s_j} E[P_{ij}] \tag{3.10}$$

and $\sigma^2_{E[p]}$ denotes the variance of the expected processing times in the job set:

$$\sigma^2_{E[p]} = \frac{1}{S-1} \sum_{j=1}^{n_J} \sum_{i=1}^{s_j} (E[P_{ij}] - \mu_{E[p]})^2 \tag{3.11}$$

S denotes the total number of processing steps in the job set:

$$S = \sum_{j=1}^{n_J} s_j \tag{3.12}$$

Squared coefficient of variation of the processing times The actual (realized) processing times p_{ij} may differ from their expected value $E[P_{ij}]$, therefore, we may expect that this uncertainty to influence the interaction margin. The squared coefficient of variations of the processing times cv^2_p is computed as follows:

$$cv_p^2 = \frac{(b-a)^2}{3 \cdot (b+a)^2} + \frac{4 \cdot (b^2 + a \cdot b + a^2)}{3 \cdot k \cdot (b+a)^2} \tag{3.13}$$

where a and b are the lower and upper bounds of the uniform distribution for the expected processing time (see Table 3.1), and k is the shape parameter of the Erlang distribution. The derivation of cv_p^2 is included in Appendix B, along with the density function and the first two moments of the processing times.

3.6 Makespan estimation model: building and evaluation

3.6.1 Interaction margin estimation models

The six interaction margin variables we discussed in Section 3.5 are used in this section to construct the regression models. These models are generated by means of multiple linear regression techniques, as follows. We used the SPSS statistical software package for our analysis.

First, we construct an estimation model (A) that contains only the main effects of the interaction margin variables. We use backward regression to eliminate variables that did not show a significant contribution. Second, we construct estimation models that included two-way interactions. The number of possible two-way interactions is 15 and hence, the number of regressor variables is 21. We use stepwise regression to determine the estimation models. All the estimation models are determined using the ordinary least squares (OLS) technique.

We examine the regression models for multicollinearity, because a high degree of multicollinearity makes the parameter estimates in the model not stable. Multicollinearity exists whenever an independent variable is highly correlated with one or more of the other independent variables and it can be detected by using *variance inflation factors* (VIF's). The variance inflation factor for the r-th regressor variable is determined as follows (Montgomery & Peck, 1992):

$$VIF_r = \frac{1}{1 - R_r^2}, \tag{3.14}$$

where R_r^2 is the coefficient of multiple determination obtained by regressing the r-th regressor variable on the other regressor variables. In other words, the value for R_r^2 indicates if there is a strong linear relationship between regressor variable r and the other regressor variables. VIFs larger than 10 imply serious problems with multicollinearity. Among the regression models found by stepwise regression we only consider those models that do not have VIFs exceeding 10.

To test the adequacy of the regression models, we perform a residual analysis. The standardized residuals versus the standardized predicted values plots presented in Appendix C.1 show the tendency of increased variability as the dependent variable

increases. This indicates that the variances of the errors are not constant (i.e. heteroscedasticity). The assumption of constant variance is a basic requirement of OLS. It is important to detect and correct a nonconstant error variance. If this problem is not eliminated, the OLS estimators will still be unbiased, but they will no longer have the minimum variance propriety. The usual approach for dealing with inequality of variance is to apply a suitable transformation to the response variable. A linear regression model is then fitted, and its validity is tested, provided that the necessary assumptions regarding residuals apply *after* transformation. This process is described in most statistics books (see e.g. Montgomery & Peck, 1992). We apply the natural logarithm transformation to the interaction margin variable. Let us denote the transformed interaction margin variable by:

$$Y = \ln(I(S_J)). \tag{3.15}$$

We then fit a multiple linear regression model for this new response variable. Diagnostic checks on the subsequent models confirm the appropriateness of this transformation.

Table 3.2 gives the name, the regressor variables, the adjusted coefficient of multiple determination (adj. R^2) and the standard error of regression ($\widehat{\sigma}$) of the subsequent regression estimation models. The latter two statistics are measures of fit of the regression model.

Table 3.2: Log-transformed interaction margin estimation models

Model	*Regressor variables*	*adj.* R^2	$\widehat{\sigma}$
A	$\mu_s, \mu_g, cv^2_{E[p]}, cv^2_p, \rho_{\max}, n_J$	0.77	0.107
B	$cv^2_p \cdot \rho_{\max}$	0.36	0.177
C	$cv^2_p \cdot \rho_{\max}, \mu_s \cdot \rho_{\max}$	0.61	0.138
D	$cv^2_p \cdot \rho_{\max}, \mu_s \cdot \rho_{\max}, cv^2_{E[p]} \cdot n_J$	0.74	0.113
E	$cv^2_p \cdot \rho_{\max}, \mu_s \cdot \rho_{\max}, cv^2_{E[p]} \cdot n_J, \mu_s \cdot cv^2_{E[p]}$	0.75	0.111

The coefficient of multiple determination (R^2) measures how much of the total variation is explained by the estimated regression equation. R^2 is defined by

$$R^2 = 1 - \frac{\sum_{i=1}^{n^{cons}} (Y_i - \widehat{Y}_i)^2}{\sum_{i=1}^{n^{cons}} (Y_i - \overline{Y})^2} \tag{3.16}$$

where n^{cons} is the number of observations: $n^{cons} = n^{cons}_{JS} \cdot n_{repl}$. However, R^2 always increases when variables are added to the model. Therefore, we use the adjusted R^2, which corrects for the number of variables d in the model:

$$\text{adj. } R^2 = R^2 - \frac{d(1 - R^2)}{n^{cons} - d - 1} \tag{3.17}$$

Whereas the adj. R^2 is a measure of relative fit, the standard error of regression ($\widehat{\sigma}$)

is an absolute measure of the fit of a model because its value is dependent on the scale of the response variable. This $\widehat{\sigma}$ specifies the amount of error incurred when the least-squares regression equation is used to predict values of the dependent variable:

$$\widehat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n^{cons}} (Y_i - \widehat{Y}_i)^2}{n^{cons} - d - 1}} \tag{3.18}$$

The smaller the standard error of regression, the closer the estimate is likely to be to the actual value of the dependent variable.

For each of the estimation models, we use an F-test to test the significance of the model as a whole. The null hypothesis tested by this F-test is that all coefficients of the regression model, except for the intercept, are equal to zero. For the models in Table 3.2, the P-value is 0.000, indicating that this null hypothesis is rejected.

Except for models B and C, the performance measures (adj. R^2 and $\widehat{\sigma}$) look quite good, so we consider only the models A, D and E for further analysis. Table 3.3 shows the regression parameters' least squares estimates for these three models. We perform a two-tailed t-test to test the significance of the individual regressor variables. The null hypothesis tested by this test is that the individual coefficient is either significantly higher or lower than zero. Table 3.3 also gives the t-statistic value (see the values in parentheses). A comparison of the absolute values of the t-statistic with $t_{1-\alpha/2,n-(d+1)}$ (i.e. the $1-\alpha/2$ upper critical point of the t distribution with $n-d-1$ degrees of freedom) implies that all the regressor variables in Table 3.3 are significant, at $\alpha = 0.01$ level of significance.

Table 3.3: OLS parameters' estimates (t-statistic in parentheses)

Regressors	*Model*		
	A	*D*	*E*
Intercept	-1.984 (-210.854)	-1.152(-343.590)	-1.130(-343.417)
μ_s	0.111 (99.586)		
μ_g	-0.130 (-12.604)		
$cv^2_{E[p]}$	-0.883 (-208.429)		
cv^2_p	0.911 (347.897)		
$\rho_{\max}$	1.466 (282.278)		
n_J	-0.003 (-79.485)		
$cv^2_p \cdot \rho_{\max}$		1.020(338.379)	1.085(347.491)
$\mu_s \cdot \rho_{\max}$		0.177(236.292)	0.172(235.115)
$cv^2_{E[p]} \cdot n_J$		-0.016(-197.854)	-0.009(-64.782)
$\mu_s \cdot cv^2_{E[p]}$			-0.083(-62.452)

The predictive performance of the A, D and E models is further evaluated through the testing data set. The quality of these models was quantified by the mean prediction error (ME) and square root of the mean square prediction error ($\sqrt{MSE}$). Additionally, the percentage of variability in the new data explained by the model

(R^2_{pred}) is compared with the R^2 of the building model, where R^2_{pred} is computed as follows:

$$R^2_{pred} = 1 - \frac{\sum_{i=1}^{n^{test}} (Y_i - \widehat{Y}_i)^2}{\sum_{i=1}^{n^{test}} (Y_i - \overline{Y})^2} \tag{3.19}$$

where n^{test} is the number of observations: $n^{test} = n^{test}_{JS} \cdot n_{repl}$.

The results are presented in Table 3.4. The mean prediction error is nearly zero for all the considered models, so the estimation models produce unbiased predictions. Table 3.4 reveals that the adj. R^2 for the new data explained by the model is less than the adj. R^2 for the construction phase. Furthermore, the $\widehat{\sigma}$ values in Table 3.2 (which may be thought of as the standard deviation of the residuals from the fit) is smaller than $\sqrt{MSE}$. This indicates that the regression models do not predict new data as good as they fit the existing data. However, the degradation in performance is not severe.

Table 3.4: Predictive quality of the log-transformed interaction margin regression models in the testing data set

Model	ME	$\sqrt{MSE}$	R^2_{pred}
A	0.02	0.111	0.72
D	0.00	0.115	0.71
E	0.01	0.115	0.71

We conclude that our regression models are likely to provide good predictions for new observations. Although the three models have similar predictive performance, model A is slightly better, using as criteria MSE and R^2_{pred}. However, comparison of the prediction performance of the makespan estimation models on the testing data set will provide us with a basis for final model selection. This is address in the next section.

3.6.2 Makespan estimation models

The models developed in the previous section are used for predicting the mean interaction margin, given a job set and a resource configuration. In this section, makespan estimates are obtained by using a lower bound on the makespan and an interaction margin estimate, as described in Section 3.3 (see equation 3.2)

The interaction margin estimation models provide predicted values for the interaction margin in a log-transformed scale. Therefore, to obtain makespan estimates, it is necessary to convert the predicted values back to the original units. Unfortunately applying the inverse transformation directly to the predicted values gives an estimate of the median of the distribution of the response instead of the mean (Montgomery & Peck, 1992). Thus, following Miller (1984), we introduce the adjustment factor $\widehat{\sigma}^2/2$

to remedy this bias:

$$\widehat{I(J)} = e^{\hat{Y}+\hat{\sigma}^2/2} \tag{3.20}$$

where $\hat{Y}$ denotes the estimate of the mean of the transformed interaction margin.

The quality of the makespan estimation models is evaluated through both the construction and the testing data sets. Two characteristics of importance are investigated when the "quality" of a makespan estimate is considered: accuracy and precision. Accuracy refers to how close, on average, the individual estimates are to their true values. This is the same quantity as the expected value of the estimation errors (ε), which are defined as the difference between the actual (ex post) makespan and the estimated makespan:

$$\varepsilon = C_{\max}(S_J) - \widehat{C}_{\max}(J). \tag{3.21}$$

Precision refers to the variability of the estimation errors. These two characteristics are quantified by the mean estimation error (ME) and the standard deviation of the estimation error (SDE). The results are presented in Table 3.5.

Table 3.5: Quality of the makespan estimation models

	Construction data set		*Testing data set*	
Model	*ME*	*SDE*	*ME*	*SDE*
A	-0.14	74.08	11.83	68.70
D	-2.72	76.10	-2.78	70.12
E	-2.38	68.04	0.68	70.21

Table 3.5 suggests that the three makespan estimation models have similar performance levels. However, we chose model A for further use to support customer order acceptance decisions in a situation with random order arrivals. Two reasons determined our choice. First, recall that for the interaction margin, model A proved to have a slightly better performance. Second, simpler models, including only main effects, are easier to understand and may be more robust (Raaymakers, 1999).

3.7 Conclusions

Previous research (Raaymakers, 1999) showed that, in settings with deterministic processing times, accurate estimates for the makespan of a job set may be obtained by using a limited number of aggregate job set and resource characteristics. In this chapter, we extended this work by considering Erlang distributed processing times. We investigated the statistical relationship between the interaction margin (i.e. the relative difference between the realized makespan of a job set and the lower bound on the makespan) and the following aggregate job set characteristics:

- workload balance of resource types $\rho_{\max}$,
- average number of processing steps per job in the job set μ_s,
- average overlap of processing steps in the job set μ_g,
- expected processing times variation in the job set $cv^2_{E[p]}$,
- the number of jobs in the job set n_J,
- squared coefficient of variation of the processing times cv^2_p.

Simulation showed that there is little doubt that linear regression between the interaction margin and the regressor variables is statistically discernible. As in the deterministic case, we realized a high explanatory value - measured by adj. R^2 - with only these six aggregate job set characteristics. This means that we identified the most important job set characteristics that influence the interaction margin. By means of multiple regression analysis, we developed both simple models, which only include the main effects, and more complex models, which include main effects and two-way interactions between these characteristics. Among the models we developed to predict the average interaction margin, the simplest model appears to be the "best"; i.e., it has the highest adj. R^2 value (0.77), and the lowest $\widehat{\sigma}$ on both the construction data set (0.107) and testing data set (0.111). In addition, the simple model has the advantage of being easier to understand.

We used the estimated interaction margin - in addition to the workload - to predict the makespan of the job set. From a production point of view, the makespan of a job set is more interesting than the interaction margin, because the estimated makespan indicates whether a job set can be completed within a given planning period. Investigations of both the construction data set and the testing data set showed that accurate predictions for the makespan may be obtained. To support order acceptance decisions, it is necessary to have an accurate model of what workload and job mix can be realized by a production department in a given planning period. The use of the estimation models developed in this chapter, in a situation with random order arrivals, is the subject of the following chapter.

Chapter 4

Dynamic order acceptance

4.1 Introduction

In this chapter[1] we develop models to support order acceptance decisions. In Chapter 2, we defined order acceptance as the decision function that accepts or rejects orders based on the availability of sufficient capacity to complete the orders before their requested due date. Different policies may be used to evaluate whether sufficient capacity is available to produce an order before the due date requested by the customer. Most of the order acceptance literature (see Section 4.2) focuses on the use of either aggregate information or detailed information. This resembles the policies used in industry, which are generally workload based if the capacity complexity is low or sufficient slack exists in the system, or detailed scheduling based if less slack exists in the system or increased complexity (Raaymakers, 1999). Batch chemical plants can be considered as complex job shops with additional constraints. Due to the high scheduling complexity and high interrelations between jobs in multipurpose batch chemical shops, schedule based evaluations are very time consuming. On the other hand, the main problem resulting from neglecting detailed scheduling information is that many of the accepted orders cannot be completed in the planned period. This results in low delivery reliability and many replanning activities.

Previous research (Raaymakers, 1999) showed that a regression model that uses a small number of job set characteristics to estimate the makespan of the job set may be successfully used to support customer order acceptance decisions. However, she developed and tested the model under deterministic production conditions. Recognizing that in a real life production situation there is no such thing as deterministic processing times, we developed a regression model in the previous chapter that accounts for the processing times uncertainty. In this chapter, we examine the appropriateness of using such a regression model to support customer order acceptance decisions in the presence of production uncertainty.

[1] Earlier versions of the content of this chapter and Chapter 3 are joint work with J.C. Fransoo and J.M.W. Bertrand, and have been published in Ivanescu *et al.* (2002) and Ivanescu *et al.* (2003a).

The main advantage of an order acceptance policy based on aggregate information is that it is quick. Most customers may receive a quick confirmation of the order. Furthermore, a regression model is easy to apply. Thus, it makes sense to investigate the performance that can be obtained by using such a model for aggregate planning purposes as compared to detailed scheduling-based models, in a hierarchical production situation as outlined in Chapter 2. For this purpose, we conduct extensive simulation experiments to compare a detailed scheduling-based acceptance procedure, referred as the *scheduling policy*, and an aggregate acceptance procedure, the *regression policy*.

The remainder of this chapter is organized as follows. We provide an overview of the order acceptance literature in Section 4.2. In Section 4.3 we introduce the order acceptance policies: the regression policy in Section 4.3.1 and the scheduling policy in Section 4.3.2. In Section 4.4 we perform simulation experiments to evaluate the performance of these two order acceptance policies. In Section 4.5 we give our conclusions.

4.2 Review of literature on order acceptance

Order acceptance has received limited attention in the literature. Research on order acceptance is reported by Guerrero & Kern (1988); Kern & Guerrero (1990); Wester *et al.* (1994); Ten Kate (1994); Wang *et al.* (1994); Akkan (1997); Raaymakers *et al.* (2000a) and Raaymakers *et al.* (2000b).

In the literature, order acceptance decisions are often based on the workload that has already been accepted, compared to the available capacity. For example, Guerrero & Kern (1988) and Kern & Guerrero (1990) consider an assemble-to-order situation that is controlled by Material Requirements Planning(MRP)-II. The available capacity in a planning period is either allocated to customer orders, reserved for assembling available-to-promise products, or non-allocated. Guerrero & Kern (1988) distinguish between front loading and back loading; for both policies the workload allocated to each period should not exceed the available capacity. In front loading the earliest available capacity is allocated to the orders, whereas in back loading the order is produced as late as possible while not violating its due date. In a more recent paper, Kern & Guerrero (1990) present a mathematical model in which orders are accepted and jobs are allocated in such a way that the total workload per planning period does not exceed the available capacity. Furthermore, their model considers the availability of the materials required for assembly. Wang *et al.* (1994) also accept orders based on the workload already accepted. They consider an over-demanded job shop with non-negotiable due-date requirements. In their policy, customer orders are collected and prioritized. The acceptance decision is made for each customer order following the priority list. Orders may be accepted as long as the accepted workload for both machines and operators does not exceed the available capacity in a given period.

Another policy commonly found in the literature is order acceptance based on detailed scheduling. For example, Akkan (1997) considers a single resource production system for which orders are accepted only if they can be included in the schedule such that they can be completed before their due date, without changing the schedule for the

orders already accepted. So rescheduling is not allowed; the authors propose several heuristics that create good schedules without any rescheduling. Some of these heuristics aim at decreasing the "fragmentation of the time-line", which means that there exist many (small) gaps between the processing steps scheduled. If the time-line is less fragmented, it is easier to include orders with long processing times. The performance measure used is total present value of lost contribution due to rejected orders and cost of earliness. Simulations show that basic backward and forward scheduling heuristics perform significantly worse than heuristics that aim at decreasing fragmentation of the time-line.

Wester *et al.* (1994) consider a single resource production system with setup-times and orders for a limited number of product types. They compare three customer order acceptance policies: the monolithic policy, the hierarchic policy, and the myopic policy. The monolithic policy accepts orders based on a detailed schedule, which is constructed each time an order arrives. An order is rejected if acceptance of the order would lead to violating the requested due date for any order already accepted. The hierarchic policy accepts orders based on the total workload of all accepted orders. Orders may be accepted as long as the total workload does not exceed a maximum workload, which is determined by trial and error. The value for the maximum workload is selected such that no lateness occurs, in order to make a fair comparison with the monolithic policy. The myopic policy accepts orders using the same criterion as the hierarchic policy. The difference between the policies is that in the hierarchic policy a detailed schedule is constructed for selecting the order to be executed next, whereas in the myopic policy dispatching rules are used to select the next order. Simulation results show that the hierarchic and myopic policies perform worse than the monolithic policy if the setup times are sufficiently large and the due dates sufficiently tight. In cases with loose due dates, the monolithic policy appears to perform slightly worse than the other two policies. The results further show hardly any difference between the performances of the hierarchic and myopic policy.

Ten Kate (1994) also compares different order acceptance policies for single resource production systems with setup times. Processing times are assumed fixed and equal for each order of the same product type. Furthermore, due dates are set with fixed lead times. Two policies are considered: the hierarchical and the integrated policy. In the hierarchical policy, the order acceptance decision is based on the total work load of the orders already accepted. In the integrated policy, order acceptance and detailed scheduling are integrated. The order acceptance decision is based on a schedule, that is constructed each time an order arrives. Simulation experiments show that the performances of both policies shows little difference with respect to four performance measures: average cost, average lateness, fraction of tardy orders, and average batch size. The integrated policy performs better only in situations with extremely high utilization levels and tight due dates.

Raaymakers *et al.* (2000a) study the performance of workload rules for order acceptance in batch chemical manufacturing. They consider the total workload and the workload per work center. It turns out that these methods do not perform well in their specific setting. Raaymakers *et al.* (2000b) compare a regression based makespan estimation policy with both a workload-based policy and a detailed scheduling-based

policy for batch chemical manufacturing in a setting with deterministic processing times. When the utilization is high and there is a high variety in the job mix, the regression-based model outperforms the workload-based model.

A body of literature closely related to the literature on customer order acceptance in production systems whith demand exceeding the available capacity is the job selection literature (see e.g. Duenyas & Hopp, 1995; Duenyas, 1995; Slotnick & Morton, 1996; Ghosh, 1997; Lewis & Slotnick, 2002). The job selection problem addressed in Slotnick & Morton (1996) and Ghosh (1997) is how to maximize total net profit (total profit minus lateness costs), for a given set of customer orders - with different due dates, processing times, and revenues - that have to be processed on a single resource with a given capacity. Lewis & Slotnick (2002) extend the one-period deterministic model studied in Slotnick & Morton (1996) to a multi-period setting. Duenyas & Hopp (1995) and Duenyas (1995) develop queueing models that allow customers to leave if the due-date offered by the firm is too late. The objective is to maximize profit; the decisions concern sequencing and due-date setting.

Balakrishnan *et al.* (1996) develop a single period heuristic model called "capacity rationing" for allocating capacity between two product classes - one class yielding a higher profit contribution per unit of capacity allocated to it than the other class - to maximize overall profit. Using a decision-theory based approach, they demonstrate that a make-to-order firm encountering expected total demand in excess of installed capacity can increase its profit substantially (compared to a model that implements no capacity rationing) by selectively rejecting orders for the lower-profit product class. Barut & Sridharan (2004) extend this model to a multi-product, multi-period setting.

There are four important differences between the production situations considered in the literature and the situation considered in this thesis. First, the production situations in the studies mentioned above are relatively simple, with respect to the number of resources and the number of different products. The production situation in multipurpose batch process industries is considerably more complex. We consider a production system that contains a number of resource types and one or more identical resources per type. Each order may concern a different product so the processing structure for each job that needs to be executed may be different. However, we do not consider set-up times in our situation. Second, deterministic processing times are assumed in those studies. As mentioned in Chapter 1, in most real life situations, the processing times are uncertain. In this thesis we study settings with Erlang-distributed processing times. Third, we consider the use of discrete planning periods, whereas in the literature continuous time is considered. In our situation, orders are accepted and the resulting jobs are allocated to these periods. Fourth, in contrast to the job selection or the capacity rationing problem, we assume that all arriving customer orders generate the same revenue per unit of capacity consumed.

4.3 Order acceptance policies

In this section, we develop methods to assist the planner in his or her decision on accepting or rejecting customer orders such that a high resource utilization is reached

by the production department and a high service level is realized for the customers. Specifically, upon an order arrival, these methods estimate the completion time (i.e. makespan) of the job set that results by adding this order to the orders already accepted. The new order is accepted only if sufficient capacity is expected to be available to complete the resulting job set such that a pre-specified delivery reliability is achieved. Orders that fail this test are rejected and leave the system.

Given that we consider a stochastic environment, the estimate of the completion time has to account for its stochastic nature. One possibility would be to consider the expected value of the makespan. However, assuming a symmetric makespan distribution, using the expected value as the criterion for order acceptance implies a job set service level (defined as the probability of on-time job set completion) of only 0.5, which is rarely acceptable. We propose two order acceptance policies that aid decision makers to determine job sets that maximize the resource utilization under a job set service level constraint. The two policies require different levels of detail of information. The regression policy uses aggregate information - namely the regression model we developed in the previous chapter - to estimate the makespan of the job set. The scheduling policy uses detailed information - namely a simulated annealing algorithm and an empirically determined slack - to estimate the makespan of the job set. The remainder of this section gives a detailed and formal description of these two policies.

4.3.1 Regression policy

In the previous chapter we developed a multiple linear regression model to predict the interaction margin - the response variable defined in (3.1). Under the regression policy, the regression model is used dynamically, i.e. it is used each time an order arrives to investigate the consequences of accepting this order in addition to the orders already accepted. In other words, we use a lower bound on the makespan ($LB(J)$) and the estimated interaction margin ($\widehat{I(J)}$) to determine an estimate for the makespan of a job set (see equation (3.2)). Then, orders are accepted if

$$\widehat{C}_{\max}(J) \leq T. \tag{4.1}$$

The regression equation (3.1) estimates the mean interaction margin of a given job set J. Assuming normally distributed errors, and using this estimated interaction margin to estimate the makespan of a job set, the probability that the realized (ex post) makespan exceeds the estimated makespan given by (3.2) is only 0.5. Consequently, if a job set is obtained with an estimated makespan that equals the period length, then 50% of these job sets would not be achievable, implying a service level of 0.5. In this thesis, however, we aim at a job set service level target equal to $1-\alpha \gg 0.5$. Therefore, instead of the mean interaction margin, we consider the $100(1-\alpha)\%$ upper prediction bound for the interaction margin, denoted by $U^{1-\alpha}$. This gives the following formula for the makespan estimate:

$$\widehat{C}^{1-\alpha}_{\max}(J) = (1 + U^{1-\alpha})LB(J) \tag{4.2}$$

The way a one-sided prediction bound for multiple linear regression is determined can be found in most statistics textbooks (see e.g. Hahn & Meeker, 1991; Montgomery & Peck, 1992). We give here the formula for the upper $100(1-\alpha)\%$ prediction bound, and we refer to these textbooks for more details. Furthermore, note that this bound provided by the estimated regression equation developed in the previous chapter is in a log-transformed scale. Therefore, when determining the bound for the interaction margin, it is necessary to convert this bound back to the original units. Hence, the $100(1-\alpha)\%$ upper prediction bound for the interaction margin is given by:

$$U^{1-\alpha} = \exp\left(\widehat{Y} + t_{\alpha,n-(d+1)} \cdot \widehat{\sigma} \cdot \sqrt{1 + \mathbf{x}_*^T (X^T X)^{-1} \mathbf{x}_*}\right) \tag{4.3}$$

where

$\widehat{Y}$ = the estimated mean value of the transformed interaction margin;
$t_{\alpha,df}$ = the required critical value of Student's t distribution with df degrees of freedom;
$\widehat{\sigma}$ = the standard error of regression;
X = the $n \times (d+1)$ matrix of the values of the regressor variables;
$\mathbf{x}_*$ = the row vector identifying the coordinates at which the prediction is to be made;
n = the number of observations;
d = the number of regressors variables.

Note that if the sample size is large, the term $\mathbf{x}_*^T (X^T X)^{-1} \mathbf{x}_*$ is negligible (so the square root factor in (4.3) can be ignored).

In summary, under the regression policy, orders are accepted as long as

$$(1 + U^{1-\alpha}) \cdot LB(J) \leq T. \tag{4.4}$$

4.3.2 Scheduling policy

Under the scheduling policy, the planner investigates the consequences of accepting an arriving order by constructing a detailed schedule for the resulting job set (i.e., the job set that contains all the orders already accepted and the order that just arrived). In a deterministic situation, the ex ante makespan of a constructed detailed schedule is identical to the actually realized (ex post) makespan, and is therefore the best estimate possible. This is not the case in a stochastic situation. Portougal & Trietsch (2001) demonstrated by simulation that, when jobs have random processing times, using only the ex ante makespan for planning purposes may cause low service levels because it neglects part of the effect of processing time variation on the actual makespan: the ex ante makespan is only a lower bound for the actual (ex post) makespan that can be realized by the production system. Therefore, the planner should also account for the processing times variation when deciding on accepting an order, if appropriate delivery performance is desired. This may be done by including extra slack time in the ex ante schedule to compensate for the stochastic nature of the environment. Given that we are concerned with the makespan of a scheduled

set of jobs and not with the individual job completion times, we use a slack time proportional to the deterministic ex ante makespan.

Formally, an order arriving in period t is accepted for execution in period $t+1$ as long as an ex ante schedule of the resulting job set J can be constructed and the following holds:

$$(1 + \gamma_k^{1-\alpha}) \cdot C_{\max}^{ex\ ante}(S_J) \leq T \tag{4.5}$$

where $\gamma_k^{1-\alpha} \cdot C_{\max}^{ex\ ante}(S_J)$ is the slack time needed to compensate for the stochastic processing times; $\gamma_k^{1-\alpha}$ denotes the slack factor which is determined under the constraint that a minimum target service level equal to $1 - \alpha$ should be reached.

Slack factor estimation Upon each order's arrival, given the resulting job set and its corresponding deterministic ex ante makespan, the planner has to be able to estimate how much slack needs to be added to compensate for the stochastic processing times. The planner may either ask the experts for their (subjective) estimate for the value of the slack factor $\gamma^{1-\alpha}$ or he may use real (historical) or simulated data to determine this estimate. As said before, the research presented in this thesis is simulation based; the same construction data set used to determine the coefficients of the regression model for the regression policy is now used to estimate the slack factor.

For a given job set J in the construction data set and its corresponding deterministic ex ante schedule S_J, the need for slack is measured by the makespan increase, denoted by $\delta(S_J)$. The makespan increase is defined as the relative difference between the ex post makespan and the ex ante makespan of the job set:

$$\delta(S_J) = \frac{C_{\max}(S_J) - C_{\max}^{ex\ ante}(S_J)}{C_{\max}^{ex\ ante}(S_J)} \tag{4.6}$$

Since the actual makespan $C_{\max}(S_J)$ is a random variable determined by the schedule S_J and by individual durations P_{ij}, $\delta(S_J)$ is also a random variable. The slack factor is directly determined from the empirical distribution of $\delta(S_J)$. The size of the slack factor is determined by the target level of the delivery performance.

Given that the Erlang shape parameter k is assumed the same for all the processing times in the job set, we can obtain, from the construction data set, an empirical distribution of $\delta(S_J)$ for each different value of k. Furthermore, we determine an estimate for the $100(1-\alpha)$th quantile of the distribution of $\delta_k(S_J)$, for each Erlang shape parameter k. We use the direct-simulation quantile estimator implemented in the SPSS statistical package to compute this estimate, denoted by $\widehat{\delta_k^{1-\alpha}}(S_J)$:

$$\widehat{\delta_k^{1-\alpha}}(S_J) = \delta_{k;(\lceil n_{repl} \cdot (1-\alpha) \rceil)}(S_J) \tag{4.7}$$

where $\lceil x \rceil$ denotes the smallest integer that is greater than or equal to x and $\delta_{k;(1)}(S_J) \leq \delta_{k;(2)}(S_J) \leq ... \leq \delta_{k;(n_{repl})}(S_J)$ are the order statistics obtained by sorting the observations $\{\delta_{k;i}(S_J) : i = 1, ..., n_{repl}\}$ in ascending order.

Then, for each Erlang shape parameter k, the estimated slack factor is given by

$$\gamma_k^{1-\alpha} = \frac{\sum_{J=1}^{n_{\mathrm{JS;\,k}}^{\mathrm{cons}}} \widehat{\delta_k^{1-\alpha}}(S_J)}{n_{\mathrm{JS;\,k}}^{\mathrm{cons}}} \tag{4.8}$$

where $n_{\mathrm{JS;\,k}}^{\mathrm{cons}}$ denotes the number of job sets in the construction data set that have been allocated to the Erlang shape parameter k. For $\alpha = 0.05$ we found that the estimated slack factor has the values $0.61, 0.46, 0.38, 0.33, 0.30, 0.27, 0.25, 0.23, 0.21$ where each value corresponds respectively to an Erlang shape parameter k taking values from 2 to 10.

4.4 Performance comparison by simulation

In this section we conduct simulation experiments to compare the performance of the two order acceptance policies in a setting with random order arrivals and processing times. This section is organized as follows. Subsection 4.4.1 discusses the design of the experiments. Subsection 4.4.2 introduces the performance measures. Subsection 4.4.3 presents the experimental results.

4.4.1 Experimental design

In this subsection, we discuss both the general settings of the simulation experiments and the parameters that are varied. The following assumptions are made with respect to the simulation experiments:

- the shop consists of 10 resources of 5 types, and
- exogenously determined due date for all orders.

The simulated shop is composed of 5 resource types, each containing 2 identical resources. This resource configuration has also been used to develop the interaction margin estimation models in the previous chapter.

We assume that customer orders arrive with exponentially distributed inter-arrival times. The order arrival process is not influenced by the outcome of an acceptance decision. The arrival rate is determined by the ratio between required and available capacity. All orders arriving in a planning period have a given due date, which is equal to the end of the next planning period.

The performance of the two policies may be affected by the demand/capacity ratio, job mix variety, workload balance, and the uncertainty level in the processing times. A previous study (Raaymakers, 1999) comparing regression and scheduling policies for order acceptance showed that the first three experimental factors significantly affected the performance of these policies. This study, however, considered deterministic production situations. It is reasonable to expected that these factors will affect

Table 4.1: Levels of the experimental factors

	L	H
Demand/capacity ratio	0.7	1.0
Job mix variety	$s_j \sim U(4,7)$ $F_{E[p]} = U(15,35)$	$s_j \sim U(1,10)$ $F_{E[p]} = U(1,49)$
Workload balance	$p_m \in \{0.3, 0.25, 0.20, 0.15, 0.10\}$	$p_m = 0.20$, $m = 1, ..., 5$
Uncertainty level	$P_{ij} \sim$ Erlang-10	$P_{ij} \sim$ Erlang-2

the performance of the order acceptance policies under stochastic production situations. We consider two levels for each of these experimental factors. We chose the two levels in such a way that the difference between them allows us to properly assess the influence of each of these factors on the performances of the order acceptance policies. At the high level, the average demand requirements for capacity are equal to the total available capacity per planning period. At the low level, the average demand requirements for capacity are equal to 70% of the total available capacity per planning period. As has been shown by Raaymakers *et al.* (2000a), the no-wait structure of processing steps in each job implies capacity utilization levels between 50% and 60%. Thus, both demand levels investigated represent situations where demand effectively exceeds available capacity.

Each order consists of exactly one job with a specified structure of no-wait processing steps. The job characteristics have been generated randomly upon arrival of the order. Hence, each job arriving at the system may be different. The parameters that determine the job mix variety are the number of processing steps per job and the distribution that gives the expected processing times. In order to carry out a realistic simulation study, the values of the parameters have an order of magnitude similar to the one observed in reality. Moreover, the number of processing steps per job cannot exceed the number of resources, because each processing step of a job has to be performed on a different resource if the processing steps are overlapping. Nevertheless, jobs in a job set may differ in the number of processing steps. In the situation with high job mix variety (i.e. less homogeneous) the number of processing steps per job is uniformly distributed between 1 and 10 and the expected processing time is uniformly distributed between 1 and 49. In the situation with low job mix variety, i.e. more homogeneous, the number of processing steps per job is uniformly distributed between 4 and 7, and the expected processing time is uniformly distributed between 15 and 35. Note that in both situations the average number of processing steps and the average expected processing time is the same. We mentioned above that the arrival rate of the incoming stream of orders is determined by the ratio between the average demand requirements for capacity and the available capacity. Thus,

$$\lambda = \text{demand/capacity ratio} \cdot \frac{N}{E[s_j] \cdot \mu_{E[p]}} \tag{4.9}$$

where N is the total number of resources, $E[s_j]$ denotes the average number of processing steps and $\mu_{E[p]}$ denotes the average expected processing times.

When generating the jobs, each processing step is allocated at random to a resource type. In situations with high workload balance, the allocation probability is the same for each resource type. In situations with low workload balance, the allocation probability is different for each resource type. On average 30, 25, 20, 15 and 10% of the processing steps were allocated to the five different resource types respectively.

Given that we consider a setting with stochastic processing times, the level of uncertainty in the processing times may affect the performance of the policies. We consider two levels of uncertainty. At the low uncertainty level, the shape parameter k is equal to 10; at the high uncertainty level, the shape parameter is equal to 2.

Note that for these experiments the jobs are generated with similar characteristics as the jobs in the construction data set (see Table 3.1 for the values of s_j, p_m, and $F_{E[p]}$). Given that one of the policies uses a regression model, this is a necessary requirement: when making predictions, the underlying regression assumption is that the data for constructing the model and the future cases to be predicted are a sample from the same population. However, in order to limit our computational effort, we now combine the s_j and $F_{E[p]}$ factors of Table 3.1 into one factor, namely the job mix variety. Also, we consider here only two values for the Erlang shape parameter k instead of the whole range (from 2 to 10) that was used for the construction data set. Finally, a job set service level target value of 95% was chosen, i.e. $\alpha = 0.05$.

Table 4.2: Scenarios: combinations of four experimental factors

Scenario	*demand/capacity ratio*	*job mix variety*	*workload balance*	*uncertainty level*
1	*H*	*H*	*H*	*H*
2	*H*	*H*	*H*	*L*
3	*H*	*H*	*L*	*H*
4	*H*	*H*	*L*	*L*
5	*H*	*L*	*H*	*H*
6	*H*	*L*	*H*	*L*
7	*H*	*L*	*L*	*H*
8	*H*	*L*	*L*	*L*
9	*L*	*H*	*H*	*H*
10	*L*	*H*	*H*	*L*
11	*L*	*H*	*L*	*H*
12	*L*	*H*	*L*	*L*
13	*L*	*L*	*H*	*H*
14	*L*	*L*	*H*	*L*
15	*L*	*L*	*L*	*H*
16	*L*	*L*	*L*	*L*

We use a four-factor full factorial design. Table 4.2 gives the combinations of the experimental factors. The $2^4 = 16$ combinations will be referred to as scenarios in the remainder of this thesis. In each scenario, common random numbers are used to generate identical order arrivals for the two policies. Also, for different scenarios we use common random numbers as a variance reduction technique. In total, we performed 32 (=16 scenarios $\times$2 policies) simulation runs.

We use a simulation run-length of 15 independent planning periods. A planning period always starts with an empty system; at the end of the planning period, the system must be empty again. The length of a planning period is chosen such that the job set consists of a realistic number of jobs. The empirical study of Raaymakers *et al.* (2000a) showed that a job set of 40 to 50 jobs is realistic for this type of industrial process. To realize such job sets, we fix the length of the planning period at 1300 time units (as additional experiments indicate). The absolute length of the planning period, however, depends on the average processing time per job which is chosen arbitrarily. Each simulation run yields 15 independent job sets (see Section 2.2.3). The execution of each job set is repeated 250 times; i.e. we perform 250 ($= n_{repl}$) independent replications.

4.4.2 Performance measures

Order acceptance policies should accept orders such that a pre-specified delivery reliability is achieved, while maximizing resource utilization. Thus, the following three measures are considered: the percentage job sets on time (POT), the mean job set tardiness (JST), and the mean realized capacity utilization (RCU). A job set J is on time if the actual (realized) job set completion time (i.e. $C_{\max}(S_J)$) does not exceed the period length T.

$$POT = \frac{\sum_{r=1}^{n_{repl}} \chi_r(J)}{n_{repl}} \tag{4.10}$$

where

$$\chi_r(J) = \begin{cases} 1 & \text{, if } C_{\max,r}(S_J) \leq T \\ 0 & \text{, otherwise.} \end{cases} \tag{4.11}$$

and $C_{\max,r}(S_J)$ is the realized makespan corresponding to the r-th replication, $r = 1, ..., n_{repl}$.

Job set tardiness occurs when the actual job set completion time is greater than the due-date (T):

$$JST_r = (C_{\max,r}(S_J) - T)^+ \tag{4.12}$$

The realized capacity utilization per period is measured as follows:

$$RCU_r = \frac{\sum_{j=1}^{n_J} \sum_{i=1}^{s_j} \theta_{p_{ij}}}{N \cdot T} \tag{4.13}$$

where n_J denotes the number of jobs in the job set J, N denotes the number of resources, and $\theta_{p_{ij}}$ is defined as

$$\theta_{p_{ij}} = \begin{cases} p_{ij} & \text{, if } st_{ij} \leq T \text{ and } c_{ij} \leq T \\ p_{ij} - (c_{ij} - T) & \text{, if } st_{ij} \leq T \text{ and } c_{ij} > T \\ 0 & \text{, if } st_{ij} > T \end{cases} \tag{4.14}$$

where st_{ij} is the start time and c_{ij} is the completion time of i-th processing step of job j.

Given the definitions in equations (4.12) and (4.13), the mean job set tardiness and the mean realized capacity utilization are as follows:

$$JST = \frac{\sum_{r=1}^{n_{repl}} JST_r}{n_{repl}} \tag{4.15}$$

$$RCU = \frac{\sum_{r=1}^{n_{repl}} RCU_r}{n_{repl}} \tag{4.16}$$

Besides these three measures we also consider another internal measure, namely the feasibility performance (FEP). When using the right-shift procedure to restore the feasibility on the resources, unavoidable violations of the no-wait restrictions may occur when the job sets are executed. This may cause product quality problems, so a small number of no-wait restrictions violations is preferred. We define the feasibility performance as 1 − the fraction of processing steps that violate the no-wait restrictions, from the start to the completion of the job set.

4.4.3 Experimental results

In this section we discuss the results of our simulation experiments. The criterion for evaluating the order acceptance procedures was the ability to effectively meet the customer requirements via the percentage of job sets on time (POT) and the mean job set tardiness (JST). Two internal measures, namely the mean realized capacity utilization (RCU) and the feasibility performance (FEP) were also measured. In Table 4.3 and Figure 4.1, we present the average of the 15 independent planning periods.

Examining Table 4.3, we observe that the difference in performance between the two policies is not the same for all scenarios. For example, for scenarios with high job mix variety (1 through 4 and 9 through 12), the differences in performance are considerably higher than for scenarios with low job mix variety. Also, we observe that the difference in performance is smaller if a clear bottleneck resource type exists (i.e. low workload balance scenarios: $3, 4, 7, 8, 11, 12, 15$, and 16).

To confirm these observations formally, we investigate the main and interaction effects of the four experimental factors (i.e. demand/capacity ratio (A), the job mix variety (B), the workload balance (C), and the uncertainty level (D)) on the difference in the primary performance measures (POT and RCU) of the two policies. Testing hypotheses concerning the effects of various levels of a factor and detection of interactions between factors is generally done by using analysis of variance (ANOVA). Given that we used common random numbers across scenarios, the independence assumption of ANOVA does not hold (see Maxwell & Delaney (1990), page 110). However, the factorial design enables us to use the following procedure to detect significant effects (Kleijnen, 2004). We transform each experimental factor to fall onto -1 and

Table 4.3: Simulation results: *POT*, *JST*, and *RCU* measures for both the regression policy and the scheduling policy

Scenario	*Regression policy*			*Scheduling policy*		
	POT	*JST*	*RCU*	*POT*	*JST*	*RCU*
1	81.95	10.82	0.36	74.85	16.50	0.40
2	97.41	0.52	0.47	69.33	8.73	0.53
3	82.96	10.84	0.35	80.35	11.97	0.37
4	87.52	3.34	0.44	74.99	6.03	0.50
5	91.60	3.16	0.35	93.71	2.71	0.37
6	99.79	0.03	0.44	90.48	1.61	0.50
7	91.89	3.43	0.33	96.00	1.49	0.33
8	99.55	0.06	0.41	94.08	0.98	0.46
9	89.41	6.15	0.35	84.13	9.29	0.37
10	92.21	2.54	0.45	86.16	3.17	0.50
11	87.20	6.83	0.34	84.27	9.40	0.36
12	94.59	1.19	0.43	87.20	2.86	0.47
13	93.60	2.46	0.35	92.43	2.89	0.37
14	99.89	0.02	0.44	94.29	0.87	0.49
15	87.36	6.26	0.33	96.85	1.13	0.33
16	98.00	0.36	0.41	96.56	0.56	0.44

1 at their low and high values (see Law & Kelton (2000)). Following Law & Kelton (2000), page 649, we use regression analysis to obtain a model that estimates the main and two-way interaction effects for each planning period. Note that we use regression only as an alternative way to compute the effects' estimates using the SPSS statistical package. Since we simulated 15 independent planning periods, we get 15 i.i.d. least-squares estimators for each of the 10 effects (main and two-way interaction). We further use a two-sided one-sample t-test to investigate whether the average of these 15 values is significantly different from zero. Table 6.7 presents the results of our 20 (=2 measures $\times$10 effects) one-sample t-tests.

Table 4.4 shows that all main effects are significant at 95% confidence level. Also, some of the two-way interaction effects are significant; the demand/capacity ratio-uncertainty level interaction (AD) is significant for both measures, whereas the job mix variety-uncertainty level interaction (BD) significantly affects only the *RCU* measure. We further see that the uncertainty level (factor D) is the most important factor for both performance measures.

We next compare the performance of the two policies within each scenario. Because common random numbers were used to generate identical order arrivals for the different policies, the paired t-test was selected as the most appropriate statistical tool for determining the significance of performance variations. This technique tests the significance of the differences between average responses of a performance measure obtained by the scheduling policy versus the regression policy. Table 4.5 presents the results of the 48(= 16 scenarios $*$ 3 performance measures) paired t-tests we performed.

Table 4.4: One-sample t-test results

Main and two-way interactions	*POT*		*RCU*	
	t statistic	*P-value*	*t statistic*	*P-value*
A	2.397	0.031	-3.682	0.002
B	4.693	0.000	-3.627	0.003
C	3.900	0.002	-5.484	0.000
D	-11.735	0.000	17.727	0.000
AB	1.167	0.263	0.427	0.676
AC	0.967	0.350	0.573	0.576
AD	-3.280	0.005	3.944	0.001
BC	0.042	0.967	1.792	0.095
BD	0.042	0.967	-3.261	0.006
CD	-0.266	0.794	-1.731	0.105

A: demand/capacity ratio, B: job mix variety, C: workload balance, D: uncertainty level

Table 4.5 shows that in all but six scenarios (namely scenarios $3, 5, 10, 11, 13$ and 16) there are significant differences (at 95% confidence level) between the two policies with respect to the POT and JST measures. This indicates that only in 4 of the 8 scenarios corresponding to the case of high processing times uncertainty, we find significant differences. Among the high uncertainty level scenarios (the odd numbered scenarios) where significant differences are found, the regression policy outperforms the scheduling policy in the case of high job mix variety and high workload balance (scenarios 1 and 9). We discuss now the low uncertainty level scenarios (the even number scenarios). We find significant differences between the two policies in all these scenarios but two, namely scenarios 10 and 16. Examining the results in Table 4.3 we observe that the regression policy outperforms the scheduling policy with respect to POT and JST in the case of high job mix variety (scenarios $2, 4$, and 12). However, in the case of low job mix variety (scenarios $6, 8$, and 14), although the regression policy reaches POT values exceeding the target, the scheduling policy reaches performance values closer to the target. We further observe that both policies have a relatively poor control of the delivery performance. The results suggest that in terms of delivery reliability (POT and JST), the regression policy should be preferred if uncertainty level is high, job mix variety is high and the workload is evenly balanced (i.e., there is not a clear bottleneck). Under the remaining settings, the results are not conclusive.

With respect to the RCU performance measure, the t-test shows that in all scenarios but two (namely scenarios 7 and 15) there are significant differences between the two policies. Referring back to Table 4.3, for scenarios where we found significant differences, we observe that the scheduling policy reaches a higher realized capacity utilization than the regression policy. Note that the RCU is considerably higher if the uncertainty level is low; this holds for both policies. This may be explained as follows: in a situation with high uncertainty in the processing times more slack is needed to be added to cope with this uncertainty, resulting in a lower number of accepted jobs and therefore lower RCU values.

Table 4.3 shows that, if both the demand/capacity ratio and the job mix variety

Table 4.5: Comparing scheduling and regression policies: paired t-test results

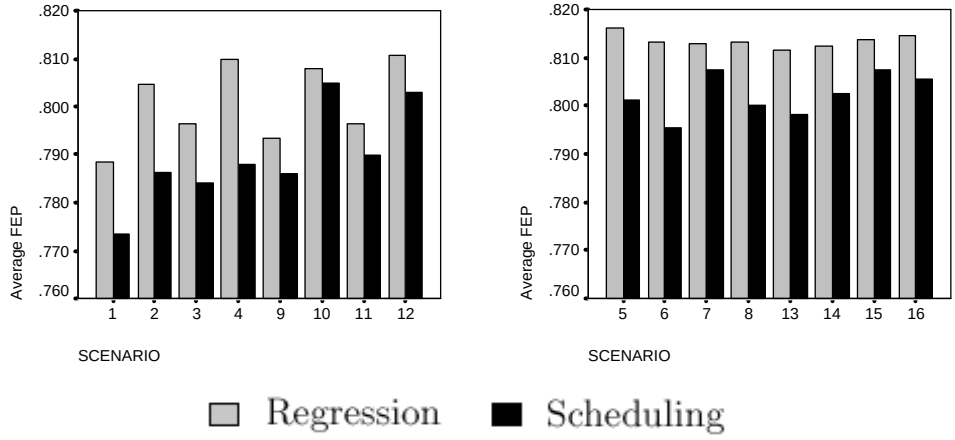
Scenario	*POT*		*JST*		*RCU*	
	t-value	P-value	t-value	P-value	t-value	P-value
1	-2.496	0.000	2.546	0.023	7.718	0.000
2	-7.786	0.000	6.747	0.000	12.791	0.000
3	-0.848	0.411	0.531	0.603	3.238	0.006
4	-2.871	0.012	2.024	0.062	9.888	0.000
5	1.179	0.258	-0.612	0.551	4.080	0.001
6	-8.263	0.000	5.930	0.000	13.846	0.000
7	2.808	0.014	-2.249	0.041	1.809	0.092
8	-7.543	0.000	6.839	0.000	12.878	0.000
9	-3.224	0.006	2.657	0.019	6.979	0.000
10	-1.051	0.311	0.290	0.776	8.381	0.000
11	-1.170	0.262	1.617	0.128	7.860	0.000
12	-2.394	0.031	2.031	0.062	8.450	0.000
13	-0.983	0.342	0.763	0.458	8.817	0.000
14	-4.976	0.000	4.308	0.001	14.992	0.000
15	4.286	0.001	-3.752	0.002	-1.459	0.167
16	-0.936	0.365	0.671	0.513	8.096	0.000

are high (scenarios 1 through 4), the scheduling policy reaches the highest capacity utilization and the poorest performance with respect to *POT*. We conjecture that this is due to the way this policy accepts the orders: by making use of detailed information when accepting the orders, the scheduling policy selects only the jobs that fit in well. The result of this selectivity is a tight schedule, which further results in high *RCU* values. However, a too tight schedule seems to be detrimental to the ability of effectively meeting the order set delivery performance target. This selectivity issue is addressed in the next chapter.

The last performance measure we discuss is the feasibility performance. The feasibility performance ranges between 0.76 and 0.82; see figure 4.1. We observe that the scheduling policy is clearly outperformed by the regression policy in all sixteen scenarios. This may also be the result of the way the procedures accept the orders, as discussed above. Consequently, in order to maintain the feasibility under the scheduling policy, additional slack may need to be included. Furthermore, the results show that in case of low job mix variety a higher feasibility performance is obtained. However, this holds for both order acceptance policies.

We conclude this section with a comment regarding the computational requirements for the two order acceptance policies. The average CPU time on a Pentium 1.7 GHz is about 18 seconds to evaluate a single order under the scheduling policy, whereas it takes negligible time under the regression policy.

Figure 4.1: Feasibility performance for both the regression policy and the scheduling policy



a. high job mix variety

b. low job mix variety

4.5 Conclusions

Raaymakers (1999) showed that a regression model performs reasonably well when used to accept orders in a setting with deterministic processing times. In most real life production situations, however, processing times are uncertain. Especially in process industries, fluctuations of quality of raw materials result in high degree of processing time variation.

In this chapter, we investigated the performance that can be obtained by using a regression model for customer order acceptance decision support in the presence of production uncertainty. We further compared the performance of the regression-based model to the performance of a detailed scheduling-based model. These two policies have been compared by means of simulation experiments with respect to four performance measures: the mean percentage of accepted job sets completed on time (POT), mean tardiness of the job set (JST), mean realized capacity utilization (RCU) and feasibility performance (FEP).

The detailed scheduling-based policy - called the scheduling policy - accepts orders based on a detailed schedule that has to be constructed each time an order arrives. In a stochastic situation, this schedule is only an estimate of what will be realized by the production system. In order to cope with the uncertainty in the processing times, a slack factor is introduced when the schedules are constructed upon order acceptance. Due to the great scheduling complexity and high interrelations between jobs in multipurpose batch chemical shops (Raaymakers & Hoogeveen, 2000), acceptance decisions based on detailed schedules are time consuming.

The regression-based order acceptance policy - called the regression policy - uses aggregate information to support customer order acceptance decisions. More precisely,

a regression model with a limited number of aggregate job set characteristics is used - in addition to the workload - to investigate the consequences of accepting an order given the orders already accepted. The main advantage of such an aggregate policy is that it is quick. Customers may get a fast confirmation of their order.

The results of our simulation study indicate that the scheduling policy performs better in situations with low job mix variety than the regression policy. Namely, this policy determines job sets that result in higher capacity utilization values and in a delivery performance closer to the pre-specified target. In the case of high job mix variety, however, the regression policy manages to determine job sets that result in a higher delivery performance and smaller job set tardiness than the scheduling policy. The scheduling policy, however, reaches higher capacity utilization values. We argue that this is due to the fact that the scheduling policy accepts orders based on detailed scheduling information, which allows for a better identification of orders that fit in with the already accepted orders. In other words, we conjecture that this policy accepts orders with specific characteristics in order to maximize the capacity utilization. We also expect that this selectivity, if present, is most evident for scenarios with high job mix variety and high demand/capacity ratio, since in those situations, the heterogeneity of the arriving orders gives more opportunities to be selective. We address this selectivity issues in the next chapter.

An important conclusion is that detailed scheduling information at the order acceptance level remains valuable also under stochastic production conditions. The beneficial effects of using detailed scheduling information for supporting order acceptance decisions has been previously established for deterministic production conditions (see e.g. Wester *et al.*, 1994; Ten Kate, 1994; Raaymakers *et al.*, 2000b). In a deterministic production situation, the schedule constructed upon order acceptance and used to support the acceptance decision is the exact representation of what will be realized later by the production department. In a highly stochastic environment, however, we may expect the performance of the scheduling policy to be highly affected by the uncertainty in the processing times, since the ex ante schedule constructed upon order acceptance is no longer an exact representation of the future status of the production system. This conjecture is confirmed by the results of our simulation study, only for situations with high variation in the job mix.

The overall objective of the order acceptance function, as discussed in Chapter 2, is to maximize the resource utilization while maintaining a minimum job set service level. Consequently, we designed the two order acceptance policies such that a pre-specified job set service level target is expected to be obtained. Our simulation study however shows that neither the regression policy nor the scheduling policy meets the pre-specified target for the job set service level. This poor control on the delivery performance makes the comparison of the two policies difficult, if both the realized capacity utilization and the realized job set service level criteria are considered. Obviously, given that both a high capacity utilization and a high delivery reliability are desired by the management of the production system, in a situation when both policies reach the pre-specified job set service level target, a comparison of the realized capacity utilization would determine the choice of which of the two policies to be used in a particular situation. However, in order to obtain the pre-specified job set

service level target, a large number of experimental runs have to be made to tune the parameters of both policies. This implies a substantial computational effort, especially for the scheduling policy. Recall that the CPU time needed on a Pentium 1.7 GHz is about 18 seconds to evaluate a single order acceptance decision based on the scheduling policy. With an average of about 100 arriving orders per planning period, and 15 planning periods, about 8 hours computer time is required for each scenario. Instead of performing a large number of experiments, we chose to concentrate on understanding why both policies have such a poor control on the delivery performance. This understanding can lead to constructive design of a new, improved order acceptance policy that controls the delivery performance. We address theses issues in Chapter 5 and Chapter 6 respectively.

We conclude this chapter with some additional considerations that may determine the choice between the two order acceptance policies in a particular situation, rather than their actual performance. First, if all data are available and reliable, and the computational time is not important, it is clear that the scheduling policy should be used for order acceptance under most circumstances. In case only general job characteristics are available at the time when the acceptance/rejection decision has to be made, it might be worth considering the regression policy, due to its good performance with limited information. In a multi-period setting, in which orders need to be allocated optimally to a particular planning period, it may also be better to use the regression policy, since in that case multiple schedule alternatives (under different allocations of jobs) need to be computed, leading to long response times when using the scheduling policy for order acceptance.

Chapter 5

Selectivity of order acceptance procedures

5.1 Introduction

Order acceptance policies should help decision makers to accept orders such that a pre-specified delivery reliability is achieved, while maximizing resource utilization. Furthermore, order acceptance policies have a considerable impact on the mix of jobs that need to be scheduled, by refusing specific jobs from the total demand. If orders are accepted on line from a stream of arriving customer orders, each order must be evaluated for its effect on the delivery reliability of the set of accepted orders. By selecting orders with specific characteristics that maximize resource utilization, an important and often unforeseen side-effect occurs, namely the mix of orders changes in such a way that the expected delivery reliability is no longer met. In this chapter[1] we investigate this selectivity effect.

Two order acceptance policies, which differ with respect to the level of information used, have been developed in the previous chapter. The first policy - called the *regression policy* - uses regression models to estimate the actual makespan of an order set. The second policy -called - the *scheduling policy* - uses simulated annealing techniques and an empirically determined slack to estimate the actual makespan of an order set. Orders are accepted as long as the estimated makespan does not exceed the period length, with a pre-specified probability. In the simulation study we performed in the previous chapter to compare these two policies, we assumed that the data from which the models' parameters are estimated (i.e. the regression coefficients and the slack factor) and the data that result from the acceptance procedure, are samples from the same population. This means we assumed that the orders are accepted in a non-selective way. However, the results indicate that this may not be the case. In

[1]The content of this chapter is joint work with J.C. Fransoo and J.M.W. Bertrand and has appeared in Ivanescu *et al.* (2003b).

this chapter we investigate whether the policies are indeed selective. We also study the impact of this selectivity, if proved present, on the system performance.

The remainder of this chapter is as follows. In Section 5.2 we investigate the possible selectivity of both the scheduling policy and the regression policy. In Section 5.3 we address the impact of selectivity on the performance of the two policies. We present our conclusions in Section 5.4.

5.2 Selective acceptance

To investigate whether the two policies accept orders selectively, we collect data on the following performance measures: the average workload per job set (μ_p), the average overlap per job (μ_g), the average number of processing steps per job (μ_s), and the acceptance rate (ACR). These measures are defined as follows. The average workload per job set is given by:

$$\mu_p = \frac{1}{n_J} \sum_{j=1}^{n_J} \sum_{i=1}^{s_j} E[P_{ij}] \tag{5.1}$$

where n_J denotes the number of jobs in the job set J, s_j denotes the number of processing steps of job j, and $E[P_{ij}]$ denotes the expected processing time of the processing step i of job j.

The average overlap per job and the average number of processing steps per job were defined in Chapter 3 (see equations (3.8) and (3.6)). The acceptance rate is the percentage of arrived orders that are accepted during one planning period. In the remainder of this thesis, the first three performance measures are computed for both the arriving jobs and the set of accepted jobs. We refer to these measures as the characteristics of the arriving jobs / accepted job sets.

To investigate whether jobs with specific characteristics are systematically rejected by the scheduling policy or the the regression policy, we compare the characteristics of the job sets accepted by each policy with the characteristics of the arriving jobs stream. If the policies do not accept jobs selectively, the accepted job sets and the arriving jobs will have similar characteristics.

For each of the three characteristics, we compute the difference between the values obtained for the arriving jobs - denoted by μ_p^{arr}, μ_g^{arr}, and μ_s^{arr} - and the values obtained for the accepted job sets - denoted by μ_p^{policy}, μ_g^{policy}, and μ_s^{policy} where $policy \in \{sched, regr\}$ indicates the scheduling/regression policy. In Table 5.1, we present this difference averaged over the 15 planning periods, across all sixteen simulated scenarios (see columns 2 to 8). Columns 9 and 10 in Table 5.1 give the average acceptance rate (ACR).

Table 5.1 implies that the job mix variety strongly influences the acceptance procedure. In the case of low job mix variety (scenarios 5 to 8, and 13 to 16), small differences may be observed between the characteristics of the arriving jobs and the characteristics of the job sets released to production. For each policy, each scenario,

Table 5.1: Selectivity measures for the regression and the scheduling policies

Scenario	$\mu_p^{arr} - \mu_p^{policy}$		$\mu_g^{arr} - \mu_g^{policy}$		$\mu_s^{arr} - \mu_s^{policy}$		*ACR*	
	Regr	*Sched*	*Regr*	*Sched*	*Regr*	*Sched*	*Regr*	*Sched*
1	33.90	31.64	-0.06	-0.05	1.22	1.15	0.47	0.52
2	24.99	22.51	-0.04	-0.03	0.87	0.79	0.57	0.63
3	31.93	25.30	-0.06	-0.04	1.15	0.90	0.45	0.47
4	23.41	22.94	-0.04	-0.04	0.84	0.83	0.52	0.60
5	-1.71	1.02	0.00	0.00	-0.06	0.02	0.35	0.38
6	-1.19	1.81	0.00	0.00	-0.03	0.06	0.45	0.51
7	-2.38	0.62	0.00	0.00	-0.07	0.00	0.33	0.34
8	-0.97	1.63	0.00	0.00	-0.03	0.06	0.40	0.47
9	23.47	21.47	-0.04	-0.03	0.86	0.78	0.57	0.62
10	14.66	11.47	-0.02	-0.01	0.54	0.43	0.69	0.76
11	21.91	21.49	-0.04	-0.03	0.80	0.82	0.56	0.60
12	15.41	13.41	-0.02	-0.02	0.57	0.50	0.67	0.73
13	0.00	2.12	0.00	0.00	0.00	0.06	0.51	0.55
14	1.44	1.94	0.00	0.00	0.04	0.05	0.66	0.72
15	0.34	1.95	-0.01	0.00	0.01	0.06	0.49	0.48
16	1.30	2.03	0.00	0.00	0.05	0.07	0.59	0.65

and each of the three characteristics, we use an one-sample t-test to detect if the average difference observed in Table 5.1 is significantly different from zero. An average significantly different from zero indicates that the characteristics of the arriving jobs and the characteristics of the job sets released to production are significantly different, i.e., the policies are selective. We present the results of the 96 (= 16 scenarios * 2 policies * 3 characteristics) one-sample t-tests in Appendix D (see Table D.1). For scenarios with low job mix variety, the one-sample t-test results show no significant difference (at 95% confidence level) between the characteristics of the arriving jobs and the characteristics of the jobs accepted by the regression policy. This indicates that this policy accepts orders non-selectively for these scenarios. The scheduling policy, however, is selective in three of the eight scenarios with low job mix variety, namely scenarios 13, 14, and 16. Jobs with few processing steps are preferred to jobs with many processing steps. Also the average workload per job is less than the average workload of the arrival process.

A different picture emerges for the high job mix variety scenarios (1 to 4, and 9 to 12). Large differences can be observed between the characteristics of the accepted jobs and those of the arriving jobs, for both policies. This make sense since in the case of high job mix variety, the arriving jobs are less homogeneous; so, the policies have more opportunities to be selective. The two-tailed p-value of the one-sample t-tests (see Appendix D) is equal to 0.00 for all the scenarios with high job mix variety, indicating significant differences. We conclude that both policies are highly selective in case of high job mix variety.

Furthermore, Table 5.1 shows that both policies seem to show a particular selectiveness by accepting, on average, jobs with a smaller number of processing steps, higher

overlap, and lower workload - compared to the arriving jobs.

Given this observed selectivity, we focus on the cases with high job mix variety, and we investigate whether one policy is more selective than the other. In this context, we define the concept of "degree of selectivity":

Given two policies P_1 and P_2, P_1 is more selective than P_2 if the distance between the characteristics of the arriving jobs and the characteristics of the jobs accepted by P_1 is significantly larger than the distance between the characteristics of the arriving jobs and the characteristics of the jobs accepted by P_2.

We define the distance as the square root of the sum of squared relative differences:

$$d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{policy}) = \sqrt{\left(\frac{\mu_p^{arr} - \mu_p^{policy}}{\mu_p^{arr}}\right)^2 + \left(\frac{\mu_g^{arr} - \mu_g^{policy}}{\mu_g^{arr}}\right)^2 + \left(\frac{\mu_s^{arr} - \mu_s^{policy}}{\mu_s^{arr}}\right)^2} \quad (5.2)$$

where $\boldsymbol{x}^{arr}$ ($\boldsymbol{x}^{policy}$) denotes the three-dimensional vector of the arriving jobs/job set characteristics. Table 5.2 gives this distance for the scheduling and the regression policies, for scenarios with high job mix variety.

Table 5.2: The distance for scenarios with high job mix variety

Scenario	*capacity ratio*	*workload balance*	*uncertainty level*	$d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{regr})$	$d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{sched})$
1	H	H	H	0.340	0.317
2	H	H	L	0.248	0.221
3	H	L	H	0.324	0.313
4	H	L	L	0.237	0.229
9	L	H	H	0.236	0.214
10	L	H	L	0.148	0.122
11	L	L	H	0.221	0.222
12	L	L	L	0.154	0.133

Table 5.2 shows that both policies are less selective in the case of a low arrival ratedemand/capacity ratio (scenarios 9 to 12). These results confirm our expectation that the selectivity of the policies is most clear in situations with high demand/capacity ratio and high job mix variety. Furthermore, we observe that in scenario 1, both policies show the highest distance; i.e., both policies are most selective. This is to be expected given the high heterogeneity of the arriving orders and the balanced workload.

We use paired t-tests to detect significant statistical differences between the two distances. The results presented in Appendix D (see Table D.2) show that, for all the considered scenarios but two (scenarios 2 and 10) there are no significant differences between the two distances. Thus, we may conclude that, in the case of high job mix variety, the two policies are equally selective in accepting jobs that have, on average, a smaller number of processing steps and a higher overlap than the average number

of processing steps and average overlap of the arriving jobs.

When examining the acceptance rate (columns 8 and 9 of Table 5.1), we note that in the case of high job mix variety, both policies reach a higher acceptance rate than in the case of low job mix variety. This is due to the selective way in which the policies accept jobs.

Table 5.1 shows that a higher acceptance rate is obtained in the case of low uncertainty in the processing times (even number scenarios 2, 4, 6, etc.). This makes sense, since in case of high uncertainty in the processing times a relatively large amount of slack is needed to cope with this uncertainty, and therefore a smaller number of orders is accepted.

The scheduling policy has the highest acceptance rate. Apparently, by rescheduling at every order arrival and making use of the detailed information, the scheduling policy can better identify the jobs that fit in.

5.3 Impact of selectivity on performance

In the previous section we saw that, in scenarios with high job mix variety, both policies selectively accept orders. In this section we investigate the impact of this selectivity on the performance of the two order acceptance policies. More precisely, we compare performance measures obtained in situations where no selectivity was present - i.e. the construction and the testing data sets - versus situations with random order arrivals where the selectivity proved to be present. In terms of performance measures, we consider only the percentage of on-time job sets and the realized capacity utilization measures. However, given that in the case of both the construction data set and the testing data set there are no planning periods of a fix length - both the job sets and the number of jobs in each job set were randomly generated - we cannot use these performance measures as defined in Chapter 4 (see page 47). Therefore, we use here two related measures.

Instead of the percentage of on-time job sets, we now measure the Percentage of job sets that are Completed before the Makespan Estimate ($PCME$). This performance measure is obtained by replacing T in equation (4.11) by the makespan estimate, i.e. $(1 + \gamma_k^{1-\alpha}) \cdot C_{\max}^{ex\ ante}(S_J)$ for the scheduling policy and $(1 + U^{1-\alpha}) \cdot LB(J)$ for the regression policy.

The realized capacity utilization is replaced by the effective realized capacity utilization measure:

$$ECU = \frac{\sum_{j=1}^{n_J} \sum_{i=1}^{s_j} p_{ij}}{N \cdot C_{\max}(S_J)} \tag{5.3}$$

Table 5.3 gives these performance measures for both the static case (the construction and testing data sets) and the random order arrival case. The values in Table 5.3 represent the average over all generated job sets - in the case of both the construction data set and the testing data set - and across all sixteen simulated scenarios - for the

random order arrival case.

Table 5.3: Performance measures comparison: non-selective vs. selective acceptance

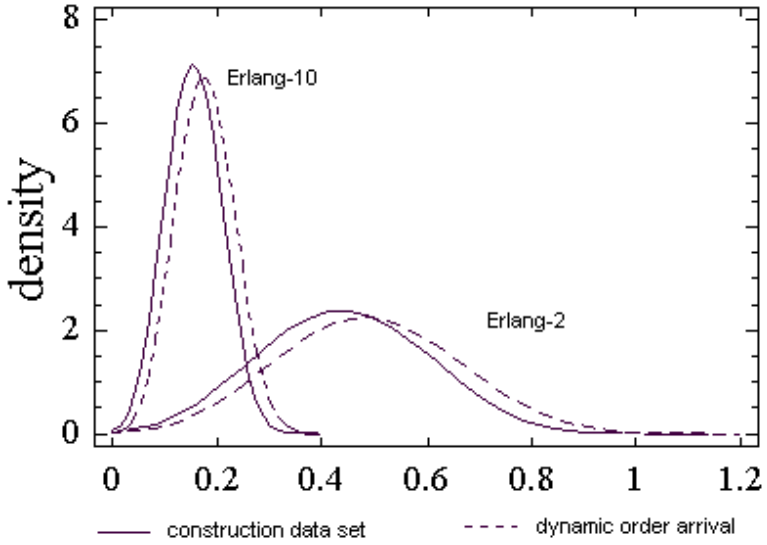
Policy	*Construction data set*		*Testing data set*		*Random order arrival*	
	$PCME$	ECU	$PCME$	ECU	$PCME$	ECU
Regression	95.02	0.4337	92.77	0.4377	90.96	0.4249
Scheduling	94.99	0.4337	94.92	0.4377	84.11	0.4492

Table 5.3 clearly shows that the control over the percentage of job sets completed before the completion time estimate in case of random order arrivals is not as good as in case of the construction and testing data sets. Remember that the job generation process is the same for random order arrivals and the construction/testing data set. The only difference is that under random order arrivals, the orders are accepted only if, according to the policy used, sufficient capacity is expected to be available to complete the resulting job set, whereas in the case of the construction and testing data sets, the job sets are generated randomly. Thus, we conclude that this performance loss is due to the selective way each policy accepts the orders in case of random arrivals.

Table 5.3 further shows that selectivity is most detrimental for the scheduling policy. Although this policy obtains a higher ECU value in the random order arrival case (0.4492) compared to the static case, it has a much lower delivery reliability. This may be explained as follows. Under the scheduling policy, an order is accepted only if a schedule can be constructed such that the resulting makespan plus an amount of slack is less than the period length. This slack is necessary to cope with the uncertainty in the processing times; it is a fraction of the ex ante makespan. This fraction - called the slack factor - has been determined empirically from the construction data set. By identifying jobs that, assuming deterministic processing times, "fit in" with the jobs already accepted, the scheduling policy selects a higher number of smaller jobs, compared to the job sets in the construction data set. As a result, the empirical distribution of the slack in the construction data set is different from the empirical distribution of the slack in the job sets that result from the acceptance procedure; see Figure 5.1.

To compare the distributions of the slack for random order arrival and static case, we apply the Kolmogorov-Smirnov test (Hollander & Wolfe 1999). This test computes the maximum difference between the distributions of the two samples. The maximum distance is 0.22, for Erlang shape parameter 2 and 0.36 for the Erlang shape parameter 10. These results imply that there is a statistically significant difference between the two distributions at the 95% confidence level, for each Erlang shape parameter.

Figure 5.1: Density trace for the slack factor for the scheduling policy



5.4 Conclusions

Order acceptance policies have a considerable impact on the mix of jobs that need to be scheduled, by refusing specific jobs from the total demand. By selecting orders with specific characteristics that maximize resource utilization, an important and often unforeseen side-effect occurs, namely the mix of orders changes in such a way that the expected delivery reliability is no longer met. In this chapter, we investigated the effects of selectivity of the scheduling and the regression policies on the delivery reliability of the set of accepted orders.

We investigated whether jobs with specific characteristics are preferred by either policy, and the conditions under which the selectivity is statistically significant. We conjectured that if the policies do not accept orders selectively, this results in job sets with similar characteristics as the arriving jobs. We therefore compared the characteristics of the job sets accepted by the two policies and the characteristics of the arriving jobs. The experiments clearly show that both policies are selective in case of high job mix variety. Both policies show a particular selectiveness; i.e. they accept jobs that have, on average, a smaller number of processing steps and a higher overlap than the average number of processing steps and the average overlap of the arriving orders.

We further investigated the impact of this selectivity on system performance. Our experiments showed that the selectivity is detrimental to the delivery reliability, especially in the case of a detailed information-based acceptance procedure. By being selective, the scheduling policy obtains a tighter schedule in the acceptance phase, compared to the construction data set, but underestimates the consequences of a

tight ex ante schedule on the realized makespan. As a result, the scheduling policy achieves a high capacity utilization, but poorer performance with respect to the delivery reliability. The regression policy is also selective, but performs much better with respect to the delivery reliability; yet, it achieves lower capacity utilization values than the scheduling policy.

The results from our study are insightful and show that selectivity is a relevant issue with high impact on delivery performance. Whereas a detailed acceptance procedure works well in maximizing resource utilization, its particular selectiveness has a negative effect for the delivery reliability. The intuition behind the explanation for this effect lies within an implicit assumption that is generally made in production control, namely that the jobs that are accepted for production are a random sample from the population of jobs that arrive at the shop.

Distinguishing between the good performance of scheduling rules in selection and their apparent poor performance in assessing the consequences of selection is an insight that deserves further research attention. In the next chapter we address the consequence of these insights for the development of production control policies, and we develop a new order acceptance policy that uses detailed scheduling rules to estimate the direct consequence of accepting an order on resource utilization and its feasibility, and aggregate, regression-based models to estimate the extra delay that is caused by constructing a high-density schedule.

Chapter 6

Hybrid policy: combining scheduling and statistical techniques

6.1 Introduction

The experimental results of the previous chapter show that the scheduling policy contributes to performance in that it can capture the consequences that a typical combination of orders have on the makespan of a job set and in that it develops tight schedules. However, we also saw that when processing times are uncertain, this tightness may be counterproductive; i.e., the realized makespan may be longer than anticipated, resulting in late deliveries. To achieve a desired delivery performance target, sufficient slack has to be added to the schedule to compensate for the effect of stochastic processing times. We conjecture that the amount of slack may depend on specific job set characteristics, such as the average overlap, and the average number of processing steps per job in the job set.

In this chapter [1] we investigate the statistical relationship between the amount of slack and specific order set characteristics, and we develop a new order acceptance policy, the *hybrid policy*. The hybrid policy uses (i) detailed scheduling to estimate the direct consequence of accepting an order on resource utilization and its feasibility, and (ii) aggregate regression models to estimate the delay that is caused by constructing a high-density schedule. We investigate the performance of the hybrid policy for a wide range of customer order and production system scenarios and we compare it with the performance of both the regression policy and the scheduling policy.

The remainder of this chapter is organized as follows. Section 6.2 discusses the development of the hybrid policy. In Section 6.3 we evaluate the performance of the

[1] The content of this chapter is joint work with J.C. Fransoo and J.M.W. Bertrand and has appeared in Ivanescu *et al.* (2004b).

hybrid policy, by means of simulation. We present our conclusions in Section 6.4.

6.2 Hybrid policy

The previous chapter showed the strengths and the weaknesses of using detailed scheduling-based information for order acceptance. While the strength of the scheduling policy is its capability of estimating the direct consequence of accepting an order on resource utilization and its feasibility, and in developing tight schedules, its weakness is its poor performance in assessing the consequences of a tight ex ante schedule on the actual makespan. Recall that under the scheduling policy, an order is accepted only if a schedule can be constructed such that the resulting makespan plus an amount of slack is less than the period length (see equation 4.5). This slack is necessary to cope with the uncertainty in the processing times and is a fraction of the ex ante makespan. This fraction has been determined empirically from the construction data set. By being selective, the scheduling policy significantly changes the mix of jobs in the accepted job sets, compared to the jobs in the construction data set. The result of this selectivity is that a high capacity utilization is obtained, but the slack added is not sufficient to compensate for the uncertainty in the processing times, resulting in late deliveries. In this section, we develop a new order acceptance policy, the *hybrid policy*, that corrects for constructing a high-density schedule, and takes into account the change in the job mix when determining the amount of slack. To estimate the correct amount of slack, we propose to use a multiple regression model that uses a limited number of specific job set characteristics. We expect this policy to give better delivery performance - without loosing much of the beneficial effects of the selectivity on the capacity utilization.

In summary, the hybrid policy accepts an order only if a schedule of the resulting job set J can be constructed such that:

$$(1 + \hat{\delta}^{1-\alpha}) \cdot C_{\max}^{\text{ex ante}}(S_J) \leq T \tag{6.1}$$

where $\hat{\delta}^{1-\alpha}$ denotes the slack factor that is determined using a regression model of specific job set characteristics under a $100(1-\alpha)\%$ delivery performance target constraint.

Slack factor estimation The relative increase of the makespan - denoted by $\delta(S_J)$ and defined in equation (4.6) - may depend on specific job set characteristics. For example, a job set containing jobs with a high number of processing steps and low overlap may experience a larger makespan increase than job sets containing jobs with a small number of processing steps and high overlap. We statistically investigate here the relationship between the makespan increase and a number of job set characteristics, and we develop a multiple regression model to estimate the slack factor needed to compensate - in the order acceptance phase - for this makespan increase.

In Chapter 3 we saw that the following job set characteristics have a significant con-

tribution in explaining the variation in the interaction margin: (i) workload balance of resource types $\rho_{\max}$, (ii) average number of processing steps per job in the job set μ_s, (iii) average overlap of processing steps in the job set μ_g, (iv) expected processing times variation in the job set $cv^2_{E[p]}$, (v) number of jobs in the job set n_J, (vi) squared coefficient of variation of the processing times cv^2_p.

We now investigate the statistical relationship between these job set characteristics and the relative makespan increase. We use the same construction data set as for the interaction margin models. We determine regression models following the same procedure as in Section 3.6. We consider both simple models - which include only the main effects - and more complex models - which include main effects and two-way interactions between these six job set characteristics. To determine our regression models, we again use the backward and stepwise regression implemented in the SPSS software package. Among the resulting models with the lowest degree of multicollinearity, we select the model with the highest adj. R^2 value. To test the adequacy of this model, we perform a residual analysis. In Appendix C.2, we present the standardized residuals versus the predicted values plots. Because heteroscedasticity was found to be present, we apply a natural logarithm transformation of the response variable; diagnostic checks on the subsequent model confirmed the appropriateness of this transformation. Equation (6.2) gives the resulting regression equation:

$$\begin{aligned} \widehat{ln(\delta(S))} = -1.720 - 0.095 \cdot \mu_s - 0.077 \cdot \mu_g - 2.134 \cdot cv^2_{E[p]} \\ + 2.116 \cdot cv^2_p + 0.178 \cdot \rho_{\max} + 0.003 \cdot n_J. \end{aligned} \tag{6.2}$$

The adj. R^2 of this model is 0.66, which indicates that 66% of the variation in the response variable is explained by the model, and the standard error of regression ($\widehat{\sigma}$) is equal to 0.228. These two statistics indicate that the estimated regression equation fits reasonably well the means of the transformed data. We also investigate the predictive quality of this model in the testing data set. The mean estimation error (ME) is 0.01, so the model seems to produce approximately unbiased predictions.

As mentioned previously, we determine the slack factor for a delivery performance target level equal to $100(1-\alpha)\%$. Thus, similar to the interaction margin estimation, the slack factor is made equal to the $100(1-\alpha)\%$ prediction bound of the response variable:

$$\hat{\delta}^{1-\alpha} = \exp\left(\widehat{ln(\delta(S))} + t_{\alpha,n-(d+1)} \cdot \widehat{\sigma} \cdot \sqrt{1 + \mathbf{x}_*^T (X^T X)^{-1} \mathbf{x}_*} \right). \tag{6.3}$$

6.3 Evaluation of the hybrid policy

The purpose of this section is to investigate the performance of the hybrid policy and to compare it with the performance of both the scheduling policy and the regression policy. We do so by simulation, using the experimental design used in Section 4.4.1.

In the experiments, we use the same random number seeds as in the experiments of Chapter 4, to generate identical order arrivals.

We first investigate whether the hybrid policy is also selective. Chapter 5 confirmed previous research findings (see e.g. Wester *et al.*, 1994; Ten Kate, 1994) that a detailed scheduling-based order acceptance policy selects orders with specific characteristics in trying to maximize resource utilization. However, this selectivity changes the mix of jobs significantly, resulting in late deliveries. Although the hybrid policy also uses detailed scheduling information, it uses a slack estimation procedure that accounts for the change in the mix of jobs. Therefore, we expect this policy to be less selective. As a result, we expect better system performance.

The remainder of this section is organized as follows. We investigate the selectivity of the hybrid policy in Section 6.3.1. Next, we compare the performance of the hybrid policy with the performances of both the regression policy and the scheduling policy in sections 6.3.2 and 6.3.3. The performance is measured by the percentage of job sets on time, the mean job set tardiness, the mean realized capacity utilization and the feasibility performance. Finally, in Section 6.3.4 we investigate the robustness of the hybrid policy.

6.3.1 Selectivity of the hybrid policy

In this section we investigate to what extent, the hybrid policy prefers jobs with specific characteristics. Similar to the investigation in Chapter 5, we compare the characteristics of the accepted job sets and the characteristics of the job arrival stream. The same three job set characteristics are considered: the average workload per job set (μ_p), the average overlap per job in a job set (μ_g), and the average number of processing steps per job in a job set (μ_s). If the hybrid policy does not accept jobs selectively, the accepted job sets and the arriving jobs will have similar characteristics. Table 6.1 gives, for each of the three characteristics, the average difference between the values obtained for the arriving jobs - denoted by μ_p^{arr}, μ_g^{arr}, and μ_s^{arr} - and the values obtained for the accepted job sets - denoted by μ_p^{hybrid}, μ_g^{hybrid}, and μ_s^{hybrid}. The last column in Table 6.1 gives the average acceptance rate of the hybrid policy.

Examining Table 6.1, we arrive at the same conclusion as in Chapter 5: the job mix variety strongly influences the acceptance procedure. We observe smaller differences between the characteristics of the arriving jobs and the characteristics of the job sets accepted by the hybrid policy, and lower acceptance rates in case of low job mix variety - compared with the case of high job mix variety. For each scenario and each of the three characteristics, we use an one-sample t-test to detect if the mean differences in Table 6.1 are statistically significant different from zero. The individual test level is set at 0.05. The results - presented in Appendix D - show that, while in most of the low job mix variety scenarios there are no significant differences between the characteristics of the arriving jobs and the characteristics of the jobs accepted by the hybrid policy, in all high job mix variety scenarios significant differences are observed. This means that in scenarios with low job mix variety the hybrid policy does accept orders non-selectively, whereas in high job mix variety scenarios the hybrid policy is

Table 6.1: Selectivity measures for hybrid policy

Scenario	$\mu_p^{arr} - \mu_p^{hybrid}$	$\mu_g^{arr} - \mu_g^{hybrid}$	$\mu_s^{arr} - \mu_s^{hybrid}$	*ACR*
1	14.05	-0.01	0.37	0.39
2	15.59	-0.02	0.50	0.57
3	17.79	-0.02	0.52	0.40
4	13.25	-0.02	0.41	0.51
5	0.37	0.00	0.01	0.37
6	0.53	0.00	0.02	0.49
7	-0.13	0.00	0.00	0.35
8	0.82	0.00	0.04	0.44
9	17.00	-0.01	0.53	0.55
10	9.24	0.00	0.32	0.71
11	15.88	-0.01	0.54	0.54
12	13.00	-0.01	0.47	0.71
13	1.95	0.00	0.05	0.55
14	1.72	0.00	0.05	0.70
15	0.80	0.00	0.03	0.49
16	2.21	0.00	0.07	0.64

selective. However, comparing the $\mu_i^{arr} - \mu_i^{hybrid}$ values, $i \in \{p, g, s\}$, with the values in Table 5.1 (see Chapter 5, page 57) we observe that the $\mu_i^{arr} - \mu_i^{hybrid}$ values are the smallest among the three policies. This indicates that the hybrid policy is the least selective policy. This observation is also confirmed by Table 6.2, which gives the distance between the characteristics of the arriving jobs and the characteristics of the jobs accepted by the three order acceptance policies, for the scenarios with high job mix variety.

The paired t-tests indicate that indeed, for all eight scenarios but one (scenario 12) this distance is significantly smaller than the distance for both the scheduling policy and the regression policy.

Table 6.2: The distance for the hybrid policy

Scenario	$d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{regr})$	$d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{sched})$	$d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{hybrid})$
1	0.340	0.317	0.154
2	0.248	0.221	0.185
3	0.324	0.313	0.207
4	0.237	0.229	0.163
9	0.236	0.214	0.162
10	0.148	0.122	0.111
11	0.221	0.222	0.153
12	0.154	0.133	0.132

We next compare the acceptance rate (ACR) performance (see column 5 in Table 6.1 for the hybrid policy and columns 8 and 9 in Table 5.1 for the regression and the scheduling policies). We observe that in scenarios with high demand/capacity

ratio and high job mix variety (scenarios 1 to 4), the hybrid policy accepts a smaller number of jobs than the regression and the scheduling policies. This observation in addition to the fact that the hybrid policy is the least selective policy indicates that indeed, the hybrid policy corrects for the selectivity of the scheduling policy. We expect therefore better performance for this policy, especially in these scenarios. The performance comparison is addressed in Section 6.3.2.

Given that the hybrid policy is also selective, we further investigate the impact of this selectivity on system performance. As in Chapter 5, we consider the following two performance measures: the percentage of job sets completed before the makespan estimate ($PCME$), and the effective capacity utilization (ECU) (see the definitions on page 59). Table 6.3 gives these performance measures for both the static case (the construction and testing data sets) and the random order arrival case, for all three order acceptance policies.

Table 6.3: Performance measures comparison: non-selective vs. selective acceptance

Policy	*Construction data set*		*Testing data set*		*Random order arrival*	
	$PCME$	ECU	$PCME$	ECU	$PCME$	ECU
Regression	95.02	0.4337	92.77	0.4377	90.96	0.4249
Scheduling	94.99	0.4337	94.92	0.4377	84.11	0.4492
Hybrid	96.02	0.4337	95.05	0.4377	94.49	0.4418

Table 6.3 shows that, the hybrid policy succeeds in realizing $PCME$ values very close to the target values; it outperforms both the scheduling policy and the regression policy. With respect to the ECU, the hybrid policy outperforms the regression policy, but not the scheduling policy. However, it is noteworthy to mention that the hybrid policy manages to determine job sets that result in a delivery performance very close to the target values, without loosing much of the beneficial effect of using detailed scheduling information on the ECU measure.

6.3.2 Performance comparisons among the three policies

In this section we compare the performance of the hybrid policy with the performance of the regression and the scheduling policies - in the random order arrival case. The performance is again compared with respect to the mean percentage of accepted job sets completed on time (POT), mean tardiness of the job set (JST), mean realized capacity utilization (RCU), and feasibility performance (FEP) (as defined in Chapter 4, page 47).

Table 6.4 presents the first three measures for each of the 16 scenarios. Each value in this table is the average of the 15 independent planning periods. Columns $2, 3$, and 4 give the performance values for the hybrid policy under a 95% delivery performance target. For comparison reasons, we also give - for each performance measure - the difference between the scheduling policy and the hybrid policy (columns $5, 6$, and 7) and the difference between the regression policy and the hybrid policy (columns $8, 9$,

and 10). We use again a paired t-test to detect significant differences between average responses of a performance measure obtained by the scheduling and regression policies versus the hybrid policy.

Table 6.4: Simulation results: *POT*, *JST*, and *RCU* measures for the hybrid policy

Scenario	*Hybrid policy*			*Scheduling-Hybrid* *			*Regression-Hybrid*		
	POT	*JST*	*RCU*	*POT*	*JST*	*RCU*	*POT*	*JST*	*RCU*
1	95.25	2.41	0.35	-20.40	14.09	0.05	-13.30	8.41	0.01
2	95.57	0.95	0.50	-26.24	7.78	0.03	**1.84**	**-0.43**	-0.03
3	94.91	2.42	0.34	-14.56	9.55	0.03	-11.95	8.42	0.01
4	95.04	1.02	0.46	-20.05	5.01	0.04	**-7.52**	**2.32**	**-0.02**
5	91.04	3.70	0.37	**2.67**	**-0.99**	**-0.00**	**0.56**	**-0.54**	-0.02
6	99.17	0.11	0.48	-8.69	1.50	0.02	0.62	**-0.08**	-0.04
7	89.92	4.20	0.35	6.08	-2.71	-0.01	**1.97**	**-0.77**	-0.01
8	99.36	0.08	0.44	-5.28	0.90	0.02	**0.19**	**-0.02**	-0.03
9	94.32	3.14	0.35	-10.19	6.15	0.02	-4.91	**3.01**	**-0.00**
10	97.25	0.62	0.47	-11.09	2.55	0.03	**-5.04**	**1.92**	-0.02
11	94.77	2.60	0.34	-10.50	6.80	0.02	-7.57	4.23	**0.00**
12	96.59	0.73	0.46	-9.39	2.13	0.01	**-2.00**	**0.46**	-0.03
13	90.93	3.57	0.37	**1.50**	**-0.68**	**-0.00**	**2.67**	**-1.11**	-0.02
14	99.47	0.08	0.47	-5.18	0.79	0.02	**0.42**	**-0.06**	-0.03
15	92.05	3.49	0.34	4.80	-2.36	-0.01	**-4.69**	**2.77**	**-0.01**
16	99.44	0.08	0.43	-2.88	0.48	**0.01**	**-1.44**	**0.28**	-0.02

* the numbers highlighted in bold indicate that the difference is not significant at 95% confidence level, where t-critical=2.145

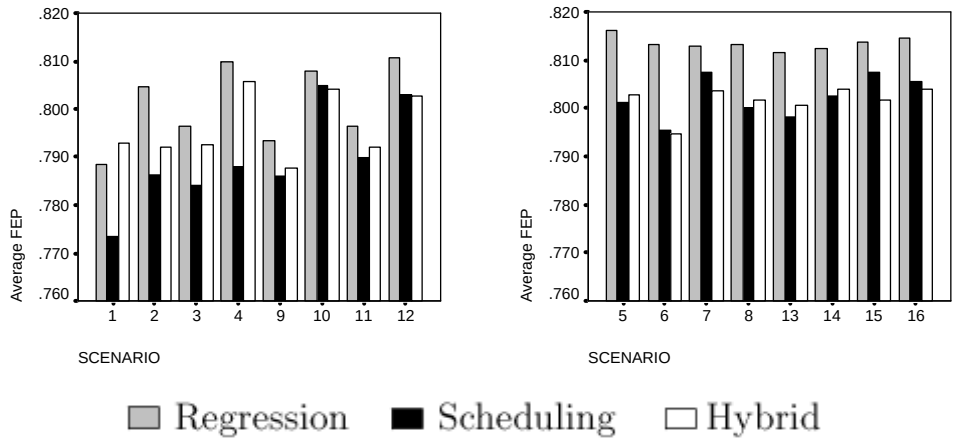
Table 6.4 shows for most scenarios a consistent Pareto improvement when comparing the hybrid and the regression policies. This reinforce our findings in Chapter 4 that using detailed scheduling information indeed improves performance.

We focus now on the comparison between the scheduling policy and the hybrid policy. Table 6.4 shows that the hybrid policy provides major performance gains with respect to the on-time delivery performance, especially for scenarios with high job mix variety (scenarios 1 to 4, and 9 to 12). These results indicate that indeed, the hybrid policy corrects for the selectivity of the scheduling policy. With respect to the realized capacity utilization, the hybrid policy obtains lower values than the scheduling policy. This is to be expected given the well-known inverse relationship between resource utilization and on-time performance. However, it is noteworthy that, in case of low demand/capacity ratio and high job mix variety (scenarios 9 to 12), the hybrid policy manages to determine job sets that result in a delivery performance very close to the target values, without loosing much of the beneficial effect of using detailed scheduling information on the *RCU* measure. Nevertheless, the poor control on the delivery performance of the scheduling policy makes the comparison between the scheduling policy and the hybrid policy difficult, if both the capacity utilization and the delivery performance criteria are considered. To obtain further insight on which of the two

policies should be used in a particular situation, we perform additional experiments in which we tune the slack factor such that similar performance is obtained with respect to one criterion (i.e. the delivery performance). This is addressed in the next section.

The last performance measure we discuss is the feasibility performance. The feasibility performance - presented in Figure 6.1 - ranges between 0.76 and 0.82. The hybrid policy clearly outperforms the scheduling policy in cases with high job mix variety and high demand/capacity ratio. This is the result of the fact that the scheduling policy accepts a larger number of small jobs, particularly in these scenarios, and therefore a tighter schedule is obtained. In a tight schedule, a high number of jobs will result is a higher number of no-wait restrictions violations. In the case of low job mix variety, the difference between the scheduling policy and the hybrid policy is almost undistinguishable. Furthermore, in cases with low job mix variety, a higher feasibility performance is obtained. However, this holds for all three acceptance policies.

Figure 6.1: Feasibility performance for the three policies



a. high job mix variety b. low job mix variety

6.3.3 Pareto comparison of scheduling and hybrid policies

As we previously mentioned, the poor control of the scheduling policy over delivery performance makes the comparison with the hybrid policy difficult, if both the realized capacity utilization and the realized job set service level criteria are considered. Obviously, given that both a high capacity utilization and a high delivery reliability are desirable, in situations where both policies reach the pre-specified job set service level target, a comparison of the realized capacity utilization would determine the choice of which of the two policies to be used in a particular situation. In this section we perform such a comparison. However, to obtain the pre-specified job set service level target, we need no additional experiments for the hybrid policy; for the scheduling policy, however, a large number of experimental runs have to be made to

tune the slack factor. This implies a large computational effort. Recall that the CPU time needed on a Pentium 1.7 GHz is about 18 seconds to evaluate a single order acceptance decision based on the scheduling policy. With an average of 100 arriving orders per planning period and 15 planning periods, about 8 hours computer time are required for each scenario. Therefore we select only the scenarios with high levels of demand/capacity ratio and job mix variety (scenarios 1 through 4) for our Pareto comparison.

For each of the scenario 1 through 4, we obtained the following values for the slack factor, based on additional experiments: $0.76, 0.26, 0.71$ and 0.25. Table 6.5 gives the POT, JST, and RCU performance measures for the scheduling policy - when using these slack factor values - under random order arrival.

Table 6.5: POT, JST, and RCU measures for the scheduling policy

Scenario	POT	JST	RCU
1	95.57	2.08	0.36
2	94.64	0.92	0.51
3	94.53	2.53	0.34
4	96.43	0.76	0.47

To identify if, for each scenario, the performance values in Table 6.5 differ significantly from the performance values obtained by the hybrid policy (see Table 6.4), we performed a paired t-test. Table 6.6 presents the results of the 12 ($= 4$ scenarios $* 3$ performance measures) paired t-tests.

Table 6.6: Paired t-test results: hybrid policy vs. scheduling policy

Scenario	POT		JST		RCU	
	t-value	P-value	t-value	P-value	t-value	P-value
1	-0.308	0.763	0.668	0.515	-2.178	0.047
2	1.073	0.301	0.128	0.900	-1.462	0.166
3	0.228	0.823	-0.124	0.903	-0.928	0.369
4	-1.907	0.077	1.498	0.156	-1.889	0.080

The results indicate that in all but one scenario (namely scenario 1), for similar delivery performance (i.e. no significant differences between the hybrid policy and the scheduling policy with respect to the POT and JST measures), similar values for the realized capacity utilization are obtained by both policies. This would point to equal performances in terms of these three performance measures. For scenario 1, the scheduling policy obtains a higher realized capacity utilization. However, the difference in performance is small. Thus, we conclude that, the additional effort of using a more sophisticated method, i.e. a regression model, to determine the slack - required by the hybrid policy - is justified in the increase of the control over the delivery performance obtained.

6.3.4 Robustness of the hybrid policy

The regression model used to estimate the slack factor for the hybrid policy is developed to cover a large variety of production situations in batch process industries. Production situations in batch process industries, however, may differ with respect to job variety, demand/capacity ratio, workload balance and processing times uncertainty. It is therefore important to investigate the sensitivity of the hybrid policy to these differences.

Recall that, we developed the hybrid policy to correct for the selectivity of the scheduling policy. Since selectivity is most detrimental to the delivery reliability, we will discuss the robustness of the hybrid policy with respect to the *POT* performance measure. Table 6.4 revealed that for each of the scenarios 1 through 4, the hybrid policy realized performance values very close to the 95% target. For the other scenarios, however, performance values are ranging from 89.92% for scenario 7 to 99.47% for scenario 14, implying substantial deviations from the target value.

Now, we investigate the main and interaction effects of the four experimental factors (i.e. demand/capacity ratio (A), the job mix variety (B), the workload balance (C), and the uncertainty level (D)) on the *POT* performance measure. Since the use of common random numbers across scenarios makes it invalid to use the ANOVA procedure, we use the procedure described in Chapter 4: we transform each experimental factor to fall onto -1 and 1 at their low and high values, we use regression analysis to obtain a model that estimates the main and interactions effects for planning period, and we use a two-sided one-sample t-test to test if each of the observed main effects and two-way interactions are significant. Table 6.7 presents the results of our one-sample t-test.

Table 6.7: One sample t-test results

Main effects & two-way interactions	*t statistic*	*P-value*
A	-1.117	0.283
B	0.389	0.703
C	0.514	0.615
D	-14.601	0.000
AB	0.057	0.956
AC	0.720	0.483
AD	0.828	0.422
BC	0.604	0.556
BD	7.287	0.000
CD	-0.399	0.696

A: demand/capacity ratio, B: job mix variety, C: workload balance, D: uncertainty level

Table 6.7 shows that only the uncertainty level D and the job mix variety-uncertainty level interaction (BD) have significant effects on this performance measure. This indicates that the hybrid policy appears to be fairly robust to changes in demand/capacity

ratio, job mix variety, and workload balance.

A closer examination of the results in Table 6.4 suggests that under scenarios with low job mix variety (scenarios 5 to 8 and 13 to 16), the hybrid policy tends to be more "nervous" (or responsive) to variations in the processing time uncertainty, resulting in over- or under- compensation. Moreover, the same observation holds for the regression policy. This strongly indicates that, in the model estimation phase, the situations with low job mix variety had little impact on determining the regression model parameter values, and thus, they may not be adequately described by the final predictive model. Essentially, there exists a trade-off between robustness and optimal performance. Using a single predictive model to cover all the variants of production situations may lead to a robust model, but the resulting model may not perform so well within certain regions of the parameter space. This suggests that, to develop an order acceptance policy that also achieves (near) optimal performance, we may need to determine different regression models for different production situations. For instance, different regression models may be developed for production situations characterized by low product mix variety. We address this issue in the next section.

6.3.4.1 Specific models

The results in the previous section indicate that different regression models may need to be developed for different production situations. More precisely, we saw that in the case of low job mix variety, both the regression policy and the hybrid policy are very responsive to variations in the processing time uncertainty. We hypothesized that this is due to the fact that the situations with low job mix variety had very little impact on determining the regression parameter values in the model determination phase. We test this hypothesis in this section. Namely, we determine regression models for the production situations with low product mix. We next use these specific regression models to support order acceptance decisions in settings with random order arrivals and production situations with low product mix. Our hypothesis is true if better performance measures are obtained with the specific models compared to the performance obtained with the general models, which cover a very large variety of production situations in batch process industries.

Specific regression models In this section we determine specific regression models for both the regression policy and the hybrid policy. To determine these models, we use the following construction data set. From the initial construction data set that we generated (see Section 3.4, page 26), we select only the cases with low product mix (i.e. $s_j \sim U(4,7)$ and $F_{E[p]} \sim U(15,35)$ in Table 3.1). This yields a construction data set with 19 500 observations. The testing data set is obtained in a similar way, yielding a data set with 5 500 observations. We determine the regression models following the procedure described in Section 3.6. We obtained the following regression equations:

$$\begin{aligned} ln(\widehat{I(S_J)}) = -1.586 + 0.060 \cdot \mu_s - 0.294 \cdot \mu_g - 3.080 \cdot cv^2_{E[p]} \\ + 1.007 \cdot cv^2_p + 1.589 \cdot \rho_{\max} - 0.003 \cdot n_J \end{aligned} \tag{6.4}$$

$$\begin{aligned} ln(\widehat{\delta(S)}) = -1.987 - 0.156 \cdot \mu_s + 0.263 \cdot \mu_g + 4.747 \cdot cv^2_{E[p]} \\ + 2.460 \cdot cv^2_p + 0.188 \cdot \rho_{\max} + 0.002 \cdot n_J \end{aligned} \tag{6.5}$$

For equation (6.4), the adj. R^2 and the $\widehat{\sigma}$ are equal to 0.80, and 0.091 respectively. Comparing these values with the values obtained by model A in Section 3.6, Table 3.2 (adj. $R^2 = 0.77$ and $\widehat{\sigma} = 0.107$) we observe that the specific model has a higher adj. R^2 value and a smaller standard error of regression. We may therefore expect better system performance when this model is used to support order acceptance decisions in a setting with random order arrivals. The same holds for the slack estimation model (i.e. the model given by (6.5)). For this model, we obtained an adj. R^2 equal to 0.70 and a $\widehat{\sigma}$ equal to 0.201 - compared to the model given by (6.2) for which we obtained adj. R^2 equal to 0.66 and a $\widehat{\sigma}$ equal to 0.228.

We also investigate the predictive quality of these two models on the testing data set. We obtain a mean estimation error equal to 0.023 for the interaction margin specific model and -0.023 for the slack factor specific model. These values indicate that both specific models produce approximately unbiased predictions.

6.3.4.2 Simulation experiments

In this section we perform simulation experiments to investigate whether specific models perform better when used to support order acceptance decisions than general models, which cover a very large variety of production situations in batch process industries. The two regression models we developed in the previous section (see equations (6.4) and (6.5)) are now used by the regression policy and the hybrid policy respectively, to support order acceptance decisions in a setting with random order arrivals (see equations (4.4) and (6.1)). Recall that orders are accepted under a job set service level target equal to $1 - \alpha$. Therefore, for both the regression policy and the hybrid policy we use the $100(1 - \alpha)\%$ upper prediction bound for the interaction margin and the slack factor respectively.

The experimental design in this section is similar to the one in Section 4.4.1. This time, however, we consider only the production situations with the low job mix variety (see Table 4.1 on page 45). This means that we now consider only the following three experimental factors: demand/capacity ratio, workload balance, and the uncertainty level of the processing times. The full factorial design yields $2^3 = 8$ scenarios. For an easy comparison, we keep the scenario notation used in the previous chapters.

The results of our experiments are presented in Table 6.8, which shows the percentage of job sets on time (POT), the mean job set tardiness (JST) and the mean realized capacity utilization (RCU).

Comparing the Tables 6.4 and 6.8, we conclude that performance improvements are indeed obtained by using specific regression models. With respect to POT, Table 6.8 reveals that for all scenarios, both policies realize performance values exceeding the 95% target value. The biggest performance improvement may be observed in the

Table 6.8: Simulation results for production situations with low product mix

Scenario	*Regression policy*			*Hybrid policy*		
	POT	*JST*	*RCU*	*POT*	*JST*	*RCU*
5	97.15	1.07	0.34	96.91	1.11	0.36
6	99.87	0.01	0.44	98.72	0.20	0.48
7	94.61	2.22	0.32	97.57	0.86	0.33
8	99.04	0.13	0.41	98.00	0.32	0.44
13	97.07	1.18	0.34	96.88	1.00	0.36
14	99.81	0.04	0.44	99.31	0.11	0.47
15	94.88	2.23	0.32	98.05	0.66	0.32
16	98.96	0.18	0.41	98.85	0.16	0.44

cases with high uncertainty in the processing times (odd numbered scenarios). For these scenarios, however, we see a small decrease in RCU. We also observe a slight improvement with respect to the JST performance measure.

6.4 Conclusions

In Chapter 5 we saw that both the scheduling policy and the regression policy are selective in the kind of jobs they accept. While the scheduling policy accepts orders with specific characteristics that maximize resource utilization, this particular selectiveness deteriorates the delivery reliability. This is due to the fact that the slack added to cope with uncertainty in the processing times is not sufficient to compensate for the selectivity. The regression policy is also selective, but better estimates the slack needed to cope with stochastic processing times.

This insight motivated the development of a new order acceptance policy - *the hybrid policy* - which combines the strengths of both the detailed and aggregate acceptance procedures. The hybrid policy uses simulated annealing techniques to estimate the direct consequences of accepting an order on resource utilization and its feasibility, and regression techniques to estimate the delay that is caused by constructing a high-density schedule and the uncertainty in the processing times. In a simulation study, we compared the performances of the hybrid policy with the performance of both the scheduling policy and the regression policy.

Given that both the scheduling policy and the regression policy proved to be selective, we first investigated whether the hybrid policy also prefers jobs with specific characteristics. The results showed that although the hybrid policy is also selective in the cases with high job mix variety, it is the least selective policy.

Simulation experiments further showed that the hybrid policy provides major performance gains with respect to the on-time delivery performance measure, outperforming the scheduling and the regression policies. Being the least selective policy, the hybrid policy manages to determine job sets that result in delivery performance very close to the 95% target, without loosing much of the beneficial effects of using detailed

scheduling information on utilization.

The regression model used to estimate the slack factor for the hybrid policy was developed to cover a large variety of production situations in batch process industries. However, in the simulation study, the model is applied to specific production situations. It is therefore interesting to investigate how robust is the hybrid policy to differences in production situations.

We found that the *POT* performance is not affected by changes in the job mix variety, demand/capacity ratio, or workload balance. This indicates that the hybrid policy is fairly robust, and captures the effects of the demand/capacity ratio, job mix variety, or workload balance differences quite well. However, the performance of the hybrid policy is affected by variations in the processing time uncertainty. Whereas in case of high job mix variety the performance differences are only weakly related to uncertainty levels differences, for scenarios with low job mix variety the results showed that the hybrid policy tends to be more responsive to variations in the processing time uncertainty, resulting in over- or under-compensation. This indicates that production situations with these characteristics (i.e., a low variety in the product mix and different levels of processing time uncertainty) had little impact on the estimation of the regression parameter values; thus, they may not be adequately described by the predictive model.

Therefore we further investigated whether the use of different regression models (i.e. regression models developed for different scenarios, namely with low levels of job mix variety) could provide better performance than the use of a single model, valid for the entire parameter space. We found that indeed, a regression model developed for specific production situations, and then applied in that production situation to support customer order acceptance decisions, provides significant performance improvements with respect to on-time delivery performance.

Chapter 7

Limited data problem: bootstrap solution

7.1 Introduction

In the previous chapter we saw that regression modelling does provide the decision makers with a powerful and relatively straightforward tool for supporting order acceptance decisions in complex settings with orders arriving randomly over time and stochastic processing times, especially when regression is combined with detailed scheduling. However, the question then arises whether this approach can be successfully applied in practice. In the previous chapters, the regression models we developed were tested extensively through simulation. These simulations used a large variety of shops and job sets to estimate the coefficients of the regression models. Application of such models in real life assumes that sufficient historical data (regarding customer orders and the production system) is available for estimating these regression coefficients with acceptable accuracy. Unfortunately, this assumption does not always hold: the quantity of historical data at hand may be rather limited, or not all the available data may be relevant, because the system and its environment changed. We denote this problem as the *limited data problem* in production control, and we address this problem in this chapter[1].

It is well known that a regression model can be used reliably for prediction if and only if the relationship between the independent variables and the dependent variable does not change. Changes in technology, raw materials, customer attitudes, and needs can have a lasting impact on estimated regression relationships. If there are indications that the relationship between independent and dependent variables changes, it becomes necessary to collect a new set of data in order to re-estimate the regression coefficients. This implies that predictions are made from a small historical database.

[1]The content of this chapter is joint work with J.M.W. Bertrand, J.C. Fransoo and J.P.C. Kleijnen and has appeared in Ivanescu *et al.* (2004a).

In this chapter, we investigate to what extent the limited data problem impacts the performance of our order acceptance policies.

Although the problem we address is of major practical interest, the research presented in this chapter is simulation based; i.e., we do not use real data but simulated data, allowing us to conduct extensive controlled experiments. For the production department described in Chapter 2, we generate shop-specific historical data by mimicking the planner's actions. After investigating the magnitude of the limited data problem, we develop a solution procedure based on the bootstrap principle. The bootstrap was introduced by Efron (1979) as a computer-based method for measuring the accuracy of statistical estimates. It implies re-sampling - with replacement - of a given (limited) sample of i.i.d. observations. In our research, we use re-sampling to generate additional data (namely job sets) with the right mixture of variety across the job sets and characteristics similar to the observed historical data. Next, we investigate whether performance improvements are obtained through this bootstrapped data set.

The remainder of this chapter is as follows. In Section 7.2 we investigate to what extent limited data affects the performance of the regression and the scheduling policies. Recall that both the regression policy and the hybrid policy are using a regression model. In order to limit the computer time of the experiments, among the policies that use regression analysis we chose to investigate the effect of limited data only for the regression policy, as this policy requires less computer time. In Section 7.3 we detail the bootstrap method we propose for generating additional data. In Section 7.4 we investigate the performance of this method in a setting with random order arrival and processing times for the regression and the scheduling policies. In Section 7.5 we present our conclusions.

7.2 Effects of limited data

Both the regression policy and the scheduling policy discussed in Chapter 4 require some parameter estimates (namely the regression coefficients and the slack factor; see equations (4.4) and (4.5)). In a real life application, the best estimation quality may be obtained when the parameters are estimated from historical production data of a specific production department. As we mentioned in Section 7.1, in real life, data may be rather limited. Therefore, we now investigate the sensitivity of the order acceptance policies to the size of the historical database. We do so by means of simulation. We develop a simulation model of a single production department that has the characteristics described in Chapter 2. The simulation experiments are carried out in three stages: (i) run the simulation model to obtain data (i.e., the historical database), (ii) estimate the policies' parameters, and (iii) evaluate the performance of the resulted order acceptance policies. These stages are addressed in the following sub-sections.

7.2.1 Generating the historical data base

As we mentioned above, to estimate the order acceptance policies' parameters, shop-specific historical data are required. We run the simulation model and generate a sufficiently large "historical" database of the production department. This database contains information regarding the accepted customer orders and the production system, for a large number of planning periods ($n_H = 512$). We refer to this database as the *construction data set*:

$$CONS = \{J_{t,t+1} | t = 1, ..., n_H\} \tag{7.1}$$

Recall that $J_{t,t+1}$ denotes the set of orders that are accepted in period t and that are due at the end of period $t + 1$ (see Section 2.2.1).

The setting for the experiments in this chapter is similar to the one described in Chapter 4 (see page 44). For computational reasons, however, we now consider only the situation with high demand/capacity ratio, high job mix variety, high workload balance, and high uncertainty in the processing times. We again consider a job set service level target equal to 95% ($\alpha = 0.05$).

Given that the construction data set contains the jobs accepted in a certain planning period, there must exist a mechanism to accept or reject arriving jobs. We use the following workload-based rule: orders are accepted as long as (i) the total workload does not exceed a specified maximum workload, and (ii) the workload per resource type does not exceed the available capacity per resource type (Raaymakers *et al.*, 2000b). Each resulting job set is scheduled and executed by the production department. In order to properly capture the effect of stochastic processing times when developing the regression models, we simulate the execution of the job set several times. Analogous to Chapter 4, we perform 250 independent realizations of the processing times, which yields i.i.d. observations $\{C_{max,i}(S_{J_{t,t+1}});\ i = 1, ..., 250\}$ on the makespan for each job set $J_{t,t+1}$.

The size of the construction data set is given by the number of planning periods, namely n_H (see (7.1)). We expect this size to affect the performance of the order acceptance policies. To investigate this effect, we consider six construction data sets with sizes ranging from 12 to 175 planning periods, namely $n_H \in \{12, 25, 50, 100, 150, 175\}$. The interpretation for these values is as follows: a construction data set of size 12 would correspond to one year production, 25 to about two years, etc. These construction data sets are obtained by randomly sampling from the large "historical" database we generated.

7.2.2 Regression parameters estimation

Each of the six construction data sets is used to estimate the parameters of the regression policy. Recall that the regression model used by the regression policy estimates the average interaction margin by using the following six regressors (see Section 3.5):

- workload balance of resource types $\rho_{\max}$,
- average number of processing steps per job in the job set μ_s,
- average overlap of processing steps in the job set μ_g,
- expected processing times variation in the job set $cv^2_{E[p]}$,
- the number of jobs in the job set n_J,
- squared coefficient of variation of the processing times cv^2_p.

Given that - for all construction data sets - the processing times in each job set are identically distributed (namely, $P_{ij} \sim$ Erlang-2), the sixth regressor, cv^2_p, is constant in the experiments in this chapter. Thus, we consider as regressors only the first five characteristics. However, we still want our regression models to account for the processing time uncertainty. Therefore, instead of regressing the interaction margin, we regress the $100(1-\alpha)$th quantile of the empirical distribution of the interaction margin $I(S_{J_{t,t+1}})$ of a given job set $J_{t,t+1}$. Since for each job set $J_{t,t+1}$, 250 i.i.d. observations for the actual makespan are available, it allows us to determine an empirical distribution for $I(S_{J_{t,t+1}})$. We use the direct-simulation quantile estimator implemented in the SPSS statistical package (see equation (4.7) on page 43) and we denote this quantile by $I^{1-\alpha}$.

In the following we detail the regression models. We again use the SPSS software package for our analyses. To measure the fit of the regression equations to the data in the construction data set, we again compute the adjusted coefficient of multiple determination (adj. R^2), and the standard error of regression ($\widehat{\sigma}$). We also check the regression models for multicollinearity. For all the models we obtained that the VIF value (see equation (3.14)) for the n_J variable was larger than 10. Thus, we eliminate this variable from the list of regressors.

We perform a residual analysis to test the adequacy of the estimated regression models. The plots of standardized residuals versus predicted values plots presented in Appendix C.3 indicate no violation of the basic regression assumptions. In Table 7.1, we present the name, the adj. R^2, and $\widehat{\sigma}$ for the models we developed.

Table 7.1: Measures of fit for the regression models based on n_H observations

Data base size (n_H)	*adj.R^2*	$\widehat{\sigma}$
12	0.85	0.044
25	0.82	0.065
50	0.79	0.062
100	0.85	0.061
150	0.79	0.068
175	0.80	0.069

We use an F-test to test the significance of each regression model as a whole, namely, all coefficients of the regression model, except for the intercept, are equal to zero.

For the regression models presented in Table 7.1, the P-value is 0.000 (see Appendix E.4), so this null hypothesis is rejected. The relatively high adj. R^2 values and low $\widehat{\sigma}$ values in Table 7.1 indicate that the models give accurate predictions. Table 7.2 gives the regression parameters' least squares estimates.

Table 7.2: Least squares estimates of the regression parameters

n_H	*Regressors*				
	Intercept	μ_s	μ_g	$cv^2_{E[p]}$	$\rho_{\max}$
12	-0.277	0.010	-0.766	-1.637	3.209
25	-1.707	0.121	-0.025	0.842	2.722
50	-0.896	0.117	0.110	-0.418	2.209
100	-1.575	0.136	0.287	0.407	2.443
150	-1.469	0.116	0.177	0.406	2.515
175	-1.278	0.133	0.347	-0.269	2.340

We further evaluate the predictive performance of the models via a testing data set. This data set contains 100 new job sets generated independently of the job sets in the construction data set. The quality of the estimation models is evaluated through the mean prediction error (ME) and the square root of the mean square prediction error ($\sqrt{MSE}$). Additionally, we compare the percentage of variability in the new data explained by the model (R^2_{pred}) with the adj. R^2 of the building model. The results are presented in Table 7.3.

Table 7.3: Predictive quality of the regression models in the testing data set

n_H	ME	$\sqrt{MSE}$	R^2_{pred}
12	0.0061	0.1199	0.35
25	0.0262	0.0910	0.62
50	0.0242	0.0863	0.66
100	0.0253	0.0835	0.68
150	0.0244	0.0839	0.68
175	0.0251	0.0839	0.68

The mean prediction error is nearly zero for all models, so the regression models seem to produce unbiased predictions. Comparing Tables 7.1 and 7.3, we observe that $\widehat{\sigma}$ is smaller than $\sqrt{MSE}$, for all models. Furthermore, the R^2_{pred} is less than the adj. R^2 from the construction phase. These two observations indicate that the regression models do not predict new data as well as they fit the existing (historical) data. However, except for the models based on 12 and 25 observations, the degradation of performance is not severe. We conclude that the regression models we developed are likely to be successful as predictors, except for the models based on the two smallest data sets ($n_H = 12$ or 25).

7.2.3 Slack factor estimation

As we mentioned in Chapter 4, the slack factor $\gamma_k^{1-\alpha}$ (see equation (4.5)) is added in order to compensate for the effect of stochastic processing times. The same procedure as in Section 4.3.2 is now used to compute this factor. Table 7.4 gives the values for the slack factor and a 95% confidence interval for these values. It is not surprising to observe that the larger the construction data set, the smaller the confidence interval is.

Table 7.4: Slack factor values for scheduling policy, estimated from n_H observations

n_H	$\gamma_2^{0.95}$	95% *confidence interval*
12	0.62	[0.591, 0.643]
25	0.64	[0.627, 0.658]
50	0.65	[0.641, 0.667]
100	0.64	[0.632, 0.648]
150	0.64	[0.631, 0.646]
175	0.64	[0.637, 0.651]

7.2.4 Impact of limited data on performance

In this section, we perform experiments to examine if the performance of the two order acceptance policies is indeed affected by the size of the construction data set. In this second stage, the estimated parameters (e.g. the regression coefficients and $\gamma_2^{0.95}$) are now frozen and used by the policies . We repeat the experimental setting of the data collection phase. This time, however, the arriving orders are accepted or rejected according to each of the policies. We simulate a planning horizon of one year; i.e., 12 replications of a planning period are performed.

We denote the regression policy that uses estimates based on a data base of size n_H by $Regr_{n_H}$. Similarly, $Sched_{n_H}$ denotes the scheduling policy that estimates the slack factor from n_H observations. Note that the slack factor $\gamma_2^{0.95}$ is the same for the sizes $n_H = 25, 100, 150$ and 175. This yields three models for the scheduling policy.

Given that the parameters were determined under a 95% job set service level target, we are interested in the ability of effectively meeting this target. This may be characterized by the average percentage of job sets completed on time (POT). The results are presented in Table 7.5.

Table 7.5 shows that none of the proposed policies reaches the pre-specified target. However, the larger the construction data set is, the better the performance (i.e. the average *POT* is closer to the target and the confidence interval is tighter). We observe that the models developed in the case of very limited data ($n_H = 12$) give the poorest performance, as expected. Such a small sample size is likely to be encountered in practice, because relevant historical information about the accepted orders may be available only over a horizon of one year or less (which means 12 observations or less).

Table 7.5: Average POT and 95% confidence interval

Policy	*POT*	95% *conf. interval*
$Regr_{12}$	84.93	[77.07, 92.80]
$Regr_{25}$	88.27	[83.89, 92.65]
$Regr_{50}$	88.37	[83.52, 93.22]
$Regr_{100}$	88.43	[84.60, 92.27]
$Regr_{150}$	88.67	[84.90, 92.43]
$Regr_{175}$	89.63	[84.99, 94.28]
$Sched_{12}$	76.43	[70.33, 82.53]
$Sched_{50}$	81.77	[76.69, 86.84]
$Sched_{25;100;150;175}$	79.43	[76.23, 82.64]

7.3 Bootstrap solution for limited data problem

In the previous section we saw that the size of the construction data set affects the performance of both the regression policy and the scheduling policy. Therefore, in a situation with very limited historical data (e.g., $n_H \leq 12$), application of these policies may be jeopardized. What is required is a large number of job sets with the right variety across the jobs and with characteristics similar to the observed historical data. To generate these additional data, we propose a method based on the bootstrap principle.

The bootstrap principle consists of repeated random re-sampling - with replacement - of the original observations (Efron & Tibshirani, 1993). Bootstrapping is an approach to statistical inference that makes few assumptions about the underlying probability distribution that describes the data. The basic approach assumes that the original observations are i.i.d. observations. Using these original data as an approximation to the unknown population density function, data are re-sampled with replacement from the observed sample to create an empirical sampling distribution for the statistic under consideration. In the following subsection we present the classic non-parametric bootstrap method and its application to regression. Next we describe our solution procedure to the limited data problem that is based on the bootstrap principle.

7.3.1 Classical bootstrap application to regression models

In this section, we briefly present the bootstrap method and its application to regression models, as discussed in the seminal book on bootstrapping by Efron & Tibshirani (1993). The name "bootstrap" originates from the expression "pulling yourself up by your own bootstraps", and refers to the basic idea of sampling with replacement from the data at hand.

Let us consider the following situation. A random sample $\mathbf{x} = (x_1, x_2, ..., x_n)$ from an unknown probability distribution F has been observed:

$$F \rightarrow (x_1, x_2, ..., x_n), \tag{7.2}$$

and we wish to estimate a parameter of interest θ on the basis of $\mathbf{x}$. For this purpose, we calculate an estimate $\widehat{\theta}$ from $\mathbf{x}$. The bootstrap was introduced in 1979 as a computer-intensive method for estimating the standard error of $\widehat{\theta}$.

Bootstrap methods depend on the notion of *bootstrap sample*. Let $\widehat{F}$ denote the empirical distribution function, i.e. the discrete distribution that puts probability $1/n$ on each value $x_i, i = 1, 2, ..., n$. A bootstrap sample is defined to be a random sample of the same size n drawn with replacement from $\widehat{F}$, say $\mathbf{x}^* = (x_1^*, x_2^*, ..., x_n^*)$:

$$\widehat{F} \rightarrow (x_1^*, x_2^*, ..., x_n^*) \tag{7.3}$$

The star notation indicates that $\mathbf{x}^*$ is not the actual data set $\mathbf{x}$, but a re-sampled version of $\mathbf{x}$. Next, a *bootstrap replication* of $\widehat{\theta}$ - denoted by $\widehat{\theta}^*$ - can be computed from this bootstrap sample $\mathbf{x}^*$. By drawing many independent bootstrap samples, say B, and evaluating the corresponding bootstrap replications, the distribution of $\widehat{\theta}$ may be estimated from the empirical distribution of $\widehat{\theta}_b^*(b = 1, ..., B)$.

We now turn to the application of the bootstrap method to regression models. Two approaches to construct bootstrap samples may be distinguished: bootstrapping pairs and bootstrapping residuals. In the following we briefly describe these two approaches, and refer to Efron & Tibshirani (1993) (see pp. $105 - 121$) for details.

Let us consider the standard linear multiple regression model:

$$y_i = \mathbf{c}_i\boldsymbol{\beta} + \epsilon_i \quad \text{for} \quad i = 1, ..., n \tag{7.4}$$

where y_i is a real number called the response, $\mathbf{c}_i$ is a $1 \times k$ vector $\mathbf{c}_i = (c_{i1}, ..., c_{ik})$ called the vector of regressors, $\boldsymbol{\beta}$ is a $k \times 1$ vector of unknown parameters, and ϵ_i is the error term. The error terms ϵ_i in (7.4) are assumed to be a (i.i.d.) random sample from an unknown error distribution F having expectation 0,

$$F \rightarrow (\epsilon_1, \epsilon_2, ..., \epsilon_n) = \boldsymbol{\epsilon} \quad (E_F(\boldsymbol{\epsilon}) = 0). \tag{7.5}$$

The data set $\mathbf{x}$ for a linear regression model consists of n points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ where each $\mathbf{x}_i$ is itself a pair, say

$$\mathbf{x}_i = (\mathbf{c}_i, y_i). \tag{7.6}$$

The goal of the linear regression analysis is to infer $\boldsymbol{\beta}$ from the observed data $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$. Let C be the $n \times k$ matrix with the ith row $\mathbf{c}_i$ (i.e. the design matrix), and $\mathbf{y}$ the $n \times 1$ vector $\mathbf{y} = (y_1, y_2, ..., y_n^T)$ (the response vector). The ordinary least squares (OLS) estimator of $\boldsymbol{\beta}$ is given by

$$\widehat{\boldsymbol{\beta}} = (C^T C)^{-1} C^T \mathbf{y} \tag{7.7}$$

and the OLS residuals by:

$$\widehat{\boldsymbol{\epsilon}} = \mathbf{y} - C\widehat{\boldsymbol{\beta}}. \tag{7.8}$$

Bootstrapping pairs Each bootstrap sample is constructed by selecting n-tuples $(\mathbf{c}_i, y_i)$ from the original data set. So, the bootstrap data set $\mathbf{x}^*$ has the form

$$\mathbf{x}^* = \{(\mathbf{c}_{i_1}, y_{i_1}), (\mathbf{c}_{i_2}, y_{i_2}), ..., (\mathbf{c}_{i_n}, y_{i_n})\} \tag{7.9}$$

where $i_1, i_2, ..., i_n$ is a random sample of the integers 1 through n.

Bootstrapping residuals To generate $\mathbf{x}^*$, we first select a random sample of bootstrap error terms

$$\widehat{F} \rightarrow (\epsilon_1^*, \epsilon_2^*, ..., \epsilon_n^*) = \boldsymbol{\epsilon}^* \tag{7.10}$$

where each ϵ_i^* equals any one of the n values $\hat{\epsilon}_j, j = 1, ..., n$ with probability $1/n$.

Then the bootstrap responses y_i^* are generated according to (7.4)

$$y_i^* = \mathbf{c}_i\widehat{\boldsymbol{\beta}} + \epsilon_i^* \quad \text{for} \quad i = 1, ..., n. \tag{7.11}$$

Then the bootstrap data set $\mathbf{x}^*$ equals $(\mathbf{x}_1^*, ..., \mathbf{x}_n^*)$, where $\mathbf{x}_i^* = (\mathbf{c}_i, y_i^*)$.

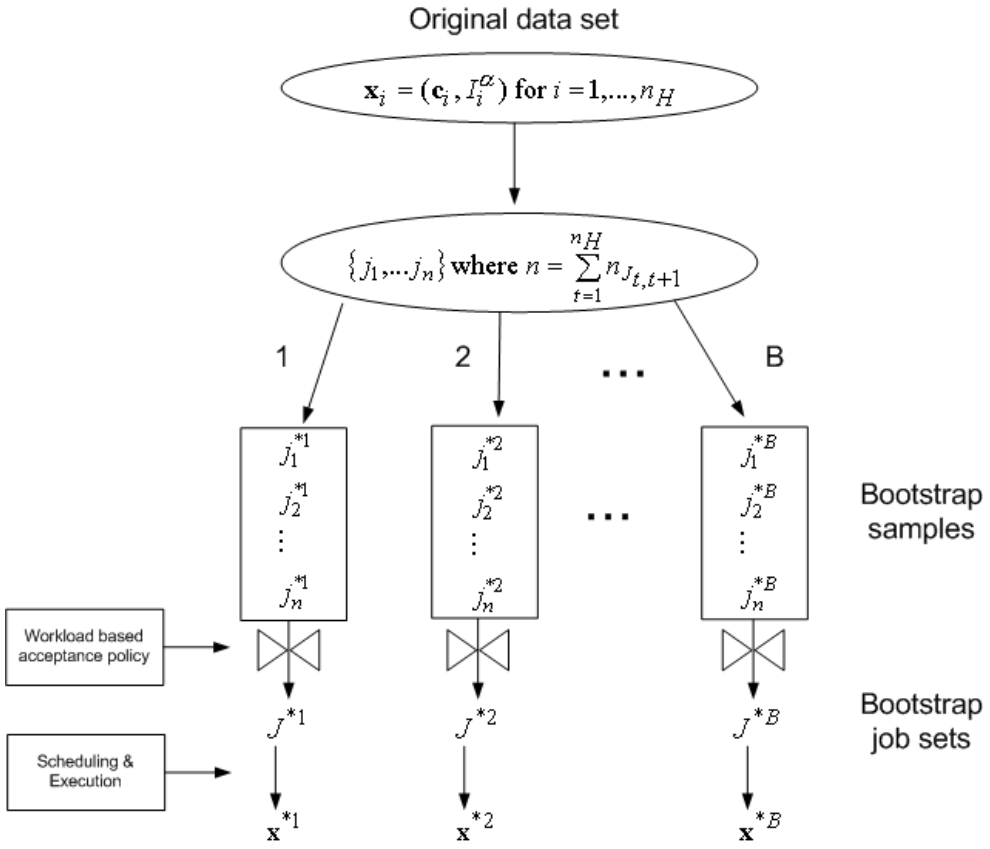
The difference between the two methods is that in the latter the $\mathbf{c}_i$ are regarded as fixed. It is assumed that the basic regression model is correct and that the residuals can be regarded as i.i.d. If, however, residuals have different variances or when errors are present in the regressors, then bootstrapping residuals will yield erroneous results. Bootstrapping pairs, on the other hand, is less sensitive to model assumptions. Furthermore, if the assumptions underlying bootstrapping residuals are met, bootstrapping pairs will yield approximately the same results (Efron & Tibshirani, 1993) (see pp. 113).

7.3.2 Proposed bootstrap method

The two approaches in the previous section are generally used for estimating the standard error or confidence intervals for $\widehat{\boldsymbol{\beta}}$, the least squares estimate of $\boldsymbol{\beta}$. However, to solve the limited data problem, we are interested in generating additional job sets rather than statistical inference. In this section, we therefore use the bootstrap principle for generating these additional job sets, called *bootstrap job sets*. Bootstrapping assumes that the observed data is a good estimate of the unknown population density function. So, we assume that the sample consisting of the jobs from all the accepted job sets is a good estimate of the population consisting of all the accepted jobs. Therefore, we re-sample with replacement from this observed sample, and generate a number of B additional job sets. The bootstrap principle ensures that this

set of B bootstrap job sets is a proxy for a set of B independent real job sets. Figure 7.1 presents schematically our procedure.

Figure 7.1: Proposed bootstrap method



In Figure 7.1, $\mathbf{x}$ denotes the original data set for the regression model which consists of n_H observations $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_H}$ where each $\mathbf{x}_i$ is a pair, i.e. $\mathbf{x}_i = (\mathbf{c}_i, I_i^{1-\alpha})$ with $\mathbf{c}_i$ denoting the vector of regressors and $I_i^{1-\alpha}$ the response variable, i.e. the $100(1-\alpha)$th quantile of the interaction margin (see Section 7.2.2).

Let $\{j_i : i = 1, ..., n\}$ denote the sample of all accepted jobs, where $n = \sum_{t=1}^{n_H} n_{J_{t,t+1}}$ and $n_{J_{t,t+1}}$ denotes the number of jobs in the job set $J_{t,t+1}$ $(t = 1, ..., n_H)$. The job arriving stream across all the planning periods is a sample with independent observations since we assume independent planning periods with each job arriving into the system being different and no precedence relations among jobs. However, given that a workload-based order acceptance procedure is used to accept or reject the arriving jobs, the accepted jobs contained in each job set may not be a random sample from the arriving job stream due to selectivity (Raaymakers, 1999). Nevertheless,

we do not expect this characteristic to violate the independence assumption. Our procedure is as follows.

1. We re-sample with replacement from the originally observed sample $\{j_1, ..., j_n\}$ of accepted jobs, to generate a bootstrap sample $\{j_1^*, ..., j_n^*\}$.

2. We assume that the jobs in the bootstrap sample are the jobs that arrive to the system during a planning period, according to a Poisson arrival process.

3. At each arrival moment, a decision to either accept or reject the job has to be made. This decision is based on the same acceptance procedure that was used to accept the jobs in the original historical data set (i.e., a workload based procedure). The result is a new job set, denoted by J^*. The simulated annealing algorithm described in Chapter 2 is used to obtain the ex ante schedule (S_{J^*}) and its corresponding ex ante makespan ($C_{\max}^{exante}(S_{J^*})$).

4. We simulate the execution of the bootstrap job set J^*. As in Section 7.2.1, 250 independent realizations of the processing times are generated that yield i.i.d. observations $\{C_{max,i}(S_{J^*}); i = 1, ..., 250\}$ for the actual makespan - which allow us to obtain the $100(1-\alpha)$th quantile of the empirical distribution of the interaction margin $I(S_{J^*})$ - the response variable.

5. We repeat steps 1 to 4 B times; we take $B = 500$. The replicate b, $b = 1, ..., B$ gives the bootstrap job set J^{*b}.

This procedure generates B bootstrap job sets. These job sets, together with the original job sets forming the historical data set, form the new construction data set.

$$CONS^* = \{J_{t,t+1} | t = 1, ..., n_H\} \bigcup \{J^{*b} | b = 1, ..., B\} \tag{7.12}$$

Based on this augmented construction data set, we estimate the parameters of the regression policy (applying the procedure described in Section 7.2.2). We hypothesize that the regression policy that uses parameter estimates based on this augmented construction data set performs better. We test this hypothesis in the next section.

7.4 Evaluation of the bootstrap solution

In this section we investigate to what extent an order acceptance policy that estimates its parameters from an augmented (bootstrap + original) construction data set improves system performance when used to support order acceptance decisions, as compared to an order acceptance policy that estimates its parameters from a small size construction data set (original only). Let the smallest construction data set (size $n_H = 12$) of Section 7.2 be the original historical data set. Applying the bootstrap procedure described in the previous section, we generate $B = 500$ additional job sets. Adding these job sets to the initial construction data set gives the augmented

construction data set, with 512 job set. We estimate the parameters for the regression policy and the slack factor for the scheduling policy by applying the procedures described in Section 7.2. We obtain the following regression model:

$$\widehat{I^{0.95}} = -1.441 + 0.133 \cdot \mu_s + 0.491 \cdot \mu_g + 0.036 \cdot cv^2_{E[p]} + 2.353 \cdot \rho_{max} \qquad (7.13)$$

and a slack factor equal to 0.63. The adj. R^2 and $\widehat{\sigma}$ are 0.77 and 0.071 for the regression model given by equation (7.13). Comparing these values with the values in Table 7.1, we observe a smaller adj. R^2 and a higher $\widehat{\sigma}$ for (7.13). However, the decrease in performance is small and gives us no reason for concern with respect to the predictive performance of this model.

Because our "real" historical data set is simulated, we can also create an ideal situation with a construction data set with 512 independent job sets. The regression model estimated from this data set is

$$\widehat{I^{0.95}} = -1.425 + 0.130 \cdot \mu_s + 0.337 \cdot \mu_g + 0.087 \cdot cv^2_{E[p]} + 2.402 \cdot \rho_{max} \qquad (7.14)$$

and the slack factor equals 0.64. The adj. R^2 and $\widehat{\sigma}$ are 0.79 and 0.068 for the regression model given by equation (7.14).

We denote the regression policy that uses estimates based on the augmented construction data set by $Regr_b$. The analogous notation holds for the scheduling policy. We compare the performance of the order acceptance policies that uses parameter estimates based on the augmented construction data set (i.e. $Regr_b$ and $Sched_b$) with the performance of the order acceptance policies that use parameter estimates based on the small construction data set (i.e. $Regr_{12}$ and $Sched_{12}$), and the large construction data set (i.e. $Regr_{512}$ and $Sched_{512}$) by means of simulation experiments. This means that six simulation runs are performed. We use the same experimental setting as in Section 7.2. For each run, we perform 36 independent replications of a planning period. Common random numbers were used throughout all runs, hence each policy deals with the same sequence of customer orders. We use the same performance measures as in Chapters 4 and 6: (i) realized capacity utilization (RCU), (ii) percentage of job sets on time (POT), (iii) job set tardiness (JST), and (iv) feasibility (FEB). Table 7.6 reports our results.

Table 7.6: Computational results

Policy	*RCU*	*POT*	*JST*	*FEB*
$Regr_b$	0.3410	92.49	3.75	0.796
$Regr_{12}$	0.3420	88.43	7.23	0.796
$Regr_{512}$	0.3449	90.82	4.85	0.795
$Sched_b$	0.3940	77.54	13.93	0.772
$Sched_{12}$	0.3930	74.74	16.41	0.774
$Sched_{512}$	0.3921	80.14	11.70	0.772

To compare the performance of $Sched_b$ with the performance of $Sched_{12}$ and $Sched_{512}$,

we use paired t-tests; that is, for each performance measure, we made two pairwise comparisons. The results are presented in Table 7.7.

The paired t-test results show that, for RCU and FEB, there are no significant differences neither between $Sched_b$ and $Sched_{12}$, nor between $Sched_b$ and $Sched_{512}$. We further observe that, by using the bootstrap construction data set to determine the slack factor - instead of the small construction data set - the performance improves significantly: $Sched_b$ manages to determine job sets that result in higher delivery performance than $Sched_{12}$, without decreasing the capacity utilization. However, $Sched_{512}$ gives significantly smaller job set tardiness and higher delivery performance than $Sched_b$.

Table 7.7: Paired t-test results for the scheduling policies in Table 7.6

Measures	$Sched_b$ vs $Sched_{12}$		$Sched_b$ vs $Sched_{512}$	
	t-value	P-value	t-value	P-value
RCU	0.426	0.672	0.588	0.561
POT	2.624	0.013	-2.279	0.029
JST	-3.097	0.004	2.448	0.020
FEB	0.893	0.378	0.214	0.831

We perform the same analysis for the three regression policies; see Table 7.8. Table 7.8 shows that, for the RCU and FEB performance measures, there are no significant differences between $Regr_b$ and $Regr_{12}$. However, the POT and JST measures show that $Regr_b$ performs significantly better. Table 7.6 shows that this policy manages to determine job sets that result in a delivery performance very close to the 95% target, and a smaller job set tardiness. We conclude that, in case of limited historical data, our re-sampling procedure gives better system performance than the case when limited historical data is used.

Table 7.8 further shows that for all performance measures, except FEB, there are significant differences between $Regr_b$ and $Regr_{512}$. With respect to the RCU measure, Table 7.6 reveals that $Regr_b$ realizes a lower RCU than $Regr_{512}$ but, this difference is very small.

Table 7.8: Paired t-test results for the regression policies in Table 7.6

Measures	$Regr_b$ vs $Regr_{12}$		$Regr_b$ vs $Regr_{512}$	
	t-value	P-value	t-value	P-value
RCU	-0.545	0.589	-7.097	0.000
POT	2.822	0.008	2.328	0.026
JST	-2.874	0.007	-2.429	0.020
FEB	-0.164	0.870	-0.997	0.326

It is remarkable that $Regr_b$ gives significantly higher delivery performance than $Regr_{512}$. To explain this result, we performed the following additional analysis. We compared

the characteristics of the job sets accepted by $Regr_b$ to those accepted by $Regr_{512}$. The same job set characteristics as in Chapter 5 are investigated; namely, the average workload per job set (μ_p), the average overlap per job (μ_g), the average number of processing steps per job (μ_s), and the acceptance rate (ACR). The results are presented in Table 7.9.

Table 7.9: Selectivity of $Regr_b$ and $Regr_{512}$

Policy	μ_p	μ_g	μ_s	ACR
$Regr_b$	118.73	0.59	4.82	40.43
$Regr_{512}$	119.86	0.58	4.86	39.65

Table 7.9 shows that $Regr_{512}$ has a significantly higher acceptance rate ($t = 3.522$, df=35). Furthermore, the jobs accepted by this policy have on average a significantly higher overlap than the jobs accepted by $Regr_b$ ($t = 3.675$, df=35). These two observations explain the better performance of $Regr_b$. More exactly, a higher number of jobs with a higher overlap result in a tighter schedule. In a highly stochastic environment, a tight schedule increases capacity utilization but decreases the delivery performance.

7.5 Conclusions

The order acceptance policies we developed in Chapter 4 were tested extensively through simulation. These simulations, however, used a large variety of shops and job sets to estimate the parameters of these policies (i.e. the coefficients of the regression models for the regression policy and the slack factor for the scheduling policy). Application of such models in real life assumes that sufficient historical data (regarding customer orders and production system) is available for estimating these parameters with acceptable accuracy. In practice, however, relevant historical data may be limited. In this chapter, we investigated to what extent this limited data problem impacts the performance of these two order acceptance policies.

For batch process industries, featuring complex job and resource structures, we first generated shop-specific historical data by simulation. This initial simulation study showed that both order acceptance policies perform less well if the size of the construction data set is small. To overcome the limited data problem, we developed a procedure based on the bootstrap principle.

Classical bootstrap methods re-sample - with replacement - the original observations. Our procedure bootstraps the original set of accepted jobs, to generate additional job sets. We assessed the performance of our bootstrap procedure by means of simulation. The results showed that the performance of the bootstrap policy (i.e. the policy that uses parameter estimates based on the augmented construction data set) clearly improves.

The results clearly demonstrate the power of extending the bootstrap principle by

applying it to the most detailed items of a data set in production control, namely the individual jobs. Rather than re-sampling the job sets, we used the individual jobs for re-sampling. This allowed us to construct new jobs sets. We believe that this principle can also be applied in other production control environments where the limited data problem may occur. We encourage further research to investigate the possible impact and limitations of our bootstrap approach.

Chapter 8

Conclusions and future research

8.1 Main research findings

In this thesis we studied the order acceptance function, which is responsible for coordinating the capacity requirements and the available capacity over time, in a multi-resource production system with overlapping processing steps, no-wait restrictions among processing steps and stochastic processing times inspired from batch process industries. In Section 1.4 we formulated the goals of this research: (i) to contribute to the development of models to support customer order acceptance decisions and (ii) to increase insight into the benefits of using regression modelling techniques to support these decisions in multipurpose batch process industries with random order arrivals and stochastic processing times.

We pursued these goals by elaborating on the following research questions:

1. How do aggregate regression models perform compared with detailed scheduling models, when used to support order acceptance decisions in settings with random order arrivals and processing times?

2. Can the strengths of both aggregate regression-based policies and detailed scheduling-based policies be combined in an improved order acceptance policy?

3. To what extent can regression analysis be used if only limited historical data is available?

In the following subsections we summarize our conclusions regarding these questions.

8.1.1 Aggregate versus detailed scheduling information

To provide the answer to the first research question, we performed a simulation study and we compared the performance of two order acceptance policies under a wide range of experimental conditions. The first policy, called the *regression policy*, uses aggregate information by means of regression techniques to estimate the actual makespan of an order set. The second policy, called the *scheduling policy*, uses detailed information by means of simulated annealing techniques and an empirically determined slack factor to estimate the actual makespan of an order set. The slack is added in order to cope with the processing times uncertainty. Under both policies, orders are accepted as long as the estimated makespan does not exceed the length of the planning period, with a pre-specified probability. These two policies were compared with respect to (i) the realized capacity utilization and (ii) the ability to effectively meet the customer requirements.

Our simulation study showed that, in situations with low job mix variety, the scheduling policy performs better. More precisely, this policy determines job sets that result in higher capacity utilization values and in a delivery performance closer to the pre-specified target. In cases with high product mix variety, however, the regression policy manages to determine job sets that result in a delivery performance closer to the pre-specified target and smaller job set tardiness than the scheduling policy. The scheduling policy, however, reaches higher capacity utilization.

The first research contribution of this thesis is that it provides insight into the behavior of both aggregate information- and detailed scheduling-based policies under stochastic production conditions. Comparison among order acceptance policies shows that detailed scheduling information at the order acceptance level remains valuable, also under stochastic production conditions. The beneficial effects of using detailed scheduling information for supporting order acceptance decisions has been previously established for deterministic production conditions (see e.g. Wester *et al.*, 1994; Ten Kate, 1994; Raaymakers *et al.*, 2000b). This is to be expected, since, in a deterministic production situation, the schedule constructed upon order acceptance and used to support the acceptance decision is the exact representation of what will be realized later by the production department. However, in a highly stochastic environment one may expect the performance of the scheduling policy to be highly affected by the uncertainty in the processing times, since the ex ante schedule constructed upon order acceptance is not anymore an exact representation of the future status of the production system. This expectation is confirmed by the results of our simulation study in situations with high variation in the job mix. In those situations, and especially when the arrival rate is high, the scheduling policy tends to selectively accept orders that "fit in" with the orders already accepted. The result is a tight schedule that maximize resource utilization. However, this selectiveness is detrimental to the delivery reliability. The experiments in Chapter 5 showed that this weakness is due to the fact that by being selective, the scheduling policy significantly changes the mix of jobs in the accepted job sets - compared to the job sets on which the slack is originally determined. Consequently, the empirically determined slack that is added into the system is not sufficient to cope with the uncertainty in the processing times. Information on the current job mix is therefore required when determining the slack.

8.1.2 Combining scheduling and regression techniques

The second research question was: "Can the strengths of both aggregate regression-based policies and detailed scheduling-based policies be combined in an improved order acceptance policy?". We showed in Chapter 5 that a detailed scheduling-based order acceptance policy contribute to performance in that it can capture the consequences that a typical combination of orders have on the makespan of an order set and in that it develops tight schedules that result in high levels of capacity utilization. However, this tightness proved to be detrimental for the delivery reliability. Recall that this policy is using simulated annealing techniques to estimate the ex ante makespan and adds a slack to this value to cope with processing time uncertainty. This slack is determined by estimating the probability distribution function of the makespan increase due to stochastic processing times. It takes into consideration the level of processing time uncertainty that is expected, but it does not explicitly consider other job set characteristics.

The analysis we performed in Chapter 5 showed that this slack underestimates the effect of the schedule's tightness on the actual makespan under stochastic production conditions, as the current mix of jobs significantly changes from the job sets on which the slack has been originally determined. We conjectured that information on the current job mix is required when determining the slack in order to obtain high levels of delivery performance. We therefore investigated the relationship between the makespan increase due to stochastic processing times and the following order set characteristics:

- workload balance of resource types,
- average number of processing steps per job in the job set,
- average overlap of processing steps in the job set,
- expected processing times variation in the job set,
- number of jobs in the job set,
- squared coefficient of variation of the processing times.

Computer experiments showed that the amount of slack needed in the system can be accurately estimated by regression models using these aggregate job set characteristics. This insight led us to develop a new order acceptance policy, called the *hybrid policy*, that combines detailed scheduling and regression models to estimate the makespan of an order set.

To answer our second research question, we performed a simulation study to investigate the performance of the hybrid policy for a wide range of customer order and production system scenarios. We also compared its performance with the performance of the regression and the scheduling policies. Simulation results presented in Chapter 6 showed that the hybrid policy significantly improved the delivery performance measure, outperforming both the scheduling policy and the regression policy. Being

the least selective policy, the hybrid policy manages to determine job sets that result in a delivery performance very close to the pre-specified target, without loosing much of the beneficial effect of using detailed scheduling information on utilization.

As mentioned in Section 1.4, when designing a new order acceptance policy we are also interested in the robustness of this policy - besides its ability to improve system performance. The results of our experiments showed that, with respect to delivery performance, the hybrid policy is fairly robust and captures the effects of the demand/capacity ratio, job mix variety, or workload balance differences quite well. However, in situations with low job mix variety the hybrid policy tends to be more responsive to variations in the processing time uncertainty, resulting in over- or under-compensation. We conjectured that production situations with these characteristics (i.e., a low variety in the product mix and different levels of processing time uncertainty) had little impact on the determination of the regression parameter values, so they may not be adequately described by the predictive model. This was confirmed by the simulation results in Section 6.3.4, which showed that using different regression models developed for different regions of the parameter space results in better delivery reliability performance than the use of a single model, valid for the entire parameter space.

Summarizing, another research contribution of this thesis is that we provide an order acceptance policy that allows jobs to be accepted such that it provides control over delivery performance. This new order acceptance policy combines the strengths of both the scheduling policy and the regression policy, and corrects for the weakness of the scheduling policy that has been uncovered by the analysis in Chapter 5 and that was discussed in the previous section.

8.1.3 Limited data problem

The third research question concerns the problem of identifying to what extent it is possible to use regression analysis to support order acceptance decisions when only limited data is available. In Chapters 4 and 6 we developed two order acceptance policies that use regression analysis and that have been tested extensively through computer simulation. Such simulation basically allows a virtually unlimited variety of job sets to be explored to determine the coefficients of the regression models. Application of such models in real life assumes that sufficient historical data on customer orders and the production system is available, to estimate these regression coefficients with acceptable accuracy. However, in real life, there may be a small amount of historical data available; this may not always be sufficient to produce good parameter estimates. We therefore investigated first to what extent limited availability of historical data impacts the performance of an order acceptance policy that uses a regression model.

The results of our simulation study showed that this impact is substantial. Regarding the ability to effectively meet customer requirements, the models developed with limited data realize the poorest performance. These results lead to the conclusion that application of regression models to support order acceptance decisions in real life

situations may be jeopardized if limited historical data are available. To overcome this problem, we proposed a bootstrap procedure. Our procedure re-samples the original set of accepted jobs, to generate additional job sets with similar characteristics as the observed historical data.

We evaluated the performance of our bootstrap method by means of simulation experiments. The results showed that the bootstrap method significantly improves performance. Another important result is that the bootstrap regression policy (i.e. the regression policy that uses parameter estimates based on the original construction data set augmented with the bootstrap job sets) reaches capacity utilization levels close to those of a regression policy that uses a very large historical data to estimate its parameters.

8.1.4 General conclusion

A general conclusion of this thesis is that, in order to improve performance in a hierarchical planning and scheduling framework, it is essential that the higher level decision makers can accurately estimate the performance of the lower level decision function. This is in line with the concept of anticipation introduced by Schneeweiß (1999), who presented a hierarchical planning framework that consists of two levels interacting with each other: the top level and the base level. Three different stages of interdependencies may be distinguished. Firstly, the top level makes some decision implying an instruction that is given to the base level. Secondly, the base level may react to this instruction so that replanning at the top level is kicked off. Thirdly, before giving its instruction, the top level takes into account the base level's relevant characteristics and anticipates the base level's reaction by either implicitly or explicitly modelling the behavior of the base level in the top level's model; this is called the anticipated base model. de Kok & Fransoo (2003) have further discussed the various types of anticipation that may exist. In general, the anticipated base model can be constructed by aggregating information and/or aggregating the base level model itself.

In this thesis, we focussed on models that estimate the production output that can be realized by the production system per planning period. We conclude that to control delivery performance in a stochastic environment, such an estimate of the production output should combine detailed scheduling and aggregate regression techniques.

8.2 Future research

8.2.1 Assumptions revisited

In Chapter 2 we justified the assumptions we made to limit the size of this research project. These assumptions, however, had an impact on developing the order acceptance policies. Therefore, some comments are needed on the application of these policies in other areas of research and in industrial practice. The following assumptions are considered:

- independent planning periods,
- single job per customer order,
- strict no-wait restrictions,
- the same Erlang shape parameter for all the processing steps in the job set, and
- specific scheduling algorithm.

First, we assumed that jobs are allocated to planning periods. At the start and at the end of each planning period, the production system is assumed to be empty. This is the case if production is not performed round-the-clock, i.e. no production is carried out during weekends. In this case, all jobs are completed before the end of the week. When constructing a schedule, start-up and close-down losses occur for each planning period. These losses are expected to be acceptable if planning periods are long relative to the processing time of the jobs. However, if production is performed round-the-clock, these losses may be avoided by considering continuous time instead of time buckets at the resource allocation (or scheduling) level. In that case, planning periods may still be used at the order acceptance level, but the coefficients of the estimation models may need to be adapted to reflect the realized production output. Furthermore, we assumed that the planning periods are independent. Nevertheless, due to the stochastic processing times, both lateness and earliness may occur. When jobs are not completed within their assigned planning period, they require capacity from the next planning period. Earliness, in contrast, causes idleness, and thus a waste of capacity. In this case, idle time can be utilized by jobs from the next period. The models we developed can still be used in the multi-period case, but again, they have to be adapted to reflect the realized production output.

Second, we assumed that each customer order results in exactly one job. Hence, there are no precedence relations between jobs allocated to a planning period. In industrial practice, however, a number of subsequent operations has to be carried out to produce a product, which means that a customer order may result in several consecutive jobs. We may still use the estimation models in that situation simply by allocating at most one job in a routing to a specific planning period. In that case, there are still no precedence relations between the jobs allocated to the same period. Consequently, the minimal throughput time of a customer order is as many periods as there are jobs resulting from the customer order, e.g. a customer order consisting of four consecutive jobs has at least a throughput time of four periods. The choice of the length of the planning periods then determines the minimal throughput time of a customer order. However, long planning periods result in high work-in-process (WIP) inventories. Alternatively, more than one job related to the same customer order may be allocated to the same planning period. In that situation, the scheduling algorithm should also meet the precedence relations between the jobs in a job set. This reduces the scheduling flexibility and may affect the production output realized. This should be reflected in the coefficients of the regression models.

Thirdly, we assumed that processing steps of a job have strict no-wait restrictions. In industrial practice, sometimes a limited waiting time is allowed. The situation with

strict no-wait restrictions is a worst-case situation. This means that in real life, the estimation models based on strict no-wait restrictions may give a pessimistic estimate of the workload and job mix that can be realized per planning period. It may be preferred to obtain estimation models with coefficients tailored to a specific situation. In addition to the six job set characteristics we considered in this thesis, we may then need another variable that represents the waiting time flexibility. Further research is required to investigate the effects of limited waiting time among the processing steps.

Fourthly, we assumed the same Erlang shape parameter for all the processing steps in the job set; in other words, we assumed that the level of uncertainty in the processing times is independent of the type of resource used to execute the processing steps. In industry, however, different resource types may result in different levels of processing time uncertainty. In that case, the estimation models may still be used, but further research is required to determine how the squared coefficient of variation of the processing times regressor should incorporate this aspect.

Finally, we assumed that the allocation of jobs to resources is done by a specific scheduling algorithm. Namely, we used a simulated annealing algorithm developed by Raaymakers & Hoogeveen (2000). The use of a different algorithm for the resource allocation decision function may require a different regression model at the order acceptance level. This is also the case for flowtime estimation in discrete manufacturing where the due-date prediction capabilities of the due-date assignment rules are affected by the different dispatching rules used at the shop floor (Chang, 1997). In this thesis, we employed a predictive-reactive scheduling approach in which a schedule is generated to optimize some performance measures (minimizing the makespan in our case) based on job completion times without considering possible disruptions at the shop floor. When the schedule released to the shop floor quickly becomes infeasible due to the dynamic nature of the production floor, it is modified (rescheduled) to restore feasibility. We used the "right-shifting" rescheduling procedure for this purpose - which is the simplest and fastest kind of rescheduling. While generally all heuristics and algorithms generate schedules that require rescheduling when feasibility can no longer be maintained, a robust schedule is of most practical use as simpler scheduling adjustments are then required. The robustness of a schedule refers to its ability to perform well under different operational environments including dynamic and uncertain conditions (Dooley & Mahmoodi, 1992). Significant research on the robustness of a schedule is due to Leon *et al.* (1994) and Daniels & Kouvelis (1995). Further research is required to investigate the effect of a robust scheduling algorithm on the performance of order acceptance policies.

8.2.2 Suggestions for further research

An interesting research topic that may follow this research is the use of regression analysis to support order acceptance and capacity loading decisions in other types of systems than the one considered in this thesis. Estimating the production output that can be realized with the available capacity is a complex problem if the level of interaction between jobs and resources and between jobs themselves is high. In capacity loading decisions in traditional job shops, queuing models are widely used

to assist the decision maker (Buzacott & Shantikumar, 1993). However, while the results of queueing analysis can often be used for simple production systems, the power of queuing models decreases as production complexity increases. In addition, research on traditional job shops disregards the behavior of the jobs between two consecutive operations, i.e. it assumes that intermediate buffers (or storages) have infinite capacity and that jobs can be stored for an unlimited amount of time. In practice, there are many examples of industrial settings in which buffer capacity has to be taken into account, at least at some stages of the production process. Moreover, there may be limits on the amount of time that a job can spend in a buffer between two consecutive operations. These additional constraints on work-in-process and intermediate inventories further complicates the capacity estimation problem, and offers increased opportunity for using regression modelling.

As we mentioned in Section 3.2, regression analysis has been previously used in job shops, for due date assignment. In this literature, the flow time of a single job is estimated based on some job and production system characteristics, in order to set due dates. Also, the use of the flow time estimation error to set more reliable due dates is considered in the due date assignment literature (Enns, 1993, 1994; Lawrence, 1995). In this literature, however, estimates are made for single jobs. With respect to a set of jobs, only the workload and the number of jobs are considered. The estimation models discussed in this thesis, may be used to determine flow time estimates for a set of jobs. This may give better capacity estimates when the job mix changes considerably over time. However, further research is required to investigate how our regression models may be adapted for job shop environments with no-wait or blocking constraints.

Order acceptance is concerned with accepting or rejecting customer orders such that the goals of the management are met as much as possible. In this thesis, we developed models to support order acceptance decisions based on the availability of capacity to produce the orders before their requested due date. However, as mentioned in Chapter 1, there are some other important aspects for the order acceptance problem that we did not consider. The first is the *capability* to manufacture the ordered product. The resources in the production facility all have their (interdependent) technological capabilities that together determine the 'aggregate capability' of the production facility. These capabilities primarily determine the make or buy decision regarding the whole product or some parts of the ordered product. If some product parts or processing steps are subcontracted, this will generally increase the delivery time of the ordered product. Therefore, next to the availability of the resources, the external process like subcontracting, transport and material acquisition should also be considered to determine the possible delivery time of an ordered product. Another aspect of order acceptance concerns the *costs and revenues* that result from accepting the order. Setting the price of an order is often limited due to competition. However, the relation between the price and the delivery time should receive more attention in the near future. A number of production situations are characterized by the fact that customers are willing to pay more for short lead times while, at the same time, there are penalties for late deliveries.

The job mix has a considerable impact on both capacity utilization and delivery per-

formance that can be realized. We developed regression models that only help the decision makers to estimate if a certain job set can be completed within a certain period. It does not directly indicate how a certain job mix contributes to actual capacity utilization or delivery performance. However, a decision maker may use this knowledge of the relations between job set characteristics and system performance measures to determine an adequate (optimal) job mix. A typical manufacturer in batch process industries runs various plants in order to serve the international markets. Moreover, it is common that a product can be produced in more than one plant. Knowledge about an adequate job mix that gives (near) optimal performance may be valuable in (re)designing the entire production network. For instance, orders that may disturb the optimal job mix for one plant may be produced by other plants.

Appendix A

Simulated annealing algorithm

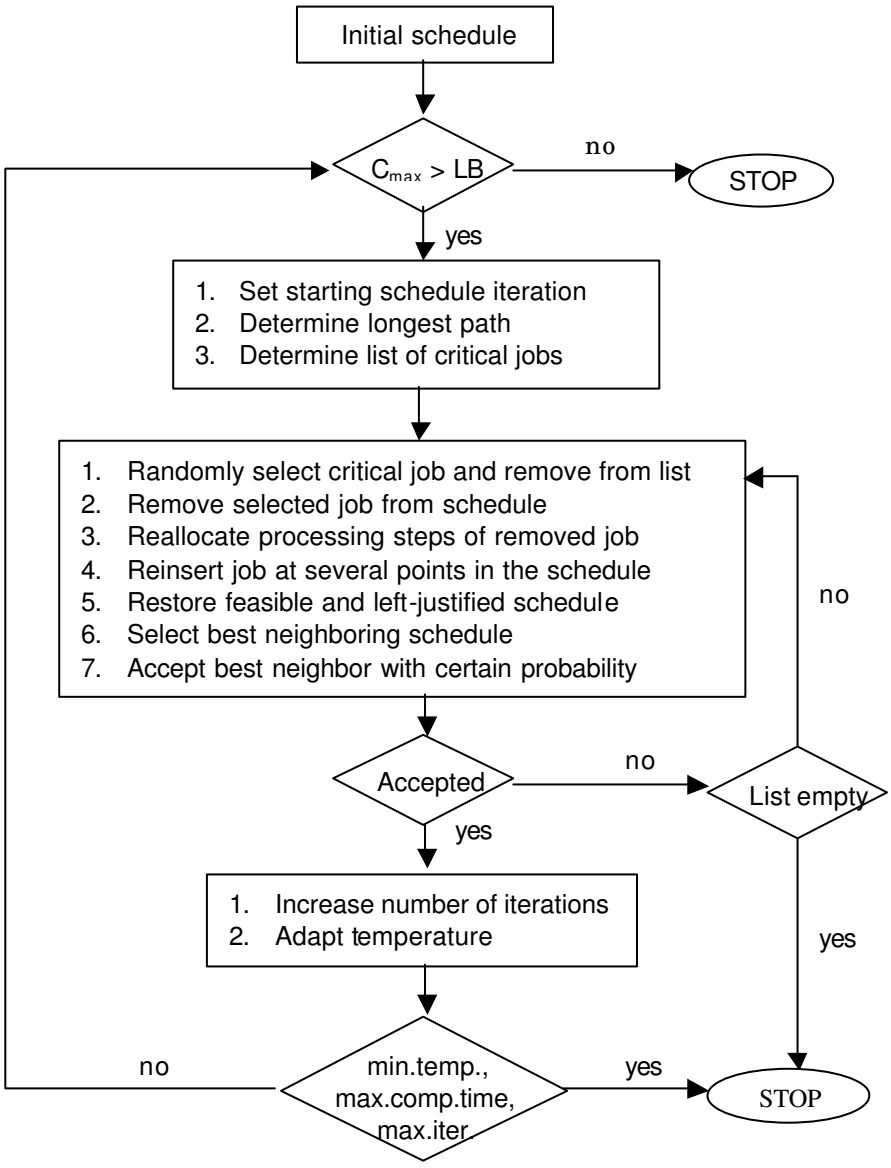
In this appendix, we discuss how the simulated annealing algorithm (SA) developed by Raaymakers & Hoogeveen (2000) is used in our experiments; we refer to their publication for details on SA. An overview of the SA algorithm is given in Figure A.1.

Raaymakers & Hoogeveen (2000) proved that the best scheduling solution results from repeating the algorithm several times. However, the variations in the quality of the scheduling solutions may have consequences for the quality of the estimation model. Therefore, we run the algorithm three times, each time with a different initial schedule, and we consider the best solution out of the three runs.

Experiments in Raaymakers & Hoogeveen (2000) showed that the dispatching rules *largest-number-of-processing-steps-first* (LNP), *bottleneck* (BNCK), and *largest-total-processing-times-first* (LTPT) give the best initial solutions. Therefore, we used these three dispatching rules to generate the initial schedules. Furthermore, the SA's so called cooling parameters are set as follows. The initial temperature depends on the size of the jobs. Since the temperature determines the probability of accepting a deterioration of the makespan, it is expected that large jobs give rise to a larger absolute deterioration of the makespan than small jobs. In order to have a reasonable acceptance probability for all job sets, we set the initial temperature equal to the average workload of a single job in a job set.

In our experiments, we chose to lower the temperature after each iteration, to limit the computational time. The algorithm stops when the minimal temperature reaches 1, when we find a makespan equal to the lower bound, or when all neighboring solutions are rejected.

Figure A.1: Overview of SA algorithm (from: Raaymakers 1999)



Appendix B

Derivation of squared coefficient of variation of actual processing times

B.1 No variation in expected processing times

Assume that all the processing steps in the job set J have the same expected processing times, i.e. $E[P_{ij}] = c, i = 1, ..., s_j$ and $j = 1, ..., n_J$. The processing times are then determined by drawing a random variable from an Erlang distribution with mean c and shape parameter k. So, the squared coefficient of variation for the effective processing times is:

$$cv_p^2 = \frac{1}{k} \tag{B.1}$$

according to the properties of an Erlang distributed random variable.

B.2 Variation in expected processing times

Assume that each processing step may be different and may have a different expected processing time $E[P_{ij}]$. We modelled this by considering $E[P_{ij}]$ as a random variable uniformly distributed $U(a, b)$. The processing times are determined by drawing a random variable X, Erlang distributed with mean $E[p_{ij}]$ and shape parameter k. In the following we compute the density function and the first two moments of the resulting processing time. We can then compute the squared coefficient of variation (which is used in the regression model).

$$f(t) = \frac{d}{dt}P(X \leq t) = \int_a^b \frac{d}{dt}P(X \leq t \mid \frac{k}{\mu} = z)\frac{1}{b-a}dz =$$

$$= \frac{1}{b-a}\int_a^b \frac{k}{z}\frac{(\frac{k}{z}t)^{k-1}}{(k-1)!}e^{-\frac{k}{z}t}dz. \tag{B.2}$$

Replacing $\frac{k}{z}$ with ω in (26) we have:

$$f(t) = \frac{k}{b-a}\int_{\frac{k}{b}}^{\frac{k}{a}} t\frac{(\omega t)^{k-2}}{(k-1)!}e^{-\omega t}d\omega. \tag{B.3}$$

Through the substitution $\omega t = s$ we arrive at the following formula for the density function of the processing time:

$$f(t) = \frac{k}{b-a}\frac{1}{(k-1)!}\int_{\frac{kt}{b}}^{\frac{kt}{a}} s^{k-2}e^{-s}ds. \tag{B.4}$$

The first two moments can now be derived straightforwardly:

$$E[X] = \int_0^\infty tf(t)dt = \int_0^\infty t\left(\frac{k}{b-a}\frac{1}{(k-1)!}\int_{\frac{kt}{b}}^{\frac{kt}{a}} s^{k-2}e^{-s}ds\right)dt =$$

$$= \int_0^\infty \left(\int_{\frac{as}{k}}^{\frac{bs}{k}} tdt\right)\frac{k}{b-a}\frac{1}{(k-1)!}s^{k-2}e^{-s}ds =$$

$$= \int_0^\infty \frac{(b^2-a^2)s^2}{2k^2}\frac{k}{b-a}\frac{1}{(k-1)!}s^{k-2}e^{-s}ds = \frac{b+a}{2}\int_0^\infty \frac{s^k}{k!}e^{-s}ds = \frac{b+a}{2} \tag{B.5}$$

knowing that $\int_0^\infty \frac{s^k}{k!} e^{-s} ds = 1$ (see the properties of the gamma function).

The second moment is computed according to the well known formula:

$$Var[X] = E[X^2] - (E[X])^2 \tag{B.6}$$

where in our case we have

$$E[X^2] = \int_0^\infty t^2 f(t) dt = \int_0^\infty t^2 \left(\frac{k}{b-a} \frac{1}{(k-1)!} \int_{\frac{kt}{b}}^{\frac{kt}{a}} s^{k-2} e^{-s} ds \right) dt =$$

$$= \int_0^\infty \left(\int_{\frac{as}{k}}^{\frac{bs}{k}} t^2 dt \right) \frac{k}{b-a} \frac{1}{(k-1)!} s^{k-2} e^{-s} ds =$$

$$= \int_0^\infty \frac{(b^3 - a^3) s^3}{3k^3} \frac{k}{b-a} \frac{1}{(k-1)!} s^{k-2} e^{-s} ds = \frac{b^2 + ab + a^2}{3} \frac{k+1}{k} \int_0^\infty \frac{s^k}{k!} e^{-s} ds =$$

$$= \frac{b^2 + ab + a^2}{3} \frac{k+1}{k}. \tag{B.7}$$

So, we have then the following formula for the second moment:

$$Var[X] = \frac{b^2 + ab + a^2}{3} \frac{k+1}{k} - \left(\frac{b+a}{2} \right)^2 = \frac{(b-a)^2}{12} + \frac{b^2 + ab + a^2}{3k}. \tag{B.8}$$

Thus, the squared coefficient of variation is:

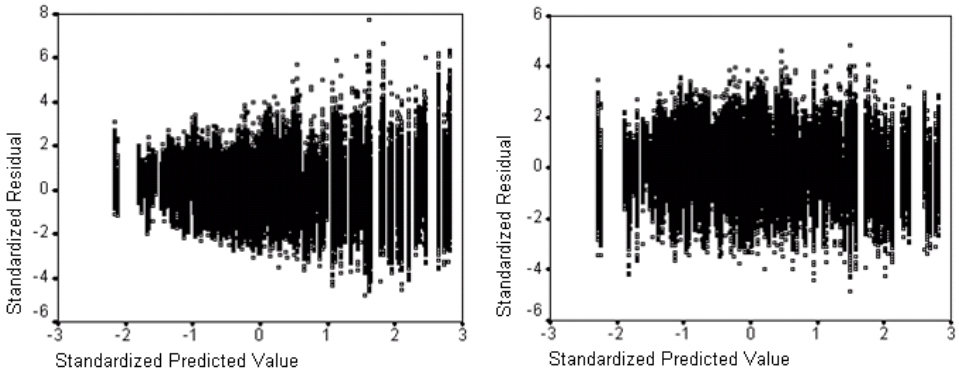
$$cv^2 = \frac{Var[X]}{(E[X])^2} = \frac{(b-a)^2}{3 \cdot (b+a)^2} + \frac{4 \cdot (b^2 + a \cdot b + a^2)}{3 \cdot k \cdot (b+a)^2}. \tag{B.9}$$

Appendix C

Residual analysis

C.1 Interaction margin models

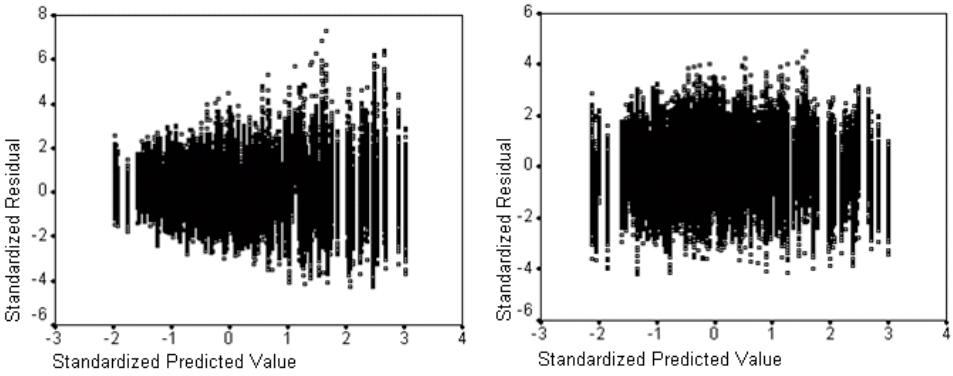
Figure C.1: Residual plots, model A



(a) before transformation

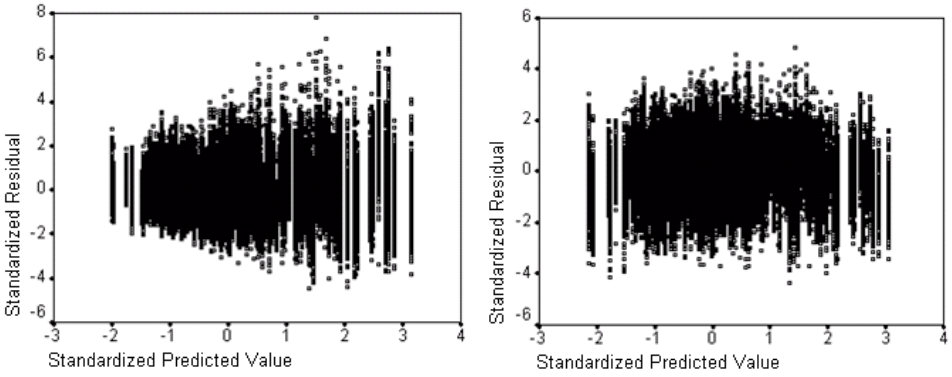
(b) after transformation

Figure C.2: Residual plots, model D



(a) before transformation (b) after transformation

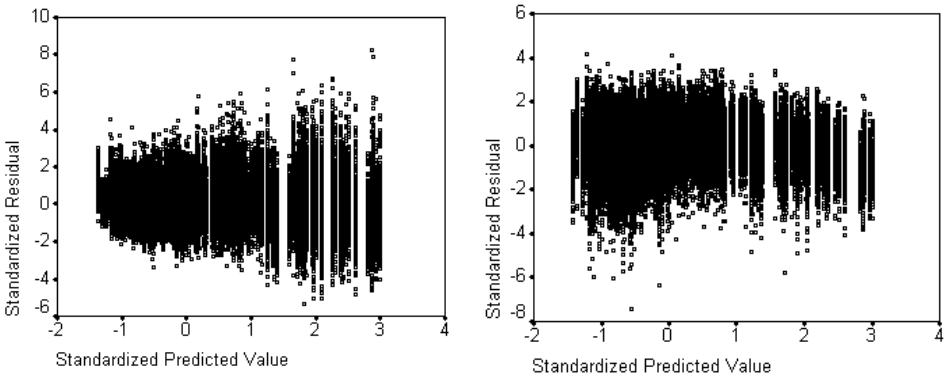
Figure C.3: Residual plots, model E



(a) before transformation (b) after transformation

C.2 Slack factor estimation model

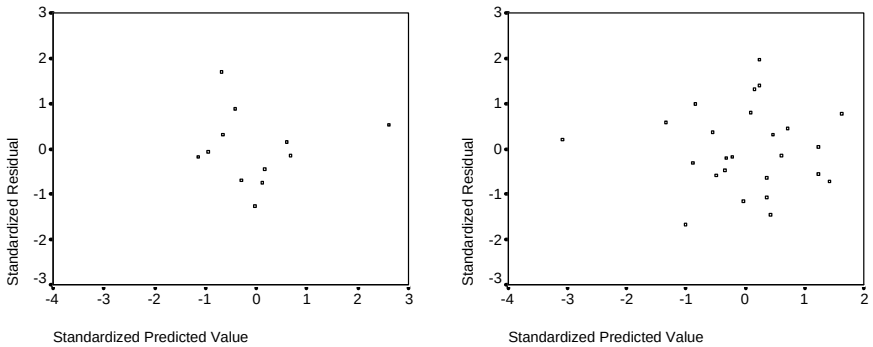
Figure C.4: Residual plots, slack factor estimation model



(a) before transformation (b) after transformation

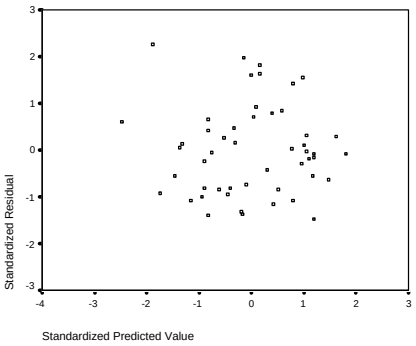
C.3 Limited data case

Figure C.5: Residual plot, limited data case

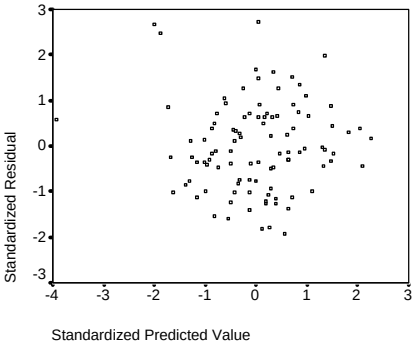


(a) Model $Regr_{12}$ (b) Model $Regr_{25}$

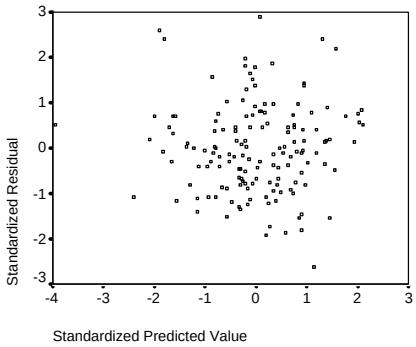
Figure C.5: Residual plot, limited data case (continued)



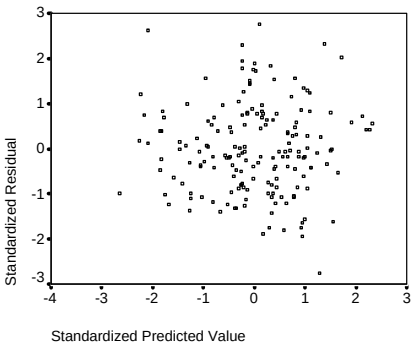
(c) Model $Regr_{50}$



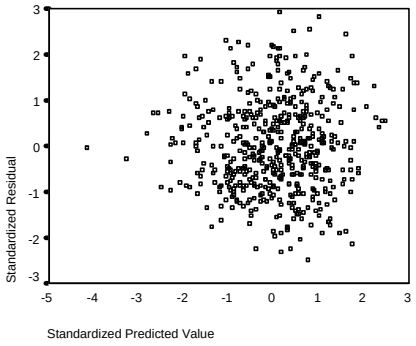
(d) Model $Regr_{100}$



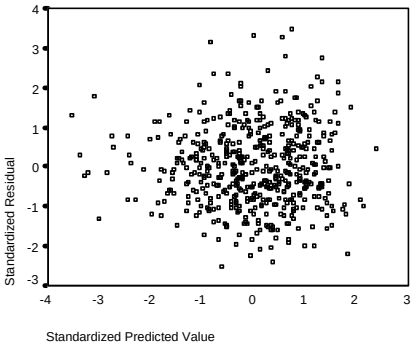
(e) Model $Regr_{150}$



(f) Model $Regr_{175}$



(g) Model $Regr_{512}$



(h) Model $Regr_b$

Appendix D

Results for t-test

For each of the sixteen scenarios, Table D.1 gives the t-statistic of the one-sample t-test we performed to detect if the mean differences observed in Tables 5.1 and 6.1 are significantly different from zero. The t-statistic is computed by dividing the mean difference (given in tables 5.1 and 6.1) by the standard error of the sampling distribution of differences (not presented here). A mean difference in Tables 5.1 and 6.1 is significantly different from zero at 95% confidence level if the corresponding absolute value of the t-statistic in Table D.1 is larger than the critical value 2.145.

Table D.1: One-sample t-test results ($df = 14$)

Scenario	$\mu_p^{arr} - \mu_p^{policy}$			$\mu_g^{arr} - \mu_g^{policy}$			$\mu_s^{arr} - \mu_s^{policy}$		
	Regr	*Sched*	*Hybrid*	*Regr*	*Sched*	*Hybrid*	*Regr*	*Sched*	*Hybrid*
1	19.66	17.73	4.07	-16.37	-14.12	-1.98	19.27	18.78	2.87
2	18.79	13.84	4.32	-10.61	-10.94	-5.61	19.65	15.08	3.97
3	14.37	4.66	4.06	-12.01	-4.99	-3.24	14.57	4.50	3.39
4	10.88	14.03	3.78	-8.63	-12.33	-3.22	12.16	16.03	3.39
5	-1.14	0.63	0.26	0.17	0.04	-0.34	-0.93	0.38	0.27
6	-1.21	2.18	0.60	-0.17	0.65	0.62	-0.79	1.70	0.51
7	-1.54	0.49	-0.10	-0.07	0.44	0.17	-1.11	0.04	-0.06
8	-0.89	2.18	0.78	-0.65	-0.19	-0.02	-0.69	1.69	0.97
9	10.66	8.77	9.75	-7.41	-5.98	-2.71	9.60	8.15	7.12
10	7.67	5.01	3.60	-5.79	-4.90	-1.87	7.68	5.42	3.40
11	10.27	9.37	8.48	-7.25	-5.26	-2.24	8.96	9.22	6.79
12	7.19	7.69	6.22	-4.58	-5.94	-3.69	6.68	6.95	5.46
13	0.00	2.84	3.16	-0.89	-0.82	-0.75	-0.04	2.28	1.82
14	2.51	3.22	3.73	-0.70	0.20	-0.38	1.71	2.56	2.88
15	1.86	0.35	1.06	-1.28	-0.71	-1.49	0.31	1.65	0.86
16	1.77	4.16	3.33	-0.74	-1.30	-0.46	2.05	3.14	2.63

We present in Table D.2 the results of the paired t-test we performed to detect significant statistical differences between $d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{sched})$ and $d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{regr})$, $d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{regr})$ and $d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{hybrid})$, and $d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{sched})$ and $d(\boldsymbol{x}^{arr}, \boldsymbol{x}^{hybrid})$.

Table D.2: Paired sample t-test results

Scenario	*df*	*t-statistic*		
		Sched vs. *Regr*	*Regr* vs. *Hybrid*	*Sched* vs. *Hybrid*
1	14	-1.947	7.782*	6.444*
2	14	-5.024*	4.952*	2.813*
3	14	-0.646	4.524*	3.873*
4	14	-0.753	3.647*	3.236*
9	14	-1.546	3.724*	2.689*
10	14	-2.291*	3.584*	1.025
11	14	0.113	3.318*	3.045*
12	14	-1.127	1.699	0.084

* significant at 95% confidence level, where t-critical=2.145

Appendix E

Results regression models

E.1 Interaction margin estimation models

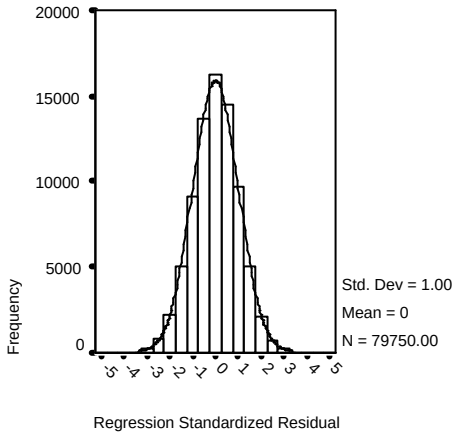
Table E.1: Results anova for the interaction margin estimation model A

	SS	df	MS	F	P-val
regression	2981.414	6	496.902	43342.071	0.000
residual	914.227	79743	0.011		
total	3895.641	79749			

Table E.2: Coefficients interaction margin estimation model A

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.984	0.009	-210.854	0.000	[-2.003,-1.966]	
μ_s	0.111	0.001	99.586	0.000	[0.109, 0.113]	1.13
μ_g	-0.130	0.010	-12.604	0.000	[-0.150,-0.110]	1.15
$cv^2_{E[p]}$	-0.883	0.004	-208.429	0.000	[-0.891,-0.875]	2.23
cv^2_p	0.911	0.003	347.897	0.000	[0.906, 0.916]	2.20
$\rho_{\max}$	1.466	0.005	282.278	0.000	[1.456, 1.477]	1.11
n_J	-0.003	0.000	-79.485	0.000	[-0.003,-0.003]	1.04

Figure E.1: Residual histogram, interaction margin estimation model A



E.2 Slack factor estimation model

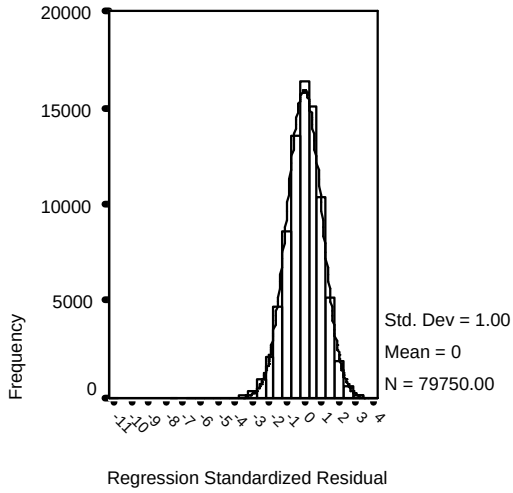
Table E.3: Results anova for the slack factor estimation model

	SS	df	MS	F	P-val
regression	8016.678	6	1336.113	25600.107	0.000
residual	4161.922	79743	0.052		
total	12178.600	79749			

Table E.4: Coefficients slack factor estimation model

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.720	0.020	-85.683	0.000	[-1.760,-1.681]	
μ_s	-0.095	0.002	-40.017	0.000	[-0.100,-0.090]	1.13
μ_g	-0.077	0.022	-3.492	0.000	[-0.120,-0.034]	1.15
$cv^2_{E[p]}$	-2.134	0.009	-236.045	0.000	[-2.152,-2.116]	2.23
cv^2_p	2.116	0.006	378.713	0.000	[2.105, 2.127]	2.20
$\rho_{\max}$	0.178	0.011	16.028	0.000	[0.156, 0.199]	1.11
n_J	0.003	0.000	49.104	0.000	[0.003, 0.004]	1.04

Figure E.2: Residual histogram, slack factor estimation model



E.3 Specific models

Table E.5: Results anova for specific models

		SS	df	MS	F	P-val
Interaction margin	regression	655.181	6	109.197	13326.414	0.000
estimation model	residual	159.726	19493	0.008		
	total	814.907	19499			
Slack factor	regression	1862.169	6	310.362	7684.941	0.000
estimation model	residual	787.238	19493	0.040		
	total	2649.407	19499			

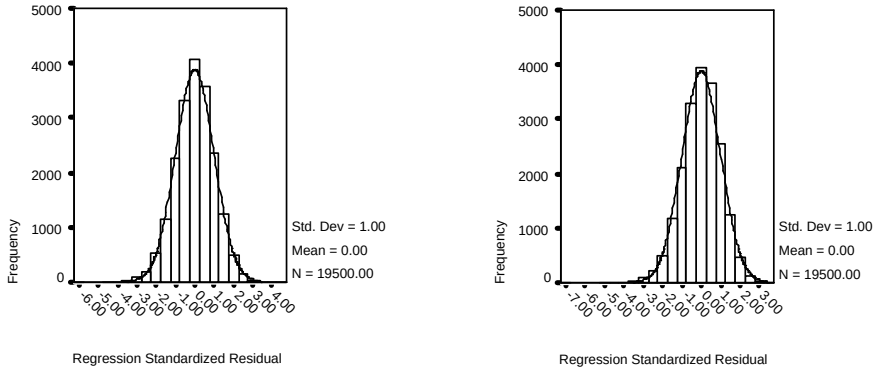
Table E.6: Coefficients interaction margin specific estimation model

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.586	0.027	-58.357	0.000	[-1.639,-1.533]	
μ_s	0.060	0.003	17.937	0.000	[0.054, 0.067]	1.02
μ_g	-0.294	0.026	-11.494	0.000	[-0.345,-0.244]	1.08
$cv^2_{E[p]}$	-3.080	0.188	-16.361	0.000	[-3.449,-2.711]	1.05
cv^2_p	1.007	0.005	190.243	0.000	[0.997, 1.018]	1.06
$\rho_{\max}$	1.589	0.008	204.967	0.000	[1.573, 1.604]	1.01
n_J	-0.003	0.000	-51.739	0.000	[-0.003,-0.003]	1.05

Table E.7: Coefficients slack factor specific estimation model

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.987	0.060	-32.943	0.000	[-2.106,-1.869]	
μ_s	-0.156	0.007	-20.906	0.000	[-0.171,-0.142]	1.02
μ_g	0.263	0.057	4.621	0.000	[0.151, 0.374]	1.08
$cv^2_{E[p]}$	4.747	0.418	11.356	0.000	[3.927, 5.566]	1.05
cv^2_p	2.460	0.012	209.214	0.000	[2.437, 2.483]	1.06
$\rho_{\max}$	0.188	0.017	10.914	0.000	[0.154, 0.222]	1.01
n_J	0.002	0.000	16.361	0.000	[0.002, 0.002]	1.05

Figure E.3: Residual histogram, specific models



(a) interaction margin estimation model

(b) slack factor estimation model

E.4 Limited data case

Table E.8: Results anova, limited data case

		SS	df	MS	F	P-val
$Regr_{12}$	regression	0.124	4	0.031	16.218	0.001
	residual	0.013	7	0.002		
	total	0.137	11			
$Regr_{25}$	regression	0.485	4	0.121	28.457	0.000
	residual	0.085	20	0.004		
	total	0.570	24			
$Regr_{50}$	regression	0.721	4	0.180	46.663	0.000
	residual	0.174	45	0.004		
	total	0.895	49			
$Regr_{100}$	regression	2.059	4	0.515	137.854	0.000
	residual	0.355	95	0.004		
	total	2.413	99			
$Regr_{150}$	regression	2.690	4	0.672	144.469	0.000
	residual	0.675	145	0.005		
	total	3.365	149			
$Regr_{175}$	regression	3.366	4	0.842	179.231	0.000
	residual	0.798	170	0.005		
	total	4.164	174			
$Regr_{512}$	regression	8.718	4	2.180	471.024	0.000
	residual	2.346	507	0.005		
	total	11.064	511			
$Regr_b$	regression	8.519	4	2.130	426.109	0.000
	residual	2.534	507	0.005		
	total	11.052	511			

Table E.9: Coefficients model $Regr_{12}$

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-0.277	0.794	-0.349	0.737	[-2.155,1.600]	
μ_s	0.010	0.044	0.229	0.826	[-0.094,0.114]	2.26
μ_g	-0.766	0.437	-1.754	0.123	[-1.799,0.267]	2.09
$cv^2_{E[p]}$	-1.637	0.862	-1.899	0.099	[-3.675,0.401]	1.15
$\rho_{\max}$	3.209	0.478	6.717	0.000	[2.079,4.339]	1.05

Table E.10: Coefficients model $Regr_{25}$

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.707	0.457	-3.738	0.001	[-2.659,-0.754]	
μ_s	0.121	0.034	3.616	0.002	[0.051, 0.192]	1.29
μ_g	-0.025	0.424	-0.059	0.954	[-0.910, 0.860]	1.18
$cv^2_{E[p]}$	0.842	0.441	1.912	0.070	[-0.077, 1.762]	1.07
$\rho_{\max}$	2.722	0.332	8.198	0.000	[2.029, 3.414]	1.18

Table E.11: Coefficients model $Regr_{50}$

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-0.896	0.301	-2.981	0.005	[-1.501,-0.290]	
μ_s	0.117	0.022	5.337	0.000	[0.073, 0.161]	1.27
μ_g	0.110	0.280	0.394	0.696	[-0.454, 0.674]	1.25
$cv^2_{E[p]}$	-0.418	0.311	-1.343	0.186	[-1.046, 0.209]	1.01
$\rho_{\max}$	2.209	0.191	11.567	0.000	[1.824, 2.593]	1.02

Table E.12: Coefficients model $Regr_{100}$

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.575	0.176	-8.791	0.000	[-1.923,-1.226]	
μ_s	0.136	0.015	9.087	0.000	[0.106, 0.166]	1.24
μ_g	0.287	0.185	1.549	0.125	[-0.081, 0.655]	1.18
$cv^2_{E[p]}$	0.407	0.193	2.110	0.037	[0.024, 0.789]	1.03
$\rho_{\max}$	2.443	0.132	18.552	0.000	[2.181, 2.704]	1.06

Table E.13: Coefficients model $Regr_{150}$

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.469	0.163	-8.999	0.000	[-1.792,-1.147]	
μ_s	0.116	0.013	9.215	0.000	[0.091, 0.141]	1.23
μ_g	0.177	0.167	1.064	0.289	[-0.152, 0.506]	1.23
$cv^2_{E[p]}$	0.406	0.173	2.348	0.020	[0.064, 0.747]	1.05
$\rho_{\max}$	2.515	0.123	20.463	0.000	[2.272, 2.757]	1.03

Table E.14: Coefficients model $Regr_{175}$

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.278	0.161	-7.934	0.000	[-1.596,-0.960]	
μ_s	0.133	0.012	10.846	0.000	[0.109, 0.157]	1.32
μ_g	0.347	0.165	2.112	0.036	[0.023, 0.672]	1.27
$cv^2_{E[p]}$	-0.269	0.167	-1.609	0.110	[-0.599, 0.061]	1.03
$\rho_{\max}$	2.340	0.109	21.454	0.000	[2.125, 2.556]	1.05

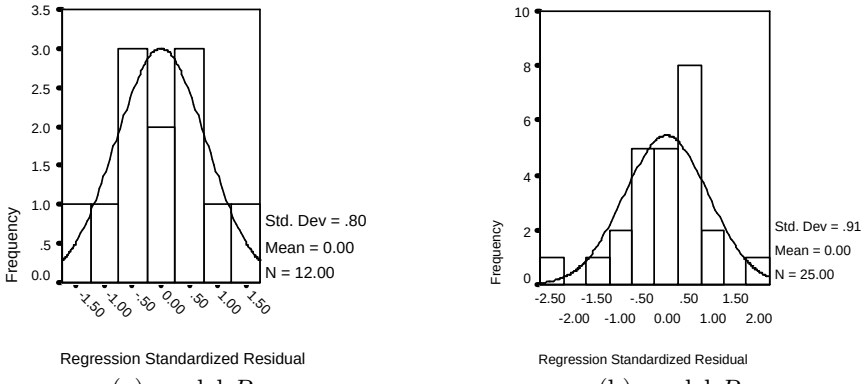
Table E.15: Coefficients model $Regr_{512}$

	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.425	0.092	-15.478	0.000	[-1.605,-1.244]	
μ_s	0.130	0.007	18.214	0.000	[0.116, 0.144]	1.22
μ_g	0.337	0.086	3.915	0.000	[0.168, 0.506]	1.20
$cv^2_{E[p]}$	0.087	0.096	0.902	0.367	[-0.102, 0.276]	1.01
$\rho_{\max}$	2.402	0.065	36.874	0.000	[2.274, 2.530]	1.01

Table E.16: Coefficients model $Regr_b$

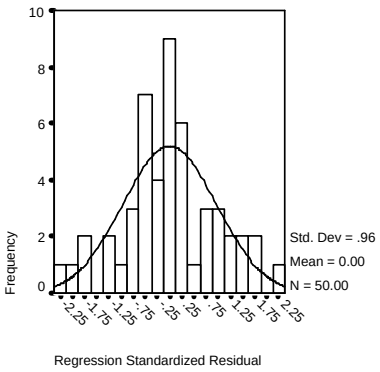
	B	st.dev. B	t	P-val.	95% conf. interval	VIF
Intercept	-1.441	0.105	-13.689	0.000	[-1.648,-1.234]	
μ_s	0.133	0.008	15.925	0.000	[0.117, 0.150]	1.40
μ_g	0.491	0.094	5.203	0.000	[0.306, 0.676]	1.40
$cv^2_{E[p]}$	0.036	0.088	0.412	0.681	[-0.137, 0.209]	1.00
$\rho_{\max}$	2.353	0.063	37.216	0.000	[2.229, 2.478]	1.00

Figure E.4: Residual histogram, limited data case

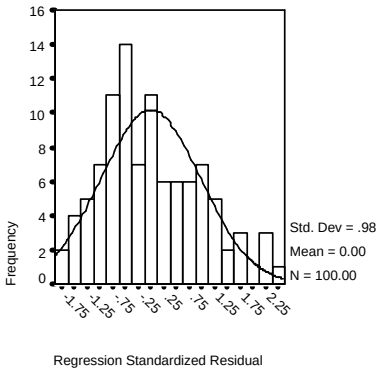


(a) model $Regr_{12}$ (b) model $Regr_{25}$

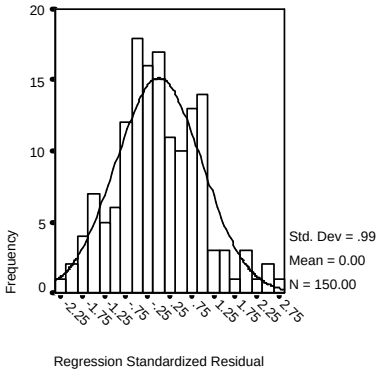
Figure E.4: Residual histogram, limited data case (continued)



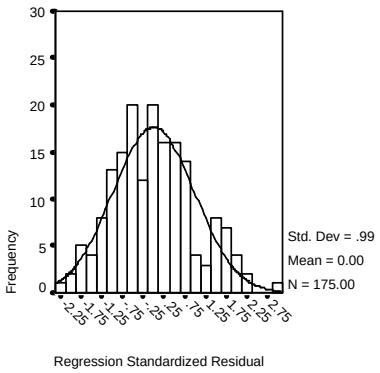
(c) Model $Regr_{50}$



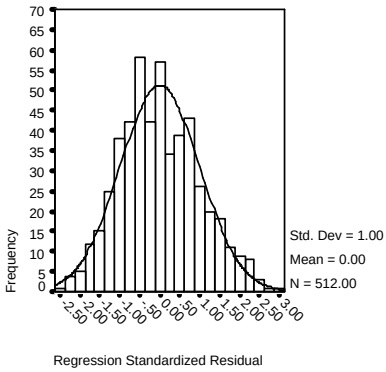
(d) Model $Regr_{100}$



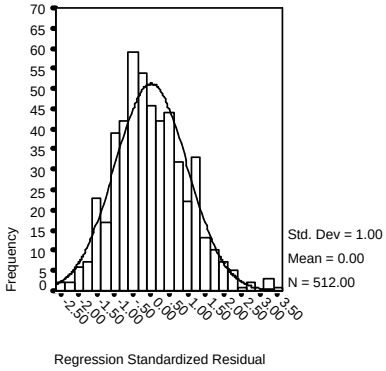
(e) Model $Regr_{150}$



(f) Model $Regr_{175}$



(g) Model $Regr_{512}$



(h) Model $Regr_b$

Appendix F

Workload-based rule

In this appendix we present the workload-based rule used in Chapter 7 to generate the historical data base. Under this rule, orders are accepted to be executed in a specific planning period as long as the total workload does not exceed a specified maximum total workload and the workload per resource type does not exceed the available capacity per resource type. The available capacity is computed as the number of resources in the production system multiplied by the length of the planning period. Consequently, an order is accepted as long as the following conditions are met for the resulting job set J:

$$\sum_{j=1}^{n_J} \sum_{i=1}^{s_j} E[P_{ij}] \leq (1-\tau) \cdot N \cdot T \tag{F.1}$$

and

$$\forall m : \sum_{j=1}^{n_J} \sum_{i \in P_m} E[P_{ij}] \leq n_m \cdot T \tag{F.2}$$

where P_m gives set of processing steps that need to be executed on resource type m, and τ ($0 \leq \tau < 1$) is the safety parameter that determines the maximum total workload allowed and takes into consideration the processing times uncertainty. This parameter τ is computed from the construction data set as follows:

$$\tau = 1 - \frac{1}{n_{JS}^{cons} \times n_{repl}} \cdot \sum_{J=1}^{n_{JS}^{cons}} \sum_{r=1}^{n_{repl}} \frac{\sum_{j=1}^{n_J} \sum_{i=1}^{s_j} p_{ij}}{N \cdot C_{\max,r}(S_J)} \tag{F.3}$$

where for each job set J, $C_{\max,r}(S_J)$ denotes the realized (ex post) makespan corresponding to the r's replication. We obtained $\tau = 0.62$.

Summary

The subject of this thesis is customer order acceptance in batch process industries. Batch process industries produce a large variety of products that follow different routings through a production department. Differences exist between products with regard to the number and duration of the required processing steps. In that respect, they resemble traditional job shops. However, an important difference with traditional job shops is the existence of no-wait restrictions between consecutive processing steps. These no-wait restrictions are caused by unstable intermediate products.

The demand volume for individual products is low and highly variable. As a result, the capacity requirements and the mix of jobs vary considerably over time. Also, different resources may become the bottleneck at different moments in time. One of the main difficulties in production planning of batch process industries is to determine the part of the available capacity that can be used effectively for production. Due to high capital investments in resources and a need for a highly skilled labor force, the available capacity is generally fixed on the medium term. Therefore, if total demand exceeds the available capacity, approaches such as capacity expansion or deploying overtime are not feasible for this type of industry. Hence, it becomes necessary to smooth the capacity requirements to meet the available capacity. This may be done by carefully assessing which orders to accept and how to allocate the available capacity to the accepted orders. The main objective of this thesis is to contribute to the development of models to support on-line order acceptance decisions in complex environments such as batch process industries.

Order acceptance policies discussed in the literature are either based on aggregate information or on detailed scheduling information. Most papers on order acceptance consider the single resource case with deterministic processing times. The production situation in batch process industries is, however, considerably more complex. Moreover, in most real life production situations, processing times are uncertain at the moment the acceptance/rejection decision has to be made. In this thesis, we extend previous work by developing order acceptance policies that incorporate processing time uncertainty.

We model a make-to-order environment comprised of a single batch chemical production department with multiple resources and limited capacity. We address the order acceptance problem in a hierarchical planning and scheduling framework consisting of two levels. At the first level (the order acceptance level) there is the planner who

accepts or rejects orders that are requested by the market for delivery at the end of a specific planning period. The order acceptance decision is an on-line decision and is based on the availability of sufficient capacity to complete the order before its exogenously determined due date. At the second level (the resource allocation level) there is the scheduler who allocates the processing steps of a job (the accepted order) to specific resources and determines the exact sequence and timing of the (planned) execution of the processing steps on the resources. The objective of the order acceptance level is to determine order sets for each planning period that are achievable and realize high capacity utilization. An order set is "achievable" if, at the second level, a schedule can be constructed in which all orders are completed before the end of the planning period, i.e. their due date.

When developing order acceptance policies, it is essential to know whether an order set is likely to be completed within the allocated planning period. In other words, the models used to support order acceptance decisions should accurately estimate the maximum completion time (or makespan) of a particular set of orders. The maximum completion time of order sets is influenced by the workload and the order mix. We identify six aggregate characteristics of the order set that influence the makespan. These characteristics are: (1) workload balance of resource types, (2) average number of processing steps per job in the job set, (3) average overlap of processing steps in the job set, (4) expected processing times variation in the job set, (5) number of jobs in the job set, (6) squared coefficient of variation of the processing times. We use these characteristics, in addition to the workload, to estimate the makespan of a given order set. With the aim to achieve a target service level for the order set (defined as the probability of on-time order set completion), say α, we use regression analysis to develop models that provide an α-reliable estimate for the makespan of the order set. Our order acceptance policy, called the *regression policy*, then accepts orders as long as this estimate does not exceed the length of the planning period.

In this thesis, we compare this regression policy with another method commonly used in the literature and in practice. This policy, called the *scheduling policy*, accepts orders based on a detailed schedule that is constructed every time an order arrives. The schedule is constructed with a simulated annealing algorithm. Given that we consider a stochastic environment, a slack is to be added to account for the processing time uncertainty. The slack is determined by estimating the probability distribution function of the makespan increase due to stochastic processing times. This slack is a static parameter that takes into consideration the level of processing time uncertainty, but it does not explicitly consider other order set characteristics.

We test the two policies under a wide range of experimental conditions (scenarios) characterized by variations in order arrival rate, order mix variety, shop balance, and level of uncertainty in the processing times. Our simulation results show that, for scenarios with low order mix variety, the scheduling policy performs better. More precisely, this policy determines job sets that result in higher capacity utilization and a delivery performance that comes closer to the target. This policy, however, is time consuming as it accepts orders based on a detailed schedule constructed every time an order arrives. In scenarios with high order mix variety, while the scheduling policy maintains high capacity utilization values, the delivery performance deteriorates. This

is especially evident for scenarios with high arrival rate. In these cases, the scheduling policy tends to selectively accept orders that "fit in" with the orders already accepted. The result is a tight schedule that maximizes resource utilization. This selectiveness, however, is detrimental to the delivery reliability. This weakness is due to the fact that by being selective, the scheduling policy significantly changes the mix of jobs in the accepted job sets, compared with the job sets on which the slack is originally determined. Consequently, this slack is not sufficient to cope with the uncertainty in the processing times. In these scenarios, however, the regression policy manages to determine job sets that result in a delivery performance closer to the target and obtains much smaller job set tardiness than the scheduling policy.

We conclude that detailed scheduling information at the order acceptance level is valuable. However, our analysis shows that adding a fixed slack to cope with the effects of processing time uncertainty is not sufficient to maintain high levels of delivery performance. Rather, the ability to estimate the slack accurately must be looked at in conjunction with the characteristics of the job set. We therefore investigate the statistical relationship between the makespan increase due to stochastic processing times and the order set characteristics used by the regression policy. Simulation experiments show that the required slack can be accurately estimated by using these aggregate order set characteristics.

This insight leads us to develop a new order acceptance policy, called the *hybrid policy*. This policy combines detailed scheduling and regression models for the slack estimation. By using regression analysis we dynamically adjust the size of the slack as the job mix in the job set changes. We investigate the performance of this policy for a wide range of customer order and production system scenarios. We also compare it with the performance of the scheduling policy. Our results show that the hybrid policy may be successfully used to determine job sets that result in a delivery performance very close to the target, without loosing the benefits of selectivity on utilization. This is an important result as we provide an order acceptance policy that, by correcting for the weakness of the scheduling policy, allows jobs to be accepted in a manner that provides control over delivery performance.

The order acceptance policies we developed have been tested extensively in various experimental settings using computer simulation. This basically allows a virtually unlimited variety of shops and job sets to be explored to estimate the coefficients of the regression models. Application of such models in a real life situation assumes that sufficient historical data (regarding customer orders and the production system) is available to estimate these coefficients with acceptable accuracy. However, in real life, there may be only a limited amount of historical data available; this may not always be sufficient to produce good regression estimates. We therefore investigate to what extent limited data impacts the performance of an acceptance procedure that uses a regression model. We investigate only the regression policy and the scheduling policy under the limited data case. Our simulation study shows that, with respect to the ability of effectively meeting the customer requirements, the models developed with limited data realize the poorest performance. These results lead to the conclusion that application of regression models to support order acceptance decisions in real life situations may be jeopardized if limited historical data is available. To overcome this

problem, we develop a procedure based on the bootstrap principle. The bootstrap is a computer-based method for measuring the accuracy of statistical estimates. It implies re-sampling - with replacement - of a given (limited) sample of i.i.d. observations. In this thesis, we use such re-sampling to generate additional data (namely order sets) with the right mixture of variety across the order sets and characteristics similar to the observed historical data. We again assess the performance of our bootstrap method by means of simulation experiments. The results show that the bootstrap method gives significant performance improvements for both the regression policy and the scheduling policy.

The findings of this thesis increase our understanding of how statistics, and in particular linear regression models, can be useful in operational planning decisions. Statistical models have an advantage over queuing models, since the former can be applied in settings where the complexity is more than can be captured by the latter. Furthermore, since we show that only a limited amount of data is needed, statistical models can be used in settings that change rather frequently. Combining statistical models with detailed scheduling has an advantage over classical detailed scheduling models, since we demonstrated that the former can better capture the effects of processing time uncertainty than the latter, which would typically address this problem by adding an average slack. We expect that the insight developed in this thesis will assist further developing of applications of regression analysis in operations planning, for instance in standard job shops with constraints on intermediate storage.

Samenvatting

Het onderwerp van dit proefschrift is klantorderacceptatie in de batchprocesindustrie. De batchprocesindustrie produceert een grote variëteit aan producten die verschillende routes door een productieafdeling volgen. Produkten verschillen in aantal en tijdsduur van de benodigde bewerkingen. In dat opzicht bestaat er een grote overeenkomst met klassieke job shops. Echter, een belangrijk verschil met de klassieke job shops is de aanwezigheid van wachttijdrestricties tussen opeenvolgende bewerkingen. Deze wachttijdrestricties worden veroorzaakt door instabiele halffabrikaten, waardoor het in bepaalde situaties noodzakelijk is om bewerkingen direct na elkaar uit te voeren.

De vraag naar individuele producten is laag en varieert sterk in de tijd. Hierdoor varieert de capaciteitsbehoefte en de mix van productie-orders sterk. Bovendien is niet altijd hetzelfde productiemiddel de bottleneck. Een van de grote problemen bij de productieplanning van productie-afdelingen in de batchprocesindustrie is het bepalen welk deel van de aanwezige productie-capaciteit kan worden benut voor productie. Vanwege de hoge investeringskosten in productiemiddelen en het hoge opleidingsniveau van het personeel is de beschikbare capaciteit al vastgelegd op de middellange termijn. Daarom zullen maatregelen zoals capaciteitsuitbreiding van productiemiddelen en van personeel wanneer de totale vraag de beschikbare capaciteit overschrijdt niet haalbaar zijn voor dit type industrie. Dus is het noodzakelijk om de gevraagde capaciteit aan te passen aan de beschikbare capaciteit. Dit kan bereikt worden door zorgvuldig te beoordelen welke orders er geaccepteerd kunnen worden in een tijdsperiode en hoe de beschikbare capaciteit moet worden ingezet voor de geaccepteerde orders. Het doel van dit proefschrift is om bij te dragen aan de kennis voor de ontwikkeling van modellen die on-line de klantorderacceptatiebeslissingen in dergelijke complexe productieomgevingen ondersteunen.

Orderacceptatiemethoden die in de literatuur worden besproken zijn gebaseerd op aggregaatinformatie of op detailplanningsinformatie. De meeste artikelen over orderacceptatie gaan over situaties met slechts één productiemiddel met deterministische bewerkingstijden. De productiesituatie in de batchprocesindustrie is echter complexer. Voorts zijn in de meeste werkelijke situaties de bewerkingstijden niet met zekerheid bekend op het moment dat de acceptatie/weigerbeslissing genomen moet worden. In dit proefschrift breiden we het eerdere werk uit door orderacceptatiemethoden te ontwikkelen die rekening houden met onzekerheid in bewerkingstijden.

We modelleren een make-to-order omgeving die bestaat uit een chemische produc-

tieafdeling die in individuele batches produceert met meerdere productiemiddelen en een beperkte capaciteit. We positioneren het orderacceptatieprobleem in een hiërarchisch planning- en schedulingmodel bestaande uit twee niveaus. Op het eerste niveau (het orderacceptatieniveau) accepteert of weigert een planner orders die geleverd moeten worden aan het einde van de volgende planningperiode. De orderacceptatiebeslissing is een on-line beslissing die gebaseerd is op de beschikbaarheid van voldoende capaciteit om de order, te samen met de reeds eerder geaccepterde orders, af te ronden binnen de gestelde termijn. Op het tweede niveau (het detailplanningsniveau waar de individuele batches worden toegewezen aan de productiemiddelen) wijst een detailplanner elke processtap van een geaccepteerde order toe aan specifieke productiemiddelen en bepaalt hij de exacte volgorde en tijden van de geplande uitvoering. Het doel van het orderacceptatieniveau is om voor elke planningsperiode verzamelingen van orders samen te stellen die haalbaar zijn en die een hoge bezettingsgraad kunnen realiseren. Een verzameling van orders is "haalbaar" wanneer op het tweede niveau een detailplan geconstrueerd kan worden waarin alle orders afgerond zijn voor het einde van de planningperiode, de leverdatum.

Bij het ontwikkelen van orderacceptatiemethoden is het dus essentieel om te weten of van een verzameling orders te verwachten is of deze uitgevoerd kan worden binnen de daarvoor gereserveerde periode. Met andere woorden, de modellen die gebruikt worden om de orderacceptatiebeslissingen te ondersteunen moeten met een zekere betrouwbaarheid de verwerkingstijd van een bepaalde verzameling van productieorders kunnen schatten. De verwerkingstijd van een verzamelings van orders wordt beïnvloed door de werklast en de ordermix. We identificeren zes aggregaatkarakteristieken van de verzameling orders die de verwerkingstijd benvloeden. Deze karakteristieken zijn: (1) de werklastbalans over de typen productiemiddelen, (2) het gemiddelde aantal bewerkingen per productieorder in de verzameling orders, (3) de gemiddelde overlap tussen bewerkingen van productieorders, (4) de verwachte bewerkingstijdvariatie in de verzameling orders, (5) het aantal productieorders in de verzameling, en (6) het kwadraat van de variatiecoëfficiënt van de werkelijke bewerkingstijden. We gebruiken deze karakteristieken, in aanvulling op de werklast, om de makespan (het tijdstip waarop de allerlaatste order is verwerkt) van een gegeven verzameling orders te schatten. Ons doel is om een bepaalde leverbetrouwbaarheid, aangeduid met α, te bereiken voor een geaccepteerde verzameling orders (gedefinieerd als de waarschijnlijkheid om een verzameling orders op tijd af te ronden). We gebruiken regressie-analyse om modellen te ontwikkelen die voorzien in een een α-betrouwbaarheid schatting voor de werkelijke makespan van de verzameling orders. Gebaseerd op deze schatting worden orders geaccepteerd zolang de schatting van de makespan niet de lengte van de planningperiode overschrijdt.

In dit proefschrift vergelijken we deze regressiemethode met een methode die karakteristiek is voor methoden die vaak gebruikt worden in de literatuur en de praktijk. Deze methode, die we aan zullen duiden als de detailplanningsmethode, accepteert orders gebaseerd op een detailplan dat elke keer wordt geconstrueerd met een simulated annealing algoritme. Gegeven dat we rekening houden met een stochastische omgeving, moet er speling worden toegevoegd om rekening te houden met onzekerheid in de bewerkingstijden. De speling is gebaseerd op een schatting van de kansverdelingsfunctie van de verlenging van de makespan die wordt veroorzaakt door de stochastische be-

werkingstijden. Deze speling is een statische parameter die rekening houdt met de onzekerheid in de bewerkingstijden maar niet expliciet met andere karakteristieken van de verzameling orders.

De twee methoden zijn door middel van simulatie intensief getoetst voor een groot aantal scenario's, gekarakteriseerd door variaties in de vraag/capaciteitsratio, de ordermixvariëteit, de werklastbalans en het niveau van onzekerheid in de bewerkingstijden. Onze simulatieresultaten laten zien dat in situaties met een lage variëteit in de ordermix, de detailsplanningsmethode het beste presteert. Deze methode stelt verzamelingen van orders samen die resulteren in een hogere bezettingsgraad en een betere leverprestatie. Echter, de detailplanningsmethode is rekenintensief omdat het orders accepteert gebaseerd op een detailplan dat telkens opnieuw wordt gemaakt wanneer een order aankomt. In het scenario met een hoge variëteit in de ordermix houdt de detailplanningsmethode de bezettingsgraad op hetzelfde hoge niveau, terwijl de leverprestatie vermindert. Dit effect treedt het sterkst op bij scenario's met een hoge capaciteitbehoefte. In deze scenario's heeft de detailplanningsmethode de neiging om selectief orders te accepteren die goed passen bij de al eerder geaccepteerde orders. Het resultaat is een krap detailplan dat productiemiddelen maximaal benut. Deze selectiviteit is echter schadelijk voor de leverprestatie. Deze schadelijkheid wordt veroorzaakt vanwege het feit dat door selectief te zijn, de detailplanningsmethode de mix van orders significant verandert ten opzichte van de verzamelingen van orders waarvoor de speling oorspronkelijk was bepaald. Daarom is de aldus bepaalde speling niet geschikt. In deze scenario's leidt het gebruik van de regressiemethode daarentegen tot orderverzamelingen die resulteren in een leverprestatie die dicht bij het van te voren gespecificeerde doel komen en tot een veel kleinere orderverzamelings*tardiness* dan de detailplanningsmethode.

De resultaten van onze simulatiestudie laten zien dat de detailplanningsinformatie bij de orderacceptatiebeslissing zeer nuttig is. Echter, onze analyse laat zien dat het gebruiken van een vaste speling om met de effecten van onzekerheid in de bewerkingstijden om te gaan, niet voldoende is om een hoge leverprestatie te bereiken. Blijkbaar moet bij het schatten van de speling nauwkeurig rekening worden gehouden met de karakteristieken van die verzameling orders. Daarom onderzoeken we de statistische relatie tussen de toename in de makespan (veroorzaakt door de stochastische bewerkingstijden) en de karakteristieken van de geaccepteerde verzameling orders. Simulatie-experimenten laten zien dat de benodigde speling nauwkeurig kan worden bepaald door deze te bereken op aggregaatkarakteristieken van de verzameling orders.

Dit inzicht leidt tot de ontwikkeling van een nieuwe orderacceptatiemethode, de hybride methode. Deze hybride methode combineert detailplanning en regressiemodellen om een schatting te maken van de benodigde speling. Door regressie-analyse te gebruiken, passen we dynamisch de grootte van de speling aan wanneer de order mix in de verzameling orders verandert. De prestatie van deze hybride methode is onderzocht voor een breed scala aan klantorder- en productiesysteemscenario's en vergeleken met de prestatie van de detailplanningsmethode. Onze resultaten laten zien dat de hybride methode zeer succesvol is in het samenstellen van verzamelingen orders die resulteren in een hoge leverprestatie, zonder de voordelen te verliezen van selectiviteit tijdens het gebruik. Dit is een belangrijk resultaat, omdat we zo een or-

deracceptatiemethode hebben ontwikkeld die, door een van de zwakke eigenschappen van de detailplanningsmethode te corrigeren, orders accepteert op een manier die de leverprestatie beheerst.

De orderacceptatiemethoden die we ontwikkeld hebben zijn uitgebreid getoetst onder diverse experimentele situaties door middel van computersimulatie. Dit maakt het mogelijk een zeer grote variëteit van productie-afdelingen en verzamelingen van orders te gebruiken voor het schatten van de coëfficiënten van de regressiemodellen. Toepassing van dergelijke modellen in de werkelijkheid gaat er vanuit dat voldoende historische informatie beschikbaar is om met een voldoende nauwkeurigheid deze coëfficiënten te schatten. Echter, in werkelijkheid kan het zijn dat er slechts een beperkte hoeveelheid relevante historische informatie beschikbaar is; dit kan onvoldoende zijn om goede regressieschattingen te produceren. We onderzoeken daarom in hoeverre beperkte beschikbaarheid van historische informatie invloed heeft op de prestatie van een orderacceptatiemethode gebaseerd op een regressiemodel. We beperkten ons tot alleen de regressiemethode en de detailplanningsmethode onder de beperkte gegevensbeschikbaarheid. We ontdekken dat, wat betreft het vermogen om effectief aan de klanteisen tegemoet te komen, de modellen die ontwikkeld zijn voor het geval van een beperkte gegevensbeschikbaarheid, resulteren in de slechtste prestatie. Deze resultaten leiden tot de conclusie dat de toepassing van regressiemodellen om orderacceptatiebeslissingen in realistische situaties te ondersteunen, in gevaar kan worden gebracht wanneer slechts beperkte historische informatie beschikbaar is. Om dit probleem aan te pakken, ontwikkelen we een procedure op basis van het bootstrap-principe. Het bootstrap-principe houdt in dat met een gegeven beperkte hoeveelheid gegevens nieuwe gegevensverzamelingen worden gevormd door trekking met teruglegging. In dit proefschrift gebruiken we re-sampling om additionele informatie (verzamelingen van productieorders) te genereren met de juiste mix van variteit over de orderverzamelingen en met de karakteristieken vergelijkbaar met die van de waargenomen historische informatie. We onderzoeken de prestatie van onze bootstrapmethode door middel van simulatie-experimenten. De resultaten laten zien dat door onze bootstrapmethode te gebruiken, significante prestatieverbeteringen te behalen zijn voor zowel de regressiemethode als de detailplanningsmethode.

De bevindingen van dit proefschrift dragen bij aan onze kennis over hoe statistische methoden, in het bijzonder lineaire regressiemodellen, gebruikt kunnen worden in operationele planningsbeslissingen. Statistische modellen hebben een voordeel ten opzichte van analytische (wachtrij, detailplanning) modellen aangezien statistische modellen toegepast kunnen worden in omgevingen waar de complexiteit zo groot is dat het gebruiken van analytische methoden alléén niet werkt. We laten bovendien zien dat slechts een beperkte hoeveelheid historische informatie nodig is; statistische modellen kunnen dus ook gebruikt worden in omgevingen die vrij regelmatig veranderen. Statistische modellen blijken goed gecombineerd te kunnen worden met analytische methoden (detailplanning). We verwachten dat de inzichten die ontwikkeld zijn in dit proefschrift kunnen leiden tot de verdere ontwikkeling bij het gebruik van regressiemethoden in operationele planning, bijvoorbeeld in standaard jobshops met tussenopslagbeperkingen.

References

Aarts, E.H.L. and Lenstra, J.K. (1997). *Local search in combinatorial optimization.* Wiley, Chichester.

Adam, N.R., Betrand, J.W.M., Morehead, D.C. and Surkis, J. (1993). Due date assignment procedures with dynamically updated coefficients for multi-level assembly job shops. *European Journal of Operational Research*, **68**, 212–227.

Akkan, C. (1997). Finite-capacity scheduling-based planning for revenue-based capacity management. *European Journal of Operational Research*, **100**, 170–179.

Baker, K.R. and Bertrand, J.W.M. (1981a). A comparison of due-date selection rules. *AIIE Transactios*, **13**(2), 123–131.

Baker, K.R. and Bertrand, J.W.M. (1981b). An investigation of due-date assignment rules with constrained tightness. *Journal of Operations Management*, **1**(3), 109–120.

Balakrishnan, N., Sridharan, V. and Patterson, J.W. (1996). Rationing capacity between two product classes. *Decision Science*, **27**(2), 185–214.

Barut, M. and Sridharan, V. (2004). Design and evaluation of a dynamic capacity apportionment procedure. *European Journal of Operational Research*, **155**(1), 112–133.

Battersby, A. (1967). *Network Analysis for Planning and Scheduling.* Macmillan, New York.

Bernal-Haro, L., Azzaro-Pantel, C., Pibouleau, L. and Domenech, S. (2002). Multiobjective batch plant design: a two-stage methodology. 2. Developement of a genetic algorithm and result analysis. *Industrial and Engineering Chemistry Research*, **41**, 5743–5758.

Bertrand, J.W.M. (1983a). The effect of workload dependent due dates on job shop performance. *Management Science*, **29**(7), 799–816.

Bertrand, J.W.M. (1983b). The use of workload information to control job lateness in controlled and uncontrolled release production systems. *Journal of Operations Management*, **3**(2), 79–92.

Bertrand, J.W.M. and van de Wakker, A.M. (2002). An investigation of order release and flow time allowance policies for assembly job shops. *Production Planning and Control*, **13**(7), 639–648.

Bertrand, J.W.M., Wortmann, J.C. and Wijngaard, J. (1990). *Production control: a structural and design oriented approach.* Elsevier, Amsterdam.

Bok, J.K. and Park, S. (1998). Continuous-time modeling for short-term scheduling of multipurpose pipeless plants. *Industrial and Engineering Chemistry Research*, **37**, 3652–3659.

Buzacott, J.A. and Shantikumar, J.G. (1993). *Stochastic models of manufacturing systems.* Prentice Hall, Englewood Cliffs, NY.

Carlier, J. (1987). Scheduling jobs with release dates and tails on identical machines to minimize the makespan. *European Journal of Operational Research*, **29**, 298–306.

Chang, F.R. (1997). A study of factors affecting due-date predictability in a simulated dynamic job shop. *Journal of Manufacturing Systems*, **13**(6), 393–400.

Cheng, T.C.E. (1985). Analysis of job flowtime in a job shop. *Journal of Operational Research Society*, **36**(3), 225–230.

Cheng, T.C.E. (1986). Due date determination for a single machine shop with SPT dispatching. *Engineering Costs and Production Economics*, **10**, 35–41.

Cheng, T.C.E. and Gupta, M. C. (1989). Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research*, **38**, 156–166.

Conway, R.W. (1965). Priority distpatching and job lateness in a job shop. *Journal of Industrial Engineering*, **16**(2), 228–237.

Daniels, R.W. and Kouvelis, P. (1995). Robust scheduling to hedge against processing times uncertainty in single stage production. *Management Science*, **41**, 363–376.

de Kok, A.G. and Fransoo, J.C. (2003). Planning supply chain operations: definition and comparison of planning concepts. In de Kok, A.G. and Graves, S.C. (eds), *Handbooks in operations research and management science, Vol. 11, Supply chain management: design, coordination and operation.* North-Holland Publishing Company, Amsterdam.

Dimitriadis, A.D., Shah, N. and Pantelides, C.C. (1997). RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Computers and Chemical Engineering*, **21**(Suppl.), S1061–S1066.

Dodin, B.M. (1985). Bounding the project completion time distribution in PERT Networks. *Operations Research*, **24**(4), 862–881.

Dooley, K.J. and Mahmoodi, F. (1992). Identification of robust scheduling heuristics: application of Taguchi methods in simulation studies. *Computers and Industrial Engineering*, **22**, 359–368.

Duenyas, I. (1995). Single facility due date setting with multiple customer classes. *Management Science*, **41**, 608–619.

Duenyas, I. and Hopp, W.C. (1995). Quoting customer lead times. *Management Science*, **41**, 43–57.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, **7**, 1–27.

Efron, B. and Tibshirani, R.J. (1993). *An introduction to the bootstrap*. Chapman & Hall, New York.

Eilon, S. and Chowdhury, I.G. (1976). Due dates in job shop scheduling. *International Journal of Production Research*, **14**(2), 223–237.

Enns, S.T. (1993). Job shop flowtime prediction and tardiness control using queueing analysis. *International Journal of Production Research*, **31**(8), 2045–2057.

Enns, S.T. (1994). Job shop leadtime requirements under conditions of controlled delivery performance. *European Journal of Operational Research*, **77**, 429–439.

Enns, S.T. (1995). A dynamic forecasting model for job shop flowtime prediction and tardiness control. *International Journal of Production Research*, **33**(5), 1295–1312.

Ford, F.N., Bradbard, D.A., Ledbetter, W.N. and Cox, J.F. (1987). Use of operations research in production management. *Production and Inventory Management*, **3rd Qtr.**, 59–62.

Fransoo, J.C. (1993). *Production control and demand management in capacitated flow process industries*. Ph.D. thesis, Eindhoven University of Technology.

Fransoo, J.C. and Rutten, W.G.M.M. (1994). A typology of production situations in process industries. *International Journal of Operations and Production Management*, **14**(12), 47–57.

Fransoo, J.C., de Kok, A.G. and Paulli, J. (1994). *Makespan estimation in flexible manufacturing systems*. Tech. rept. 94-14. TUE/BDK/LBS, Eindhoven University of Technology, Eindhoven, The Netherlands.

Fry, T.D., Philipoom, P.R. and Markland, R.E. (1989). Due date assignment in a multistage job shop. *IIE Transactions*, **21**(2), 153–161.

Gee, E.S. and Smith, C.H. (1993). Selecting allowance policies for improved job shop performance. *International Journal of Production Research*, **31**(8), 1839–1852.

Ghosh, J.B. (1997). Job selection in a heavily loaded shop. *Computers and Operations Research*, **24**(2), 141–145.

Gordon, V.S., Proth, J.M. and Chu, C. (2002). Due date assignment and scheduling: SLK, TWK and other due date assignment models. *Production Planning and Control*, **12**(2), 117–132.

Grunow, M., Günther, H.O. and Lehmann, M. (2002). Campaign planning for multi-stage batch processes in the chemical industry. *OR Spectrum*, **24**(3), 281–314.

Guerrero, H.H. and Kern, G.M. (1988). How to more effectivelly accept and refuse orders. *Production and Inventory Management*, **29**(4), 59–62.

Hahn, G.J. and Meeker, W.Q. (1991). *Statistical intervals. A guide for practitioners.* John Wiley and Sons, New York.

Hax, A.C., Majluf, N.S. and Pendrock, M. (1980). Diagnostic analysis of a production and distribution system. *Management Science*, **26**(9), 871–889.

Hayes, R.H. and Wheelwright, S.C. (1979). Link manufacturing process and product life cycles. *Harvard Business Review*, **57**(1), 133–140.

Hollander, M. and Wolfe, D.A. (1999). *Nonparametric statistical methods.* John Wiley and Sons, New York.

Ierapetritou, M.G. and Floudas, C.A. (1998). Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Industrial and Engineering Chemistry Research*, **37**, 4341–4359.

Ishii, N. and Muraki, M. (1996). A process-variability-based online scheduling system in multiproduct batch process. *Computers and Chemical Engineering*, **20**(2), 217–234.

Ivanescu, C.V., Fransoo, J.C. and Bertrand, J.W.M. (2002). Makespan estimation and order acceptance in batch process industries when processing times are uncertain. *OR Spectrum*, **24**(4), 467–495.

Ivanescu, C.V., Fransoo, J.C. and Bertrand, J.W.M. (2003a). Planning and scheduling in the process industry. In Günther, H.O. and van Beek, P. (eds), *Advanced planning and scheduling solutions in process industries.* Springer, Berlin.

Ivanescu, V.C., Fransoo, J.C. and Bertrand, J.W.M. (2003b). *On the selectivity of order acceptance procedures in batch process industries.* BETA WP 99.

Ivanescu, V.C., Bertrand, J.W.M., Fransoo, J.C. and Kleijnen, J.P.C. (2004a). *Bootstrapping to solve the limited data problem in production control: an application to batch process industries.* under review.

Ivanescu, V.C., Fransoo, J.C. and Bertrand, J.W.M. (2004b). *A hybrid policy to correct for the selectivity of order acceptance procedures in batch process industries.* under review.

Kallrath, J. (2003). Planning and scheduling in the process industry. In Günther, H.O. and van Beek, P. (eds), *Advanced planning and scheduling solutions in process industries.* Springer, Berlin.

Kern, G.M. and Guerrero, H.H. (1990). A conceptual model for demand management in the assemble-to-order environment. *Journal of Operations Management*, **9**, 65–84.

Kim, S.B., Lee, H.K., Lee, I.B., E.S.Lee and B.Lee. (2000). Scheduling of non-sequential multipurpose batch processes under finite intermediate storage policy. *Computers and Chemical Engineering*, **24**(2-7), 1603–1610.

Kleijnen, J.P.C. (2004). An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research*, **xxx**, xxx–xxx. Article in press.

Kondili, E., Pantelides, C.C. and Sargent, R.W.H. (1993). A general algorithm for short-term scheduling of batch operations-I. MILP formulation. *Computers and Chemical Engineering*, **17**(2), 211–227.

Ku, H. and Karimi, I. (1991). An evaluation of simulated annealing for batch process scheduling. *Industrial and Engineering Chemistry Research*, **30**, 163–169.

Lane, M.S., Mansour, A.H. and Harpell, J.L. (1993). Operations research techniques: a longitudinal update 1973-1988. *Interfaces*, **23**, 63–68.

Law, A.M. and Kelton, W.D. (2000). *Simulation modeling and analysis*. 3rd edn. McGraw-Hill, New York.

Lawrence, S.R. (1995). Estimating flowtimes and setting due dates in complex production systems. *IIE Transactions*, **27**, 657–668.

Lenstra, J.K., Kan, A.H.G. Rinnooy and Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, **1**, 343–362.

Leon, V.J., Wu, S.D. and Storer, R.H. (1994). Robustness measures and robust scheduling for job shops. *IIE Transactions*, **26**, 32–43.

Lewis, H.F. and Slotnick, S.A. (2002). Multi-period job selection: planning work loads to maximize profit. *Computers and Operations Research*, **29**(8), 1081–1098.

Maravelias, C.T. and Grossmann, I. (2003). New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Industrial and Engineering Chemistry Research*, **42**, 3056–3074.

Mauderli, A. and Rippin, D.W.T. (1979). Production planning and scheduling for multipurpose batch chemical plants. *Computers and Chemical Engineering*, **3**(1-4), 199–206.

Maxwell, S.E. and Delaney, H.D. (1990). *Designing experiments and analyzing data. A model comparison perspective*. Wadsworth, Inc., Pacific Grove, California.

Miller, D.M. (1984). Reducing transformation bias in curve fitting. *The American Statistician*, **38**, 124–126.

Miyazaki, S. (1981). Combined scheduling system for reducing job tardiness in a job shop. *International Journal of Production Research*, **19**(2), 201–211.

Montgomery, D.C. and Peck, E.A. (1992). *Introduction to linear regression analysis*. Wiley, New York.

Papageorgiu, L.G. and Pantelides, C.C. (1993). A hierarchical approach for campaign planning of multipurpose batch plants. *Computers and Chemical Engineering*, **17**(Suppl.), S27–S32.

Papageorgiu, L.G. and Pantelides, C.C. (1996). Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. Mathematical formulation. *Industrial and Engineering Chemistry Research*, **35**, 488–509.

Pinedo, M. (1995). *Scheduling: theory, algorithms and systems.* Prentice Hall, New Jersey.

Pinto, J.M. and Grossmann, I.E. (1995). A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial and Engineering Chemistry Research*, **34**, 3037–3051.

Portougal, V. and Trietsch, D. (2001). Stochastic scheduling with optimal customer service. *Journal of the Operational Research Society*, **52**, 226–233.

Raaymakers, W.H.M. (1999). *Order acceptance and capacity loading in batch proces industries.* Ph.D. thesis, Eindhoven University of Technology.

Raaymakers, W.H.M. and Hoogeveen, J.A. (2000). Scheduling no-wait job shops by simulated annealing. *European Journal of Operational Research*, **126**, 131–151.

Raaymakers, W.H.M., Bertrand, J.W.M. and Fransoo, J.C. (2000a). The performance of workload rules for order acceptance in batch chemical manufacturing. *Journal of Intelligent Manufacturing*, **11**, 217–228.

Raaymakers, W.H.M., Bertrand, J.W.M. and Fransoo, J.C. (2000b). Using aggregate estimation models for order acceptance in a descentralized production control for batch chemical manufacturing. *IIE Transactions*, **32**, 989–998.

Ragatz, G.L. and Mabert, V.A. (1984). A simulation analysis of due date assignment rules. *Journal of Operations Management*, **5**(1), 27–39.

Reklaitis, G.V. (1996). Overview of scheduling and planning of batch process operations. In Reklaitis, G.V., Sunol, A.K., Rippin, D.W.T. and Hortacsu, O. (eds), *Batch processing systems engineering.* Springer, Berlin.

Schilling, G. and Pantelides, C.C. (1999). Optimal periodic scheduling of multipurpose plants. *Computers and Chemical Engineering*, **23**(4-5), 635–655.

Schneeweiß, Ch. (1995). Hierarchical structures in organisations: a conceptual framework. *European Journal of Operational Research*, **86**, 4–31.

Schneeweiß, Ch. (1999). *Hierarchies in distributed decision making.* Springer, Berlin.

Shah, N. and Pantelides, C.C. (1991). Optimal long-term campaign planning and design of batch operations. *Industrial and Engineering Chemistry Research*, **30**, 2308–2321.

Shah, N., Pantelides, C.C. and Sargent, R.W.N. (1993). A general algorithm for short-term scheduling of batch operations - II. Computational issues. *Computers Chemical Engineering*, **17**(2), 229–244.

Slotnick, S.A. and Morton, T.E. (1996). Selecting jobs for a heavily loaded shop with lateness penalties. *Computers and Operations Research*, **23**(2), 131–140.

Subrahmanyan, S., Bassett, M.H., Pekny, J.F. and Reklaitis, G.V. (1995). Issues in solving large scale planning, design and scheduling problems in batch chemical plants. *Computers and Chemical Engineering*, **19**(Suppl.), S577–S582.

Subrahmanyan, S., Pekny, J.F. and Reklaitis, G.V. (1996). Decomposition approaches to batch plant design and planning. *Industrial and Engineering Chemistry Research*, **35**, 1866–1876.

Tavares, L.V. (1999). *Advanced Models for Project Management*. Kluwer Academic Publishers, Boston/Dordrecht/London.

Tavares, L.V., Ferreira, J.A. and Coelho, J.S. (1999). The risk of delay of a project in terms of the morphology of its network. *European Journal of Operational Research*, **119**, 510–537.

Taylor, S.G., Seward, S.M. and Bolander, S.F. (1981). Why the process industries are different. *Production and Inventory Management*, **22**(4), 9–24.

Ten Kate, H.A. (1994). Towards a better understanding of order acceptance. *International Journal of Production Economics*, **37**, 139–152.

van Bael, P. (1999). A study of rescheduling strategies and abstraction levels for a chemical process scheduling problem. *Production Planning and Control*, **10**, 359–364.

van Laarhoven, P.J.M. and Aarts, E.H.L. (1987). *Simulated annealing: theory and applications*. Reidel, Dordrecht, The Netherlands.

Vig, M.M. and Dooley, K.J. (1991). Dynamic rules for due date assignment. *International Journal of Production Research*, **29**(7), 1361–1377.

Vig, M.M. and Dooley, K.J. (1993). Mixing static and dynamic estimates for due date assignment. *Journal of Operations Management*, **11**, 67–79.

Voudouris, T.V. and Grossmann, I.E. (1993). Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory considerations. *Industrial and Engineering Chemistry Research*, **32**, 1962–1980.

Wallace, T.F. (1984). *APICS Dictionary*. 5th ed. edn. American Production and Inventory Control Society, Falls Church, VA.

Wang, J., Yang, J.Q. and Lee, H. (1994). Multicriteria order acceptance decision support in over-demanded job shops: a neural network approach. *Mathematical and Computer Modelling*, **19**(5), 1–19.

Weeks, J.K. (1979). A simulation study of predictible due dates. *Management Science*, **25**(4), 363–373.

Wellons, M.C. and Reklaitis, G.V. (1991). Scheduling of multipurpose batch chemical plants. 2. Multiple product campaign formation and production planning. *Industrial and Engineering Chemistry Research*, **30**, 688–705.

Wera, Y. (2002). *Production planning and raw materials management models in the process industry.* Ph.D. thesis, Université de Liège.

Wester, F.A.W., Wijngaard, J. and Zijm, W.H.M. (1994). Order acceptance strategies in a production-to-order environment with setup times and due-dates. *International Journal of Production Research*, **30**, 1313–1326.

Zhu, X.X. and Majozi, T. (2001). Novel continuous time MILP formulation for multipurpose batch plants. 2. Integrated planning and scheduling. *Industrial and Engineering Chemistry Research*, **40**, 5621–5634.

Curriculum Vitae

Virginia Cristina Ivănescu was born in Bucharest, Romania, on September 23rd, 1974. After completing her pre-university education at the High School "Gh. Lazăr" of Bucharest, she started in the same year to study at the University of Bucharest. She graduated from the Department of Applied Mathematics, of the Faculty of Mathematics.

After graduation, she worked as a statistician at the National Institute of Statistics, Romania, in the Department of Projection and Realization of Structural Business Surveys.

From October 2000, she worked as a trainee research assistant on a project in the Operations Planning, Accounting and Control group of the Faculty of Technology Management, at the Technische Universiteit Eindhoven. The research on applying regression analysis for supporting order acceptance decisions in batch process industries, carried out in the period October 2000 — October 2004, led to this dissertation.

Stellingen

behorende bij het proefschrift

Order acceptance under uncertainty in batch process industries

van

Virginia Cristina Ivănescu

4 oktober 2004

I

By selecting orders under a capacity constraint in order to optimize some objective function, the mix of orders may change in such a way that the model underlying the selection mechanism is no longer valid.
[this thesis, chapter 5]

II

Using the output of deterministic scheduling models - with expected job durations as input - for planning purposes may cause low service level if processing times are stochastic [V. Portougal and D. Trietsch, 2001]. To achieve an appropriate service level, we have to include safety time (or slack) into the schedule. We claim that major gains in delivery performance measure may be obtained by dynamically adjusting the size of the slack as the job mix changes.
[V. Portougal and D. Trietsch, Stochastic scheduling with optimal customer service, Journal of Operational Research Society, 52: 226-233, 2001; this thesis, chapter 6]

III

There exists a trade-off between model robustness and performance. Using a single predictive model that covers all the variants of production situations may lead to a robust order acceptance policy, but the resulting policy may perform poorly in certain regions of the parameters space.
[M.M. Vig and K.J. Dooley, Mixing static and dynamic flowtime estimates for due-date assignment, Journal of Operations Management, 11: 67-79, 1993; this thesis, chapter 6]

IV

The planning and control performance that can be obtained with the use of a regression model depends on the quality of the regression model, which in turn depends on the amount of data available for estimating the model. Many real life production situations are dynamic in terms of characteristics of product, production processes and resource configurations. As a result, only recent data are relevant for the future, which severely limits the amount of data available for developing the model. The lack of data problem may be solved by intelligent application of bootstrapping techniques.
[this thesis, chapter 7]

V

Through the use of enterprise information systems, organizations record nowadays an increasing amount of operational data. This offers increased opportunities for using statistical methods not only for measuring and monitoring purposes but also to model complex causal relationships to further improve the planning and control of production processes.

VI

For research purposes, if one is not willing to "sacrifice details and idiosyncracies for the sake of perceiving the broader picture" all one may hope for is a perfect replica of the real-world, which provides at most the same insights that the real-world itself could provide.
[M.L. Whicker and L. Sigelman, Computer simulation applications, an introduction, Applied Social Research Methods Series, Vol. 25, Sage Publications, Newbury Park, 1991]

VII

When doing research, finding the right question is the first step to finding the right answer.

VIII

All of life is an experiment. The more experiments you do, the better.
[Ralph Waldo Emerson, 1803-1882]

IX

What participating in the Olympic Games means for a sportsman, is publishing in Management Science for an OR/MS researcher.

X

For a successful PhD defense like for winning a fencing bout, all one needs is a strong parry and a fast riposte.

XI

If the sun shines after two cold, rainy days, it is probably Monday.