# Order acceptance in multipurpose batch process industries using the regression policy

Simonis, B.J.

*Award date:*
2006

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Master's thesis

# Order Acceptance in Multipurpose Batch Process Industries Using the Regression Policy

by

B.J. Simonis

Supervisors:
Dr. A. Di Bucchianico
Professor Dr. Ir. J.C. Fransoo

Eindhoven, 15th June 2006

# Preface

This Masters thesis is the result of 9 months of research at the Department of Mathematics and Computer Science, chair of Probability and Statistics, and at the Department of Technology Management, Operations Planning, Accounting and Control group, at the Eindhoven University of Technology (TU/e). With this thesis I will end my study of Industrial and Applied Mathematics, and I will receive the degree of Master of Science.

The research topic of this thesis, Order Acceptance in Multipurpose Batch Process Industries Using the Regression Policy, was introduced to me by Professor Dr. Ir. J.C. Fransoo from the Department of Technology Management and Dr. A. Di Bucchianico from the Department of Mathematics and Computer Science, who became my supervisors on this project.

I would like to take some time to thank everybody who, in one way or another, helped me during my 9 months of research. First of all there are my supervisors. They helped me every time I had a question or problem. And always in the most patient and precise way, in spite of their very busy schedules. I would also like to thank Ir W.I.P.M. Kortsmit and Ing K. Huibers for their help with programming and Dr. C. Ivanescu for her help every time something was not clear to me.

I would love to thank my parents who made it possible for me to begin (and finish) my study. Not only financially, but also for their support and love. I also want to thank my sisters and all my friends, Erik in particular, for their support and love.

# Summary

The subject of this thesis is order acceptance in multipurpose batch process industries using the regression policy. Multipurpose batch processes can be found in speciality chemicals and in the pharmaceutical industry.

Due to the unique characteristics of a multipurpose batch process, especially the waiting time restrictions and the overlapping processing steps, it is difficult and time wasting to schedule jobs and also to find the makespan of a job set. Historically the Simulated Annealing algorithm was used, a very time wasting algorithm. In Ivanescu (2004) a new method is developed. An estimation model is built, using regression analysis, to estimate the makespan of a job set in a multipurpose batch process. Hence, given the job set characteristics (which are the regression variables), the makespan of a job set can be estimated relatively quick. In a part of this thesis we investigated the building and evaluation of this model.

To give a customer a quick answer to whether or not his/her order can be produced within certain time limits an order acceptance function can be defined. In this thesis the regression policy is considered. The regression policy is based on the already constructed estimation model and has a predefined service level.

When using the regression policy it can be observed that the predefined service level is not reached. This problem is called selectivity.

To get a better understanding of the concept of selectivity we simulated the regression policy. By doing so we could investigate and identify the bias, $b(x)$, that occurs.

To solve the problem of not reaching the predefined service level two solutions are handed. Firstly it is possible to give the regression policy a predefined service level that is higher than the service level that we want to reach in the end. Hence, a decrease in service level is not a problem anymore. Secondly it is possible to make a correction for the bias each time the regression policy is used.

We end this thesis with some ideas for further research.

# Contents


**1 Introduction** **11**
- 1.1 Multipurpose batch process . . . 11
  - 1.1.1 Batch process versus flow process . . . 11
  - 1.1.2 Multipurpose versus multiproduct . . . 11
- 1.2 Characteristics of a multipurpose batch process . . . 12
- 1.3 Historical Background . . . 12
- 1.4 Research objective . . . 13
- 1.5 Thesis outline . . . 13

**2 Order Acceptance in Multipurpose Batch Process Industries** **15**
- 2.1 Order acceptance . . . 15
- 2.2 Order arrival/acceptance procedure . . . 15

**3 Building and Validating of the Makespan Estimation Model Using Regression Analysis** **17**
- 3.1 Estimation model formulation . . . 17
- 3.2 Model building . . . 20
  - 3.2.1 Regressors and response variable . . . 20
  - 3.2.2 Data generation . . . 21
  - 3.2.3 Models . . . 22
  - 3.2.4 Multicollinearity . . . 22
  - 3.2.5 Model adequacy checking . . . 23
  - 3.2.6 Overall adequacy of the model . . . 24
  - 3.2.7 Estimating the regression coefficients . . . 24
  - 3.2.8 $F$-test and $t$-test . . . 25
- 3.3 Model validation . . . 25
  - 3.3.1 $ME$, $\sqrt{MSE}$ and $R^2_{pred}$ . . . 25
  - 3.3.2 Estimation errors . . . 25

**4 Data-splitting** **27**
- 4.1 Description of the Duplex-algorithm . . . 28
- 4.2 Using the DUPLEX-algorithm . . . 31

**5 Regression Policy and Selectivity** **33**
- 5.1 Description of the regression policy . . . 33
- 5.2 Random order arrival data . . . 34
- 5.3 Problem: Selectivity . . . 34

5.4 Impact of Selectivity on Performance . . . 34
5.5 Problem approach . . . 35

**6 Mathematically Understanding Selectivity 37**
6.1 Simulation . . . 37
6.1.1 Overview . . . 37
6.1.2 `Mathematica` code . . . 39
6.1.3 Example . . . 39
6.2 Checking the simulation . . . 39
6.2.1 Acceptance=100% . . . 40
6.2.2 Acceptance $< 100\%$ . . . 41
6.3 Description of this bias . . . 45

**7 Possible Solutions 49**
7.1 Adapting the $\alpha$ . . . 49
7.1.1 Approach . . . 49
7.2 Correcting for the bias . . . 50
7.2.1 $b(x) = e^{\frac{\sigma^2}{2}}$ . . . 50
7.2.2 $b(x) < e^{\frac{\sigma^2}{2}}$ . . . 50

**8 Conclusion 53**

**A Probability Distributions 55**

**B Multiple Linear Regression Analysis 59**
B.1 Regression analysis . . . 59
B.1.1 Properties of random vectors . . . 59
B.1.2 Models and estimation . . . 59
B.2 Building the model . . . 61
B.2.1 Choosing initial models . . . 61
B.2.2 Multicollinearity . . . 61
B.2.3 Model adequacy checking . . . 61
B.2.4 Overall adequacy of the model . . . 62
B.2.5 Estimation of $\sigma$ . . . 62
B.2.6 $F$-test and $t$-test . . . 63
B.3 Model validation . . . 63
B.3.1 Mean Prediction Error . . . 64
B.3.2 Square root of the mean square prediction error . . . 64
B.3.3 Percentage of variability explained by the model . . . 64

**C `R` Manual 65**
C.1 Packages . . . 65
C.2 The help function . . . 65
C.3 Reading data . . . 65
C.4 Regression analysis in `R` . . . 66
C.5 Working with matrices in `R` . . . 66
C.6 `R` code . . . 66

C.7 Building the model . . . . . 67

**D `Mathematica` Code** **69**
D.1 Data-splitting: DUPLEX . . . . . 69
D.1.1 Explanation of the code . . . . . 70
D.2 Selectivity in the regression policy . . . . . 70
D.2.1 Explanation of the code . . . . . 71

**Bibliography** **75**

# Chapter 1

# Introduction

The research of this thesis is situated around order acceptance in multipurpose batch process industries. Therefore we want to give some information about this sort of industry in this chapter. We elaborate on how a multipurpose batch process industry is defined, how it differs from other industries, and what the unique characteristics are of such a process. Then the historical background on the subject of order acceptance in a multipurpose batch process is given. In the last two sections of this chapter we state our research objective and give the thesis outline.

## 1.1 Multipurpose batch process

In this section we discuss the difference between a batch process and a flow process and a multipurpose process and multiproduct process, respectively. This gives an idea of what a multipurpose batch process looks like. Examples of multipurpose batch processes can be found in the speciality chemicals and in the pharmaceutical industry.

### 1.1.1 Batch process versus flow process

There are several points of distinction between a batch process and a flow process. Firstly batch process industries are generally found in situations with many different products and relatively low production volumes, whereas flow process industries are generally found in situations with few products and high production volumes. Secondly in batch process industries a specific amount of material is processed at the same time in a vessel. Input and output of material occurs at specific points in time. In flow process industries a continuous input and output of material takes place, and materials are processed continuously. Batch processing is inherently more flexible and therefore preferred for situations with relatively small production volumes. Furthermore, batch processing is preferred if the transformation process is less controlled, because the risk of losing materials is then limited to a single batch. Production in batch process industries is generally less controlled and automated than in flow process industries.

### 1.1.2 Multipurpose versus multiproduct

Two basic types of batch process industries are often distinguished: multiproduct and multipurpose. In multiproduct situations, all products follow the same sequence of operations

along the resources. In multipurpose situations, different products may have different routings. Multipurpose batch process industries are the most flexible type of process industries with respect to the product variety that may be produced.

## 1.2 Characteristics of a multipurpose batch process

A multipurpose batch process has the following unique characteristics, which will be discussed below.

**Multipurpose resources,** the resources can perform a variety of different processing steps.

**High product variety,** this is a result from the multipurpose resources.

**Long divergent routings,** the total production times of a product are usually long and intermediate products resulting from the same operation may be used to produce different finished products. An example of a divergent routing is given in Figure 1.1.

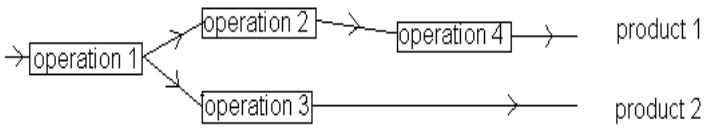


Figure 1.1: An example of a divergent routing.

**Overlapping processing steps,** the product being processed is generally a fluid or a powder that needs to be kept by a resource at any time during production.

**Waiting time restrictions,** by definition, intermediate products are not stable during an operation and, therefore must be processed further immediately without delay.

These characteristics, in particular the last two make it difficult to schedule jobs. Usually it takes a long time. It would be very convenient to have a quick method to estimate whether or not a job can be executed. So we want to have some kind of policy to give a customer a quick answer whether or not his/her order can be executed, i.e. an order acceptance function.

## 1.3 Historical Background

Finding such a method is a research project from the Department of Technology Management under the supervision of Professors W. Bertrand and J. Fransoo. The two PhD's that have worked on this project are W. Raaymakers (1995-1999) and C. Ivanescu (2000-2004).

The no-wait job shop scheduling problem is NP-hard. An NP-hard problem is a mathematical problem for which, even in theory, no shortcut or smart algorithm is possible that

would lead to a simple or rapid solution. Instead, the only way to find an optimal solution is a computationally-intensive, exhaustive analysis in which all possible outcomes are tested. For the no-wait job shop scheduling problem the Simulated Annealing (SA) algorithm is proposed that obtains near-optimal solutions with respect to makespan. The objective of the SA-algorithm is to minimize the makespan of a job set subject to the above constraints. We would like to refer to Raaymakers and Hoogeveen (2000) for a very elaborate explanation of the SA-algorithm. The SA-algorithm has two big problems, firstly it has large computation time, and secondly because it does not find optimal solutions it has to be repeated a few times per job set to get a reliable result.

In Raaymakers (1999) the research objective was to develop a method, different from the SA-algorithm, to support the order acceptance and capacity loading decisions in multipurpose batch process industries. Raaymakers chose for discrete planning periods and the aim of her thesis was to estimate the achievability of job sets for given planning periods. She considered a deterministic situation in which jobs are assumed to have predefined processing structure. The processing structure defines the number of processing steps of a job, and for each processing step the resource type, the processing time and the time delay. The processing times an time delay are assumed to be deterministic. The methods that are developed are expected to keep there validity under stochastic conditions. The method that was developed by Raaymakers is based on a regression model that estimates the makespan of a job set. The regression model is in comparison to the simulated annealing approach a lot quicker.

In Ivanescu (2004) the order acceptance function in a multi-resource production system with overlapping processing steps, no-wait restrictions among processing steps and stochastic processing times is studied. Ivanescu elaborates the regression approach of Raaymakers to investigate the order acceptance in multipurpose batch process industries.

## 1.4 Research objective

When using the order acceptance functions constructed by Ivanescu a problem with reliability occurs. The predefined job set service level is no longer reached. This problem is called selectivity. Our research objective is to look closer at one type of order acceptance, i.e. the regression policy, and try and capture selectivity . We also want to think about possible solutions for the problem. Our problem approach consists of two parts. Firstly we take a closer look at how the regression model was built in Ivanescu (2004). And secondly we simulate the problem to get an idea of what selectivity mathematically is.

## 1.5 Thesis outline

In Chapter 2 we elaborate on the order acceptance in multipurpose batch process industries. Chapter 3 gives an overview of the building and validation of the regression model used to estimate the makespan of a job set. In Chapter 4 we will take a closer look at data-splitting which can be used for the validation of a regression model. These two chapters are our first problem approach, i.e. investigating the regression model. Chapter 5 gives an explanation of the regression policy and the problem: selectivity. In Chapter 6 we try to capture selectivity mathematically by means of a simulation and calculations. Chapter 7 elaborates on the possible solutions of the problem. Finally in the last chapter the conclusions of this thesis are stated.

# Chapter 2

# Order Acceptance in Multipurpose Batch Process Industries

For a company it is important to have a method to give a customer an answer to whether or not his/her order can be carried out within certain time limits. Of course we want to have a method that gives reliable information, because a company wants to deliver on time. And on the other hand we do not want to make the customer wait to long for an answer. Before we can construct an order acceptance function that satisfies these constraints we first take a closer look at the order acceptance function in general and at the order arrival and acceptance procedure in a multipurpose batch process.

## 2.1 Order acceptance

By definition the order acceptance function is a decision function that accepts or rejects orders based on the availability of sufficient capacity to complete the orders before their requested due date. In real life a customer usually wants a quick and reliable answer to whether or not his/her order can be executed. Hence, an order acceptance function should be quick and reliable.

## 2.2 Order arrival/acceptance procedure

In this section we describe the order arrival/acceptance procedure.

First of all we have to look at our production department. There are several resources, some of the same type. According to Raaymakers (1999) 10 resources of 5 different types (so 2 resources per type) is a realistic situation. Now customers place an order to produce something. Each customer places exactly one order, we also call an order a job. The planning horizon, e.g. one day, is split up in several planning periods of the same length $T$. Orders that enter in one planning period, e.g. $[0, T]$, have to be completed at $t = 2T$ and are processed in $[T, 2T]$ so their makespan (time it takes to execute the orders) should not exceed $T$ (see Figure 2.1). If we want to see whether or not a job can be accepted we do not have to worry about the jobs that are already in process, because they have no relation with the job set that we are planning at that moment. So as a result we know that the system is empty at the beginning and at the end of every planning period. We also know that every job has certain job characteristics . For each job $j$, a specific number of no-wait processing steps $s_j$ is

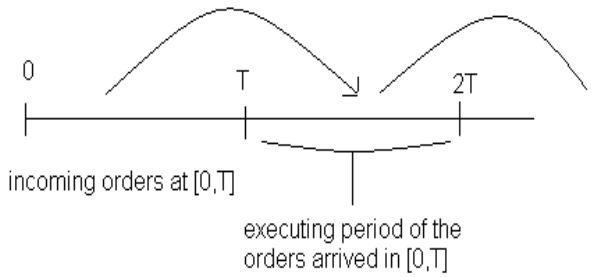


Figure 2.1: A planning period with arriving orders.

required. In multipurpose batch process industries, the processing steps may have an overlap in time. 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 $i$ relative to the start time of the first processing step. So now we already know two job characteristics: the number of processing steps and the fixed time delay. The last two are the distribution of the processing times $F_{\mathbb{E}[P_{ij}]}$ and the expected processing time $\mathbb{E}[P_{ij}]$. Having identified the job characteristics we can look at an arriving job $j$ as an object with 4 characteristics:

$$j = (s, \delta_i, \mathbb{E}[P_i], F_{\mathbb{E}[P_i]}), \qquad i = 1, ..., s. \tag{2.1}$$

Now each time an order arrives the order acceptance function is used. We want to calculate the makespan of the arriving job plus the already accepted jobs, using the job characteristics. If the makespan exceeds the length of the planning period the incoming order is rejected otherwise the order is accepted and inserted in the already existing job set. To estimate the makespan of a job set regression analysis is used by Raaymakers (1999) and Ivanescu (2004).

In the following chapter an overview of the building and evaluation of the regression model used to estimate the makespan of a job set is given.

# Chapter 3

# Building and Validating of the Makespan Estimation Model Using Regression Analysis

As said before we want to estimate the makespan of a job set. A commonly used method to do this is regression analysis. In this chapter we elaborate on the building and evaluation of the regression model used for the estimation of the makespan.

## 3.1 Estimation model formulation

If we want to calculate the makespan of a job set we have to consider two things. First of all we know that the workload on the bottleneck resource type, i.e. the resource type with the highest utilization, puts a lower bound $LB(J)$ on the makespan of the job set $J$. Secondly because of job interactions (timing, no-wait constraints) at the scheduling level, the minimal makespan for which a feasible schedule $S_J$ is realized will often exceed this lower bound. Now the job interaction can be measured by definition by the interaction margin $I(S_J)$,

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

with $C_{\max}(S_J)$ the makespan of a job set $J$ after construction of the schedule $S_J$. We know that scheduling takes a long time, so we want to estimate the interaction margin of the job $J$ without constructing the schedule $S_J$. Hence, we then can calculate the makespan $C(J)$ of a job set $J$ as follows

$$C(J) = (1 + I(J))LB(J) \tag{3.2}$$

where $I(J)$ and is $LB(J)$ are the interaction margin and the lower bound of the job set $J$, respectively. As in Ivanescu (2004) we assume that $\ln I(J)$ can be described well by a linear regression model

$$\ln I(J) \quad = \quad \beta_0 + \beta_1 x_1 + \ldots + \beta_6 x_6 + \varepsilon \qquad \text{with} \qquad \varepsilon \sim N(0, \sigma^2). \tag{3.3}$$

Hence, $\mathbb{E}[\ln I(J)] = \beta_0 + \beta_1 x_1 + \ldots + \beta_6 x_6$ and $V[\ln I(J)] = \sigma^2$. We now express (3.3) in a slightly different form

$$\ln I(J) = x\beta + \varepsilon, \tag{3.4}$$

where $\beta = [\beta_0, \ldots, \beta_6]^T$, and $x = [1, x_1, \ldots, x_6]$. We will use this notation throughout this section. By (3.2) and (3.3), we have the following relation between $C(J)$ and the predictors $x_1, \ldots, x_6$:

$$C(J) = \left(1 + e^{\ln(I(J))}\right) LB(J) = \left(1 + e^{x\beta+\varepsilon}\right) LB(J). \tag{3.5}$$

Note that taking expectations in (3.5) does not yield the formula obtained by taking expectations in (3.2). We demonstrate this by the following calculation. First note that if $\varepsilon \sim N(0, \sigma^2)$, $e^{\varepsilon}$ has a lognormal distribution, so

$$\mathbb{E}[e^{\varepsilon}] = e^{\sigma^2/2}. \tag{3.6}$$

We can now compute the expected value of the right-hand side of (3.5).

$$\begin{aligned} \mathbb{E}[I(J)] = \mathbb{E}[e^{\ln I(J)}] &= \mathbb{E}[e^{x\beta}e^{\varepsilon}] \\ &= e^{x\beta}\mathbb{E}[e^{\varepsilon}] \\ &= e^{x\beta}e^{\sigma^2/2}. \end{aligned} \tag{3.7}$$

An obvious estimator for $C(J)$ is given by

$$\widehat{C}(J) = (1 + \widehat{I}(J))LB(J). \tag{3.8}$$

The choice of this estimator can be motivated by the fact that $\widehat{\ln I(J)}$ is the ordinary least squares estimator of $\ln I(J)$, which is also the maximum likelihood estimator because of $\varepsilon \sim N(0, \sigma^2)$, and the Invariance property of maximum likelihood estimators (the Invariance property can be found in Bain and Engelhardt (1992)).

**Theorem 3.1 (Invariance property)** *If $\widehat{\theta}$ is the maximum likelihood estimator (MLE) of $\theta$ and $u(\theta)$ is a function of $\theta$, then $u(\widehat{\theta})$ is an MLE of $u(\theta)$.*

Hence, the MLE of $e^{\ln I(J)} (= I(J))$ is $e^{\widehat{\ln I(J)}} (= \widehat{I}(J))$.

We now want to investigate wether or not the equality $\mathbb{E}[C] = \mathbb{E}[\widehat{C}]$ holds. Using (3.5) and (3.7) we obtain

$$\mathbb{E}[C] = (1 + \mathbb{E}[I(J)])LB(J) = (1 + e^{x\beta}e^{\sigma^2/2})LB(J). \tag{3.9}$$

Calculating $\mathbb{E}[\widehat{C}]$ is somewhat more difficult. First of all note that $\widehat{\beta} \sim N_7(\beta, \sigma^2(X^TX)^{-1})$, a multivariate normal distribution with mean $\beta$ and variance $\sigma^2(X^TX)^{-1}$ (for more information about the multivariate normal distribution see Appendix A). To find the mean and variance of $x\widehat{\beta}$ we require the following theorem.

**Theorem 3.2** *If $X \sim N(\mu, \Sigma)$ is a multivariate normal distributed vector and $Y = BX$ is a linear transformation of $X$, then $Y$ has a multivariate normal distribution with expected value $B\mu$ and variance $B\Sigma B^T$, i.e. $Y \sim N(B\mu, B\Sigma B^T)$.*

Hence, according to this theorem, $x\widehat{\beta} \sim N(x\beta, \sigma^2 x(X^TX)^{-1}x^T)$ follows a multivariate normal distribution. The following theorem gives the moment generating function of a $p$-variate normal distribution.

**Theorem 3.3** *Given $X \sim N(\mu, \Sigma)$ a $p$-variate normal distributed variable and $t^T = (t_1, \ldots, t_p)$ a vector of real numbers, then the moment generating function of $X$ is $M_X(t) = e^{t^T\mu + \frac{1}{2}t^T\Sigma t}$.*

So,

$$\begin{aligned} \mathbb{E}[e^{x\widehat{\beta}}] &= M_{x\widehat{\beta}}(1) \\ &= e^{x\beta}e^{\frac{1}{2}x(X^TX)^{-1}x^T\sigma^2}. \end{aligned} \tag{3.10}$$

Now (3.8) and (3.10) give

$$\mathbb{E}[\widehat{C}] = (1+\mathbb{E}[\widehat{I}(J)])LB(J) = (1+e^{x\beta}e^{\frac{1}{2}x(X^TX)^{-1}x^T\sigma^2})LB(J). \tag{3.11}$$

We can observe that $x(X^TX)^{-1}x^T$ converges to 0 if the number of observations $n$, becomes large. Hence, (3.11) yields $\mathbb{E}[\widehat{C}] = (1+e^{x\beta})LB(J)$ which is not the same as (3.9). According to Miller (1984) the detransformed estimator $\widehat{I}(J)$ provides not an estimator for the mean of the response but it is an estimator for the median of the response. A simple remedy to have an estimator of the mean is to apply an estimator of the factor $e^{\sigma^2/2}$ to the detransformed estimator $\tilde{I}(J)$,

$$\tilde{I}(J) = e^{x\widehat{\beta}+\widehat{\sigma}^2/2}. \tag{3.12}$$

Because $\widehat{\sigma}^2$ is an MLE of $\sigma^2$ and the Invariance property we know that $e^{\widehat{\sigma}^2/2}$ also an MLE of $e^{\sigma^2/2}$. Now we have to calculate $\mathbb{E}[e^{\widehat{\sigma}^2/2}]$. We know that $\frac{n\widehat{\sigma}^2}{\sigma^2} \sim \chi^2(n-p)$, with $p$ is number of $\beta$'s, and

$$\mathbb{E}[e^{\widehat{\sigma}^2/2}] = \mathbb{E}[e^{\frac{\sigma^2}{2n}\cdot\frac{n\widehat{\sigma}^2}{\sigma^2}}] = M_{\frac{n\widehat{\sigma}^2}{\sigma^2}}\left(\frac{\sigma^2}{2n}\right) = \left(\frac{1}{1-\frac{\sigma^2}{n}}\right)^{\frac{n-p}{2}} \tag{3.13}$$

If $n$ becomes large this expected value converges to $e^{\sigma^2/2}$. In order to calculate $\mathbb{E}[\widehat{C}]$ we have to know that $\widehat{\beta}$ and $\widehat{\sigma}^2$ are independent. The following theorem from Bain and Engelhardt (1992) states this.

**Theorem 3.4** *If $\widehat{\beta}$ and $\widehat{\sigma}^2$ are the MLE of $\beta$ and $\sigma^2$ then they are independent.*

So now (3.11) and (3.12) and the fact that $\widehat{\beta}$ and $\widehat{\sigma}^2$ are independent gives

$$\mathbb{E}[\widehat{C}] = (1+\mathbb{E}[e^{x\widehat{\beta}}]\mathbb{E}[e^{\widehat{\sigma}^2/2}])LB(J) = (1+e^{x\beta}e^{\sigma^2/2})LB(J). \tag{3.14}$$

From this calculation we can conclude that $\mathbb{E}[C] = \mathbb{E}[\widehat{C}]$ are approximately equal for large $n$. To show how large $n$ has to be to conclude that $\mathbb{E}[e^{\widehat{\sigma}^2/2}]$ becomes $e^{\sigma^2/2}$ we give to the following numeric example.

**Example 3.1** *Suppose $p = 7$ and $\sigma = 0.107$, hence*

$$e^{\sigma^2/2} = 1.00574$$

*and*

$$\mathbb{E}[e^{\widehat{\sigma}^2/2}] = \left(\frac{1}{1-0.107^2/n}\right)^{(n-7)/2}$$

*We now give some values for different n's.*

| $n$ | $\mathbb{E}[e^{\widehat{\sigma}^2/2}]$ |
|---|---|
| 10 | 1.00172 |
| 100 | 1.00534 |
| 1000 | 1.0057 |
| 10000 | 1.00574 |

*So when n is* 10000 *we can say that* $\mathbb{E}[e^{\widehat{\sigma}^2/2}] = e^{\sigma^2/2}$. *However for relative small n′s the value for* $\mathbb{E}[e^{\widehat{\sigma}^2/2}]$ *is also very close to* $e^{\sigma^2/2}$ *(see the table above).*

From this section we can conclude that the expected makespan of a job set $J$ can be calculated as follows

$$\widehat{C}(J) = (1 + \tilde{I}(J))LB(J). \tag{3.15}$$

## 3.2 Model building

To calculate the expected makespan of a job set we use (3.15). As said before according to Ivanescu (2004) $\ln I(J)$ can be well described by a linear regression model with six regressor variables. In this section we want to look at the building of this regression model using the techniques explained in Appendix B.

### 3.2.1 Regressors and response variable

Ivanescu (2004) has identified $I(J)$ as the response variable of the regression model and the following six (italic) regression variables. Firstly consider *the number of jobs in the job set* $n_J$. So for the first order that arrives this is 1. For an order that arrives somewhere in the middle of the planning period this is the number of already accepted jobs + 1. Secondly *the average number of processing steps per job in a job set* $\mu_s$ is calculated as follows

$$\mu_s = \frac{1}{n_J} \sum_{j=1}^{n_J} s_j \tag{3.16}$$

where $s_j$ is a job characteristic and $n_J$ as before. *The average overlap of processing steps in a job set,* $\mu_g$ can also be calculated. Therefore

$$g_j = \frac{1}{s_j - 1} \sum_{i=1}^{s_j} (1 - \frac{\delta_{i,j} - \delta_{i-1,j}}{\mathbb{E}[P_{i-1j}]}) \tag{3.17}$$

where $s_j, \delta_{i,j}$ and $\mathbb{E}[P_{ij}]$ are all job characteristics, which gives the overlap for each job, is calculated. The average overlap is then

$$\mu_g = \frac{1}{n_J} \sum_{j=1}^{n_J} g_j. \tag{3.18}$$

We can measure *the variation indicated by the dissimilarity of the processing steps in the job set* $cv^2_{\mathbb{E}[p]}$ as follows

$$cv^2_{\mathbb{E}[p]} = \frac{\sigma^2_{\mathbb{E}[p]}}{\mu^2_{\mathbb{E}[p]}} \tag{3.19}$$

with

$$\mu_{\mathbb{E}[p]} = \frac{1}{S}\sum_{j=1}^{n_J}\sum_{i=1}^{s_j}\mathbb{E}[P_{ij}] \tag{3.20}$$

and

$$\sigma^2_{\mathbb{E}[p]} = \frac{1}{S-1}\sum_{j=1}^{n_J}\sum_{i=1}^{s_j}(\mathbb{E}[P_{ij}] - \mu_{\mathbb{E}[p]})^2 \tag{3.21}$$

where $S = \sum_{j=1}^{n_J} s_j$ and all the other necessary values to calculate $cv^2_{\mathbb{E}[p]}$ are job characteristics or already calculated. The actual processing times may differ from their expected value $\mathbb{E}[P_{ij}]$ therefore we calculate *the squared coefficient of variation of the processing times* $cv_p^2$.

$$cv_p^2 = \frac{(b-a)^2}{3(b+a)^2} + \frac{4(b^2+ab+a^2)}{3k(b+a)^2} \tag{3.22}$$

with $a$ and $b$ the lower and upper bounds of the uniform distribution, which we can find by looking at the job characteristic $F_{\mathbb{E}[P_{ij}]}$ and $k$ is the shape parameter of the Erlang distribution, which is randomly allocated for each job set. Finally *the workload* $\rho_{\max}$ (quantity of capacity needed to complete the job set) has to be constructed.

$$\rho_{\max} = \frac{\overline{L}}{LB(J)} \tag{3.23}$$

where the the lower bound $LB(J)$, a single resource lower bound on the makespan is computed for each job set by dividing the workload on the bottleneck resource type by the number of resources of that type. The bottleneck resource type is the resource type with the highest utilization.

$$\overline{L} = \frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{n_m} L_n \tag{3.24}$$

where $N$ is total number of resources, $n_m$ the number of resources of type $m$

$$N = \sum_{m=1}^{M} n_m, \tag{3.25}$$

and $L_n$ is the workload on resource type $n$.

### 3.2.2 Data generation

Ivanescu (2004) has generated 100 000 data points (job sets) for the building and validation of the regression model. She uses 79 750 observations as the construction data and 20 250 as the prediction data, more or less a 80-20 split. We will now give a short overview of the generation of these 100 000 job sets. Ivanescu (2004) considers a hypothetical production departments of five resource types, with two identical resources per type. This situation is chosen because it is realistic and it does not give extreme scheduling time. For the generation of the job sets

the following four factors are considered. Firstly the number of processing step per job $n_J$ is chosen from $U(25, 65)$ and the overlap of the processing steps $g_j$ is considered. The following three factors are each varied at two levels, hence we get a $2^3$-design (see Table 3.1). The factors are, the number of processing steps $s_j$, the probability that the processing steps are executed on a particular resource type $p_m$ and the probability distribution function which gives the expected processing times $F_{\mathbb{E}[p]}$. The overlap of the processing steps can be calculated using

| Factors | L | H |
|---|---|---|
| $s_j$ | $U(4, 7)$ | $U(1, 10)$ |
| $p_m$ | $0.3, 0.25, 0.2, 0.15, 0.1$ | 0.2 for $m = 1, \ldots, 5$ |
| $F_{\mathbb{E}[p]}$ | $U(15, 35)$ | $U(1, 49)$ |

Table 3.1: Experimental factor levels for generating job sets.

the processing time of the processing steps. For a more detailed explanation we refer to Ivanescu (2004). For each combination of factor levels, 50 job sets are generated. Hence, this results in 400 job sets. All the processing times in $J$ ($J = 1, \ldots, 400$) are stochastic (e.g. Erlang distributed with the same shape parameter $k$). So a single simulated execution of the job set would not be sufficient to properly capture the effect of the stochastic processing times when developing the regression models. Thus, the execution of the job sets is repeated 250 times (we only consider one uncertainty level for each job set) to reduce the variability in the results. So finally the data set consists of 100 000 observations.

### 3.2.3 Models

Firstly Ivanescu (2004) constructed an estimation model with all main effects of the 6 regressors. Backward regression showed that all the regressors were significant. Secondly, the regression models with two-way interactions were constructed and checked for significance of regressors with stepwise regression. All estimation models were determined using ordinary least squares techniques. The first method to eliminate directly some models is looking at multicollinearity.

### 3.2.4 Multicollinearity

In Ivanescu (2004) only the regression models were considered where the $VIF$'s did not exceed 10, which is a reasonable assumption. The next five models were proposed.

| Model | Regressor variables |
|---|---|
| $A$ | $\mu_s, \mu_g, cv^2_{\mathbb{E}[p]}, cv^2_p, \rho_{\max}, n_J$ |
| $B$ | $cv^2_p \cdot \rho_{\max}$ |
| $C$ | $cv^2_p \cdot \rho_{\max}, \mu_s \cdot \rho_{\max}$ |
| $D$ | $cv^2_p \cdot \rho_{\max}, \mu_s \cdot \rho_{\max}, cv^2_{\mathbb{E}[p]} \cdot n_J$ |
| $E$ | $cv^2_p \cdot \rho_{\max}, \mu_s \cdot \rho_{\max}, cv^2_{\mathbb{E}[p]} \cdot n_J, \mu_s \cdot cv^2_{\mathbb{E}[p]}$ |

Table 3.2: The five models that initially were chosen after looking at the $VIF$'s.

### 3.2.5 Model adequacy checking

**Plot of the residuals against the fitted values**
It is useful to plot the residuals against the corresponding fitted values. If the residuals can be contained in a horizontal band, then there are no obvious model defects. In our case we can detect an outward-opening funnel pattern for all models. This indicates that the variances of

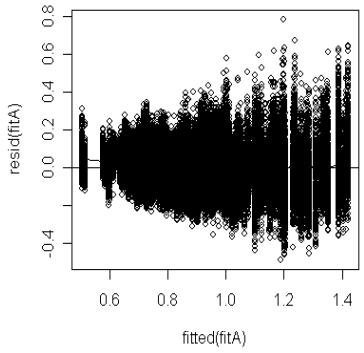


Figure 3.1: Plot of the residuals against the fitted values, outward-opening funnel pattern (model $A$).

the errors are not constant. The usual approach is applying a suitable transformation on the response variable, in this case the transformation is the natural logarithm. So we get

$$Y = \ln(I(J)), \tag{3.26}$$

with $I(J)$ our response variable. From now on we call the new response variable $Y$. After the transformation the plot of the residuals against the fitted values is a horizontal band, so there are now no obvious model defects.

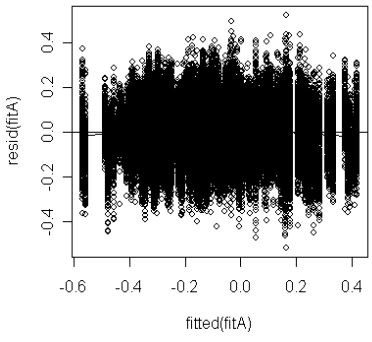


Figure 3.2: Plot of the residuals against the fitted values, horizontal band (model $A$).

**Normal probability plot**
Big violations of the normality assumption are very serious because the $t$ and $F$ statistics

and confidence and prediction intervals depend on the normality assumption. We can use the normal probability plot (also called quantile-quantile plot) to check the normality assumption.

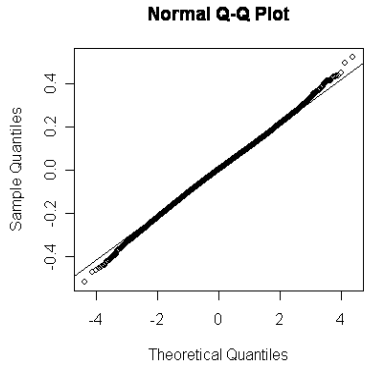


Figure 3.3: Normal probability plot (model $A$).

As we can see there are only small departures from the normality assumption, this does not affect the model greatly. Also for the other models there is no problem with normality.

### 3.2.6 Overall adequacy of the model

To check which model is the best we can use the $R^2_{adj}$ and the $\hat{\sigma}$. From the values of $R^2_{adj}$ and $\widehat{\sigma}^2$ (Table 3.3) we see that the models $A$, $D$ and $E$ have the best statistics. Hence, from now

| Model | $R^2_{adj}$ | $\hat{\sigma}^2$ |
|---|---|---|
| $A$ | 0.77 | 0.107 |
| $B$ | 0.36 | 0.177 |
| $C$ | 0.61 | 0.138 |
| $D$ | 0.74 | 0.113 |
| $E$ | 0.75 | 0.111 |

Table 3.3: $R^2_{adj}$ and $\hat{\sigma}^2$ of the initial models.

on we will only consider these three models.

### 3.2.7 Estimating the regression coefficients

The regression coefficients are estimated using ordinary least squares techniques (so we get maximum likelihood estimators because the error is normal distributed). The regression equations that belong to model $A$, $D$ and $E$ are as follows:

$$\begin{aligned}
\widehat{Y} &= -1.984 + 0.111x_1 - 0.130x_2 - 0.883x_3 + 0.991x_4 + 1.466x_5 - 0.003x_6 \\
\widehat{Y} &= -1.152 + 1.020x_4x_5 + 0.177x_1x_5 - 0.016x_3x_6 \\
\widehat{Y} &= -1.130 + 1.085x_4x_5 + 0.172x_1x_5 - 0.009x_3x_6 - 0.083x_1x_3
\end{aligned}$$

with $x_1 = \mu_s, x_2 = \mu_g, x_3 = cv^2_{\mathbb{E}[p]}, x_4 = cv^2_p, x_5 = \rho_{\max}$ and $x_6 = n_J$.

### 3.2.8 $F$-test and $t$-test

Next we are going to test the overall adequacy of the model and look at which specific regressors seem important. We may use the $F$ and $t$ statistics because we already checked the normality assumption.

**Test for significance of regression**

As can be seen in Ivanescu (2004) for the models $A, D$ and $E$ the $P$-value is very small so $H_0$ is rejected. So there is a linear relationship between the response $Y$ and any of the regressors. So there are no problems with the significance of the regression.

**Test on individual regression coefficients**

For the models $A, D$ and $E$ are all the regressor variables significant. So from these tests we know that all three models fit the data well.

## 3.3 Model validation

In the previous section we have constructed three models (model $A, D$ and $E$) that fit the data well. Now we want to test which one of these models has the best predictive performance. This is done using the data that was set aside by Ivanescu (2004) at the beginning of the study, the prediction data. The prediction data contains 20 250 observations. Ivanescu splits the data randomly, we investigate in the following chapter on data-splitting whether or not this is a good choice.

### 3.3.1 $ME$, $\sqrt{MSE}$ and $R^2_{pred}$

The quality of the three models is evaluated by means of the mean prediction error ($ME$), the square root of the mean square prediction error ($\sqrt{MSE}$) and the percentage of variability in the new data explained by the model ($R^2_{pred}$). Before we can compute these statistics we have to calculate the estimated $y_i$ using the models that we have fitted. More precisely take the values of the regressors of the 20 250 observations in the prediction data and calculate with the regression equation of each model ($A, D$ or $E$) the $\hat{y}_i$. So we get for each model 20 250 values of $\hat{y}_i$ which we can use to calculate $R^2_{pred}$, $\sqrt{MSE}$ and $ME$ . The values for the three models for these statistics are listed below. As we can see from $R^2_{pred}$ and $\sqrt{MSE}$

| Model | $R^2_{pred}$ | $\sqrt{MSE}$ | $ME$ |
|---|---|---|---|
| A | 0.7310999 | 0.1111803 | 0.00497509 |
| D | 0.7113509 | 0.1151907 | 0.00429042 |
| E | 0.7111329 | 0.1152342 | 0.00827350 |

Table 3.4: $R^2_{pred}, \sqrt{MSE}$ and $ME$ of the models $A, D$ and $E$.

and $ME$ the regression models fit the data slightly better than they predict new data. The three models have similar predictive performance but we would choose model $A$ because it gives slightly better statistics.

### 3.3.2 Estimation errors

To decide which one of the three models (that already were selected) gives the best estimation of the makespan, Ivanescu (2004) tests the quality of the models. We are going to compare the

estimated makespan (obtained by regression) to the actual realized makespan (from simulated data). For this comparison the estimation errors are introduced, which are defined as the difference between the actual makespan and the estimated makespan

$$\varepsilon = C_{\max}(S_J) - \widehat{C}_{\max}(J). \tag{3.27}$$

Two characteristics are important when the quality of a makespan estimate is considered:

- accuracy: how close are, on average, the individual estimates to their true value,
- precision: what is 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 ($SDE$). We want to calculate these two quantities for each of the three models with the construction data and the prediction data. We can see that

| Model | $ME$ | $SDE$ |
|---|---|---|
| $A$ | -0.2892601 | 74.08281 |
| $D$ | 0.1190760 | 77.28488 |
| $E$ | -2.488039 | 76.09077 |

Table 3.5: Construction data.

| Model | $ME$ | $SDE$ |
|---|---|---|
| $A$ | 0.7501452 | 69.00167 |
| $D$ | -0.9506737 | 69.818 |
| $E$ | -3.522181 | 67.84151 |

Table 3.6: Prediction data.

the variability of the estimation errors is in all models more or less the same, but the mean estimation errors of model $A$ and $D$ are a lot smaller than those of model $E$, so we could eliminate model $E$ and only consider model $A$ and $D$ because they are more accurate. Ivanescu (2004) has chosen to use model $A$.

As a remark we want to say that some of the values that have been calculated in this chapter differ from the values that can be seen in Ivanescu (2004). This is because we used `R` to calculate the values and in Ivanescu (2004) `SPSS` is used. These two program sometimes round off numbers differently which gives slightly different values. However the conclusions and choice for model $A$ remains intact.

# Chapter 4

# Data-splitting

After a regression model is fitted, using the available data, it is used for prediction, control or to learn more about the mechanism which generated the data. It should be clear that before the model is used some checks on its validity should be made as said before. Often new data is necessary to do this. In this chapter we want to look at some methods of model validation, more precisely methods to collect new data for model validation. Because the builder of the model often does not know for which purpose the user is going to give to the model, it is wise to perform all available model validation procedures if it is possible. According to Snee (1977) the following procedures are useful for checking the validity of a regression model.

**Comparison of the model predictions and coefficients with physical theory**

Negative predictions of a theoretically positive quantity or coefficients with wrong signs are indications of an inappropriate or poorly estimated model. An examination of the $VIF$'s can also give some idea of the validity of the model. We already investigated the model in the building step for large $VIF$'s so that is not necessary here.

**Collection of new data to check the model predictions**

The new data that is collected can be compared with the predictions made by the model.

**Comparison with results with theoretical models and simulated data**

In some instances theory may exist which may help to obtain insight into the model. When available, theory should always be used to check the accuracy of the model.

**Reservation of a portion of the available data to obtain an independent measure of the model prediction accuracy**

From the three methods already discussed, the collection of new data is the most preferred according to Snee (1977). In many situations this is often not possible due to lack of time, money, etc. In these cases data-splitting is used. The available data is split into two sets, one for the construction of the model (construction data) and the other to test the model validation (prediction data). A half by half split appears to be the most popular. If data are collected sequentially over time, then it seems reasonable to pick a point in time to divide the data.

According to Snee (1977) a model is not generally useful unless it has reasonably good extrapolation properties. When one finds poor prediction accuracy because the prediction data are outside the range of the construction data, it seems reasonable to use at least part of the prediction data to extend the range of the construction data and still retain the concept of the construction and the prediction data sets. On the other hand there is also need for a method to split a data set when there is no obvious criterion such as time to guide splitting. In both situations Snee (1977) proposes the DUPLEX-algorithm developed by R.W. Kennard (Kennard and Stone (1969)). The objective of this algorithm is to divide the data into two sets which cover approximately the same region and have similar statistical properties (same variance). In the remainder of this chapter we would like to elaborate on this method of data-splitting.

## 4.1 Description of the Duplex-algorithm

In this section a description of DUPLEX by Kennard and Stone (1969) and Snee (1977) is given.

Suppose we have $p$ regressor variables $(x_1, \ldots, x_p)$ and we have collected $N$ $p$-dimensional data points. We denote these $N$ points as candidates and represent them as follows

$$X = \begin{bmatrix} x_{11} & \ldots & x_{1p} \\ x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots \\ x_{\nu 1} & \ldots & x_{\nu p} \\ \vdots & \vdots & \vdots \\ x_{N1} & \ldots & x_{Np} \end{bmatrix} \tag{4.1}$$

where every row denotes the $p$ coordinates of a data point. Now we start with a list $L$ of $N$ $p$-dim candidate points and an empty construction set and prediction set $C$ and $P$, respectively. Firstly the points in $X$ are standardized as follows

$$z_{ij} = \frac{x_{ij} - \overline{x}_j}{\sum_{i=1}^{N}(x_{ij} - \overline{x}_j)^2}, \tag{4.2}$$

with $\overline{x}_j$ the average of the point $x_j$. These values are the entries of the matrix $Z$ which now has to be orthonormalized. We use the Cholesky-decomposition to orthonormalize the standardized values. Because $Z^TZ$ is a positive definite matrix (an $n \times n$ real matrix $A$ is called positive definite if $x^TAx > 0$ with $x \in \mathbb{R}^n$) it follows that

$$Z^TZ = T^TT \qquad \Rightarrow \qquad W = ZT^{-1} \qquad \begin{array}{l}\text{is standardized and orthonormalized} \\ \text{with } T \text{ an upper triangular matrix.}\end{array} \tag{4.3}$$

Now the squared Euclidean distance between every two candidate points $w_\nu = (w_{\nu 1}, \ldots, w_{\nu p}), w_\mu = (w_{\mu 1}, \ldots, w_{\mu p})$ for $\nu, \mu = 1, \ldots, N$ is calculated as follows

$$D^2_{\nu\mu} = \| w_\nu - w_\mu \|^2 = \sum_{k=1}^{p}(w_{\nu k} - w_{\mu k})^2. \tag{4.4}$$

We can now start with assigning the candidate points in $L$ to the sets $C$ and $P$. Firstly the two points that are furthest apart from each other are assigned to $C$ and deleted from $L$, the two points that are now the furthest apart are assigned to $P$ and also deleted from $L$. After this starting procedure we assign in each step a point to $C$ and $P$, respectively, according to the following strategy found in Kennard and Stone (1969).

Let $K_{1^*}, K_{2^*}, \ldots, K_{i^*}, \ldots, K_{k^*}$ with $k^* < n$ be $k$ points that have been assigned to the design (so to $C$ or $P$). Then define

$$\Delta^2_\nu(k) = \min_{i^* \in \text{design}} \{D^2_{1^*\nu}, \ldots, D^2_{k^*\nu}\} \qquad \nu \neq i^* = 1, 2, \ldots, N \tag{4.5}$$

Thus, $\Delta^2_\nu(k)$ is the minimum of the squared distances from point $\nu$ to every point $i^*$ in the design, so either to the points of $C$ or $P$.

For the $(k+1)$th point in the design, we choose among the remaining $(N-k)$ candidates using the criterion

$$\Delta^2_{k+1} = \max_{\nu \neq i^*} \{\Delta^2_\nu(k)\}. \tag{4.6}$$

Thus, we choose the point among those remaining that is farthest apart from an existing design point. Once a point is assigned it is deleted from the list $L$ of candidate points. If two or more points give the same values, the algorithm chooses randomly which point is included in the design. Hence, the algorithm does always assign a point to the design unless there are no points left to assign. The DUPLEX-algorithm stops when $L$ is empty. We implemented the DUPLEX-algorithm in `Mathematica` (the code can be found in the Appendix D). To illustrate the DUPLEX-algorithm and the implementation we work out an example given in Snee (1977).

Suppose we want to create a linear regression model and we have values from two regressor variables $(x_1, x_2)$, 16 values each. We represents these data points as follows in a matrix (see (4.7)) and in Figure 4.1 the data points are shown graphically.

$$X = \begin{bmatrix} 6 & 3 \\ 4 & 3 \\ 2 & 3 \\ 0 & 3 \\ 4 & 1 \\ 2 & 1 \\ 0 & 1 \\ -2 & 1 \\ 2 & -1 \\ 0 & -1 \\ -2 & -1 \\ -4 & -1 \\ 0 & -3 \\ -2 & -3 \\ -4 & -3 \\ -6 & -3 \end{bmatrix}. \tag{4.7}$$

After standardizing and orthonormalizing the data points we can see that the region of the values is more spherical, Figure 4.2. Now if we run the assigning procedure from the DUPLEX-algorithm we see that $C = \{1, 16, 10, 8, 3, 15, 12, 7\}$ and $P = \{4, 13, 11, 5, 2, 14, 9, 6\}$. Hence,

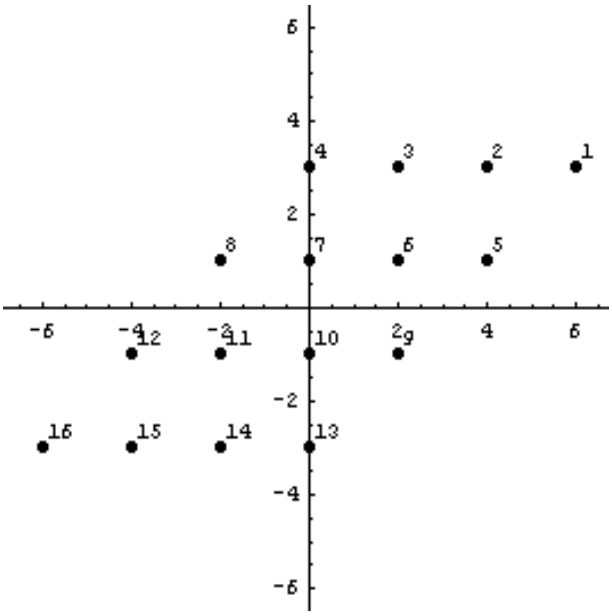


Figure 4.1: 16 data points from $X$ in (4.7).

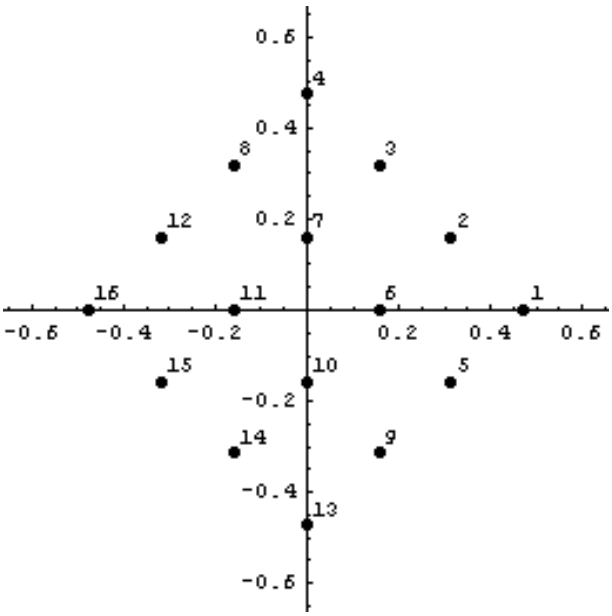


Figure 4.2: 16 data points from $X$ after standardizing and normalizing.

visually the construction and prediction points appear to be equally distributed throughout the region Figure 4.3.

To check whether or not $C$ and $P$ have the same statistical properties (same variance) we calculate the following measure

$$(|X^T X|_{\text{constr}}/|X^T X|_{\text{pred}})^{\frac{1}{p}} \tag{4.8}$$

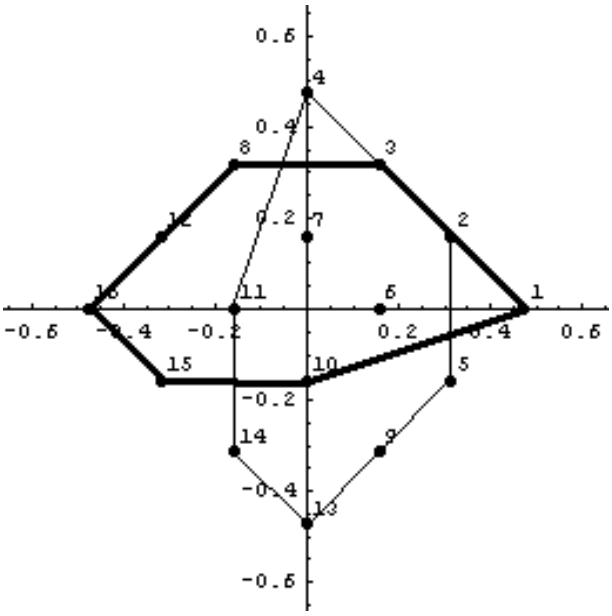


Figure 4.3: Construction set $C$ (bold line) and the prediction set $P$ (thin line) found by DUPLEX.

where $X_{\text{constr}}$ and $X_{\text{pred}}$ are the representation of the data points in $C$ and $P$, respectively. In our example this measure gives 1. Hence, we can conclude that $C$ and $P$ have relatively the same statistical properties.

## 4.2 Using the DUPLEX-algorithm

First of all we have to consider whether or not we have enough data to apply data-splitting. A rule of thumb according to Snee (1977) is that we do not consider data-splitting unless $N \geq 2p+25$. And in order to have adequate degrees of freedom for the error, giving reasonable power for significance tests and a meaningful residual analysis, the size of the construction set should be greater then $p + 10$, where $p$ is the largest number of coefficients we believe to require.

The DUPLEX-algorithm is very interesting when we know that our data set is very scattered over a region. Because if for example the construction set holds all the outliers, then the regression model will not describe the whole data very well. The DUPLEX-algorithm makes sure that the outliers are reasonably well divided between the construction and the prediction set.

To investigate the performance of the regression model constructed by Ivanescu (2004) data-splitting is also used. Because of the extreme large set of data points (100 000) and the fact that the points are simulated by a controlled sequence, it is unlikely that the set has large outliers. Hence, an arbitrary split gives no complications. When however the set should be smaller or it is uncertain whether or not the data are nicely distributed over the experimental region the use of the DUPLEX-algorithm is recommended.

# Chapter 5

# Regression Policy and Selectivity

In this section we firstly describe a method proposed by Ivanescu (2004) to assist the company in its decision on accepting or rejecting customer orders such that high resource utilization is reached and a high service level is realized for the customers. This method estimates the makespan of the already accepted job set plus the arriving order. The new order is accepted 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. Secondly we elaborate on the effect that occurs when the policy is used, i.e. selectivity.

## 5.1 Description of the regression policy

The order acceptance method that is proposed by Ivanescu (2004) is called the regression policy. This policy uses aggregate information, more precisely the regression model

$$\widehat{Y} = \widehat{\ln I(J)} = -1.984 + 0.111\mu_s - 0.130\mu_g - 0.883cv^2_{\mathbb{E}[p]} + 0.991cv^2_p + 1.466\rho_{\max} - 0.003n_J$$

to estimate the makespan of the job set. The regression model is used dynamically. This means that each time an order enters the system the regression model is used to estimate the makespan $\widehat{C}(J)$of the job set $J$ using

$$\widehat{C}(J) = (1 + \tilde{I}(J))LB(J) \tag{5.1}$$

In the regression policy orders are accepted if their expected makespan is smaller than the length of the planning period, the period in which the job set is executed. So in general, orders are accepted if

$$\widehat{C}(J) \leq T \tag{5.2}$$

where $\widehat{C}(J)$ is the expected makespan of a job set $J$ and $T$ is the length of the planning period. This equation gives a job set service level of 0.5. Ivanescu (2004) however aims at a job set service level higher than 0.5, so $1 - \alpha > 0.5$. Therefore we have an new formula for the makespan estimate

$$\widehat{C}^{1-\alpha}(J) = (1 + U^{1-\alpha})LB(J) \tag{5.3}$$

where $U^{1-\alpha}$ is the $100(1-\alpha)$ upper prediction bound for the interaction margin and calculated as follows

$$U^{1-\alpha} = e^{\widehat{\ln I(J)} + t_{\alpha,df}\widehat{\sigma}\sqrt{1+x_*(X^TX)^{-1}x_*^T}} \tag{5.4}$$

where $x_*$ and $t_{\alpha,df}$ are the row vector identifying the coordinates at which the prediction is to be made and the critical value of the Student's $t$ distribution (with $df = n - (k+1)$), respectively. So now under the regression policy orders are accepted if

$$(1 + U^{1-\alpha})LB(J) \leq T. \tag{5.5}$$

## 5.2 Random order arrival data

A simulation was set up using the regression policy to generate job sets based on this policy. For a more elaborate explanation on this simulation we refer to Ivanescu (2004). We call the job sets that result from the simulation the random order arrival data points. The random order arrival data points are within the experimental region of the regression model.

## 5.3 Problem: Selectivity

The regression policy aims at high resource utilization and a high service level. By selecting orders that maximize resource utilization, an important and often unforseen side-effect occurs, namely the mix of orders changes in such a way that the expected delivery reliability is no longer met. We call this effect selectivity. In the remainder of this chapter we show how selectivity can be observed and has its impact on performance and we give our two problem approaches.

## 5.4 Impact of Selectivity on Performance

To investigate the impact of selectivity on performance it would be a good idea to compare some performance measures in situations where no policy is used (construction and prediction data) to the situation where the regression policy is used (random order arrival). The following performance measures can be used for this investigation. Firstly the percentage of job sets that are completed before the makespan estimate is considered ,

$$PCME = \frac{\sum_{r=1}^{n_{repl}} \chi_r(J)}{n_{repl}} \cdot 100 \tag{5.6}$$

with

$$\chi_r(J) = \begin{cases} 0, & \text{if } C_{\max,r}(S_J) \leq (1 + U^{1-\alpha})LB(J); \\ 1, & \text{otherwise.} \end{cases} \tag{5.7}$$

Where $C_{\max,r}(S_J)$ is the true makespan of the $r$th job set $J$. Secondly the effective realized capacity utilization measure ,

$$ECU = \frac{\sum_{j=1}^{n_J}\sum_{i=1}^{s_j} p_{ij}}{NC(J)} \tag{5.8}$$

| | $PCME$ | $ECU$ |
|---|---|---|
| construction data | 95.02 | 0.4337 |
| testing data | 92.77 | 0.4377 |
| random order arrival | 90.96 | 0.4249 |

Table 5.1: Performance measures: non-selective versus selective.

per job set $J$ is calculated. The $ECU$ value of a entire data set is the mean of the $ECU$ values of the job sets in the data set. If we look at the values for these performance measures we observe that there is a decrease in the values from the situation where no selectivity is present to the random order arrival. In our further research we focus on the $PCME$-value of the random order arrival data. We will try to get an understanding of why this decrease in delivery reliability is present and what causes it and how it could possible be solved.

## 5.5 Problem approach

To find a solution to the problem of selectivity we follow two different paths. Firstly we already investigated the regression model. In Chapter 3 and Chapter 4 we already saw how the model was built and validated, we could not find any problems there. Therefore, for our second approach, we assume that the model is built correctly and try to simulate selectivity and capture it mathematically, this can be found in Chapter 6.

# Chapter 6

# Mathematically Understanding Selectivity

In previous chapters we have investigated the building and validation of the estimation model extensively. We could not find any problems. Hence, we assume in the following chapters that the model is correct. Now our investigation of the selectivity effect is concerned with the fact whether or not we can proof the existence of some kind of bias that induces selectivity.

## 6.1 Simulation

To get a better idea of how an order acceptance policy affects the estimation of the makespan of a job set we want to simulate the problem. The following is simulated

$$\mathbb{E}[\widehat{C}|(1+U^{1-\alpha})LB(J) \leq T] \quad - \quad \mathbb{E}[C]. \tag{6.1}$$

### 6.1.1 Overview

Before the actual programming and simulation, we make a systematic overview (see Figure 6.1) of what we want to program. First of all we assume that the regression model that was built is correct. This is a reasonable assumption because the correctness of the model is evaluated in the previous chapters. Therefore we assume that the estimated regression coefficients, the $\widehat{\beta}$'s and the estimated standard error of regression, $\widehat{\sigma}$ are true values. Hence, $\widehat{\beta}_i = \beta_i$ for $i = 0, \ldots, 6$ and $\widehat{\sigma} = \sigma$. So given the job set characteristics, $x_1, \ldots, x_6$ of a job set $J$ the evaluation of the regression model provides the true value for $\ln I(J)$, $\ln I^t(J)$, as follows

$$\ln I^t(J) = \beta_0 + \beta_1 x_1 + \ldots + \beta_6 x_6. \tag{6.2}$$

Thus, $C^t$, the true value of the makespan of $J$, is calculated using the following transformation

$$C^t = (1 + e^{\ln I^t(J)})LB(J). \tag{6.3}$$

By adding an error term, $\varepsilon \sim N(0, \sigma^2)$ to $\ln I^t(J)$, the estimated value $\ln I^e(J)$ is obtained,

$$\ln I^e(J) = \beta_0 + \beta_1 x_1 + \ldots + \beta_6 x_6 + \varepsilon. \tag{6.4}$$

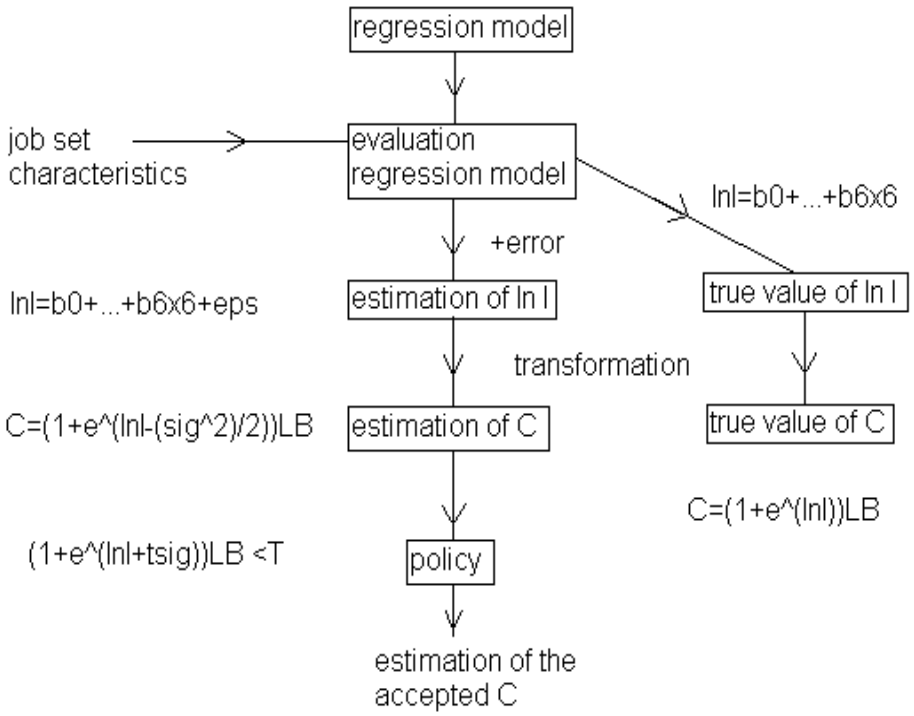


Figure 6.1: Overview of the structure of the program.

When we calculate the estimated makespan $C^e$ we have to correct the adding of the error term in the transformation to obtain an unbiased estimator for the makespan as follows,

$$C^e = (1 + e^{\ln I^e(J) - \sigma^2/2}) LB(J), \tag{6.5}$$

where we assume that $\sigma$ is a known value. Finally we apply the regression policy to the estimated makespan of a job set $J$. To get an idea of the expected value of the estimated makespan of a job set $J$ given a policy we replicate the estimated makespan of a job set several times, so we replicate the error term several times.

### 6.1.2 `Mathematica` code

The `Mathematica` code of the simulation and explanation of the code can be found in Appendix D.

### 6.1.3 Example

In this section we give an example of what our program returns as values.

**Example 6.1** *We look at the following, randomly chosen, job set $J$ with job set characteristics* $(6, 0.53, 0.2585, 0.4524, 0.87, 30) = (\mu_s, \mu_g, cv^2_{\mathbb{E}[p]}, cv^2_p, \rho_{\max}, n_J)$ *and* $LB(J) = 531$

| acceptance rate | difference | replicates |
|---|---|---|
| 0.132 | -92.5696 | 1000 |
| 0.942 | -5.32214 | 1000 |
| 1 | -1.01793 | 1000 |
| 1 | 0.0641737 | 100000 |

Table 6.1: Values from the program.

With this example we want to show the various differences in expected values when the acceptance rate is different. By adapting the length of the planning periods the acceptance rate can be manipulated. Firstly we choose a low acceptance rate (by choosing a low length for the planning period), e.g. 0.132, and observed that the difference is large in absolute value. On the other hand when the acceptance is high, e.g. 0.942, the difference is lower. When total acceptance occurs the difference is again lower and converges to 0 when we increase the number of replicates.

## 6.2 Checking the simulation

With our program we want to simulate a conditional expected value, to show that our simulation does this we elaborate in this section on the calculations made by the program. Firstly

some definitions and notions that are used in the calculations that will follow

$B$ = difference = the mean of the estimated makespans of the jobs that are accepted by the regression policy minus the true makespan of the job set.
$I(J)$ = interaction margin of the job set $J$.
$LB(J)$ = the lower bound of job set $J$.
$n$ = number of replicates.
$U^{1-\alpha}$ = $100(1-\alpha)$-upper prediction bound for $I(J)$.

### 6.2.1 Acceptance=100%

Firstly consider the case that the acceptance is 100% (i.e. no policy is present), hence every estimated makespan of the job set is always accepted. We can create an acceptance rate of 100% by choosing the length of the planning period $T$ large enough. The following holds

$$\mathbb{E}[\widehat{C}|(1+U^{1-\alpha})LB(J) \leq T] = \mathbb{E}[\widehat{C}]. \tag{6.6}$$

We examine $B$, the estimator based on the simulation, $J_i$ denotes the $i$th replicate of the job set.

$$\begin{aligned}
B &= \frac{\sum_{i=1}^{n} C_i^e}{n} - C^t \\
&= \frac{\sum_{i=1}^{n}(1+I^e(J)_i)LB(J)}{n} - (1+I^t(J))LB(J) \\
&= \frac{\sum_{i=1}^{n}(1+e^{\ln I^e(J)_i - \frac{\sigma^2}{2}})LB(J)}{n} - (1+e^{\ln I^t(J)})LB(J) \\
&= \frac{\sum_{i=1}^{n}(1+e^{\beta_0+\ldots+\beta_6 x_6+\varepsilon_i-\frac{\sigma^2}{2}})LB(J)}{n} - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\
&= \frac{nLB(J)+\sum_{i=1}^{n} LB(J)e^{\beta_0+\ldots+\beta_6 x_6+\varepsilon_i-\frac{\sigma^2}{2}}}{n} - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\
&= LB(J) + \frac{\sum_{i=1}^{n} LB(J)e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}e^{\varepsilon_i}}{n} - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\
&= LB(J) + LB(J)e^{\beta_0+\ldots+\beta_6 x_6}e^{-\frac{\sigma^2}{2}}\frac{\sum_{i=1}^{n} e^{\varepsilon_i}}{n} - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\
&= LB(J) + LB(J)e^{\beta_0+\ldots+\beta_6 x_6}e^{-\frac{\sigma^2}{2}}\frac{\sum_{i=1}^{n} e^{\varepsilon_i}}{n} - LB(J) - LB(J)e^{\beta_0+\ldots+\beta_6 x_6} \\
&= LB(J)e^{\beta_0+\ldots+\beta_6 x_6}e^{-\frac{\sigma^2}{2}}\frac{\sum_{i=1}^{n} e^{\varepsilon_i}}{n} - LB(J)e^{\beta_0+\ldots+\beta_6 x_6}
\end{aligned} \tag{6.7}$$

In the example we saw that when there is total acceptance the difference converges to 0, this gives us an estimate of the expected value of the difference. Therefore we are going to

examine whether $B$ is an unbiased estimator for the difference.

$$\begin{aligned}
\mathbb{E}[B] &= \mathbb{E}\left[LB(J)e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}\frac{\sum_{i=1}^n e^{\varepsilon_i}}{n} - LB(J)e^{\beta_0+\ldots+\beta_6 x_6}\right] \\
&= \mathbb{E}\left[LB(J)e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}\frac{\sum_{i=1}^n e^{\varepsilon_i}}{n}\right] - \mathbb{E}\left[LB(J)e^{\beta_0+\ldots+\beta_6 x_6}\right] \\
&= LB(J)\mathbb{E}\left[e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}\frac{\sum_{i=1}^n e^{\varepsilon_i}}{n}\right] - LB(J)\mathbb{E}\left[e^{\beta_0+\ldots+\beta_6 x_6}\right] \\
&= LB(J)e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}\mathbb{E}\left[\frac{\sum_{i=1}^n e^{\varepsilon_i}}{n}\right] - LB(J)e^{\beta_0+\ldots+\beta_6 x_6} \\
&= \frac{LB(J)e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}}{n}\mathbb{E}\left[\sum_{i=1}^n e^{\varepsilon_i}\right] - LB(J)e^{\beta_0+\ldots+\beta_6 x_6} \\
&= \frac{LB(J)e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}}{n}\sum_{i=1}^n \mathbb{E}[e^{\varepsilon_i}] - LB(J)e^{\beta_0+\ldots+\beta_6 x_6} \qquad (6.8)
\end{aligned}$$

We know that

$$e^{\varepsilon_i} \qquad \text{with} \qquad \varepsilon_i \sim N(0,\sigma^2)$$

has a lognormal distribution so,

$$\mathbb{E}[e^{\varepsilon_i}] = e^{\sigma^2/2}. \qquad (6.9)$$

So substituting (6.9) into (6.8) gives

$$\mathbb{E}[B] = 0 \qquad \text{when the acceptance is 100\%.} \qquad (6.10)$$

Hence, in the case of total acceptance our simulation gives an unbiased estimator for the makespan with a known $\sigma$.

### 6.2.2 Acceptance < 100%

In this section the case when the acceptance is lower than 100% is discussed. We can create an acceptance lower than 100% by choosing a small planning period, so a low value for $T$. The following indicator function for the regression policy is defined,

$$\chi_i = \begin{cases} 1, & (1+U_i^{1-\alpha})LB(J) \le T; \\ 0, & \text{otherwise.} \end{cases} \qquad (6.11)$$

Now we investigate $B$, the estimator based on the simulation.

$$\begin{aligned}
B &= \frac{\sum_{i=1}^{n} C_i^e \chi_i}{\sum_{i=1}^{n} \chi_i} - C^t \\
&= \frac{\sum_{i=1}^{n} (1 + e^{\beta_0 + \ldots + \beta_6 x_6 + \varepsilon_i - \frac{\sigma^2}{2}}) LB(J) \chi_i}{\sum_{i=1}^{n} \chi_i} - (1 + e^{\beta_0 + \ldots + \beta_6 x_6}) LB(J) \\
&= LB(J) \frac{\sum_{i=1}^{n} (1 + e^{\beta_0 + \ldots + \beta_6 x_6 + \varepsilon_i - \frac{\sigma^2}{2}}) \chi_i}{\sum_{i=1}^{n} \chi_i} - (1 + e^{\beta_0 + \ldots + \beta_6 x_6}) LB(J) \\
&= LB(J) \frac{\sum_{i=1}^{n} \chi_i + \sum_{i=1}^{n} e^{\beta_0 + \ldots + \beta_6 x_6 + \varepsilon_i - \frac{\sigma^2}{2}} \chi_i}{\sum_{i=1}^{n} \chi_i} - (1 + e^{\beta_0 + \ldots + \beta_6 x_6}) LB(J) \\
&= LB(J) \left( 1 + e^{\beta_0 + \ldots + \beta_6 x_6 - \frac{\sigma^2}{2}} \frac{\sum_{i=1}^{n} e^{\varepsilon_i} \chi_i}{\sum_{i=1}^{n} \chi_i} \right) - (1 + e^{\beta_0 + \ldots + \beta_6 x_6}) LB(J) \\
&= LB(J) e^{\beta_0 + \ldots + \beta_6 x_6 - \frac{\sigma^2}{2}} \frac{\sum_{i=1}^{n} e^{\varepsilon_i} \chi_i}{\sum_{i=1}^{n} \chi_i} - LB(J) e^{\beta_0 + \ldots + \beta_6 x_6}. \qquad (6.12)
\end{aligned}$$

The calculation of the expected value of $B$ is the following.

$$\begin{aligned}
\mathbb{E}[B] &= \mathbb{E}\left[ LB(J) e^{\beta_0 + \ldots + \beta_6 x_6 - \frac{\sigma^2}{2}} \frac{\sum_{i=1}^{n} e^{\varepsilon_i} \chi_i}{\sum_{i=1}^{n} \chi_i} - LB(J) e^{\beta_0 + \ldots + \beta_6 x_6} \right] \\
&= LB(J) e^{\beta_0 + \ldots + \beta_6 x_6 - \frac{\sigma^2}{2}} \mathbb{E}\left[ \frac{\sum_{i=1}^{n} e^{\varepsilon_i} \chi_i}{\sum_{i=1}^{n} \chi_i} \right] - LB(J) e^{\beta_0 + \ldots + \beta_6 x_6}. \qquad (6.13)
\end{aligned}$$

For the sake of readability, we write $e^{\varepsilon_i} = X_i$, and we look at the following

$$\begin{aligned}
(1 + U_i^{1-\alpha}) LB(J) &\leq T \\
(1 + e^{\ln I^e(J)_i + t_{\alpha, df}\sigma}) LB(J) &\leq T \\
(1 + e^{\beta_0 + \ldots + \beta_6 x_6 + \varepsilon_i + t_{\alpha, df}\sigma}) LB(J) &\leq T \\
e^{\beta_0 + \ldots + \beta_6 x_6 + \varepsilon_i + t_{\alpha, df}\sigma} &\leq \frac{T}{LB(J)} - 1 \\
e^{\varepsilon_i} &\leq \left( \frac{T}{LB(J)} - 1 \right) / e^{\beta_0 + \ldots + \beta_6 x_6 + t_{\alpha, df}\sigma} \\
X_i &\leq x \qquad \text{with } LB(J) \neq 0. \qquad (6.14)
\end{aligned}$$

Hence, the indicator function defined in (6.11) becomes

$$\chi_i = \begin{cases} 1, & X_i \leq x; \\ 0, & \text{otherwise.} \end{cases} \qquad (6.15)$$

To establish whether or not $\frac{\sum_{i=1}^{n} X_i \chi_i}{\sum_{i=1}^{n} \chi_i}$ is an unbiased estimator for $\mathbb{E}[X|X \leq x]$, $\mathbb{E}\left[ \frac{\sum_{i=1}^{n} X_i \chi_i}{\sum_{i=1}^{n} \chi_i} \right]$ has to be calculated on one hand and $\mathbb{E}[X|X \leq x]$ on the other hand. The calculation of

$$\mathbb{E}\left[ \frac{\sum_{i=1}^{n} X_i \chi_i}{\sum_{i=1}^{n} \chi_i} \right] \qquad (6.16)$$

is illustrated with the following example.

**Example 6.2** *Suppose as an example that $n = 2$, we then wish to calculate $\mathbb{E}\left[\frac{X_1\chi_1+X_2\chi_2}{\chi_1+\chi_2}\right]$. We now that $X_i$ is lognormally distributed hence,*

$$f_{X_i}(x) = \begin{cases} f(x), & x > 0; \\ 0, & \textit{otherwise.} \end{cases} \tag{6.17}$$

*We know now that (expected value of a transformation of two independent random variables)*

$$\mathbb{E}\left[\frac{X_1\chi_1 + X_2\chi_2}{\chi_1 + \chi_2}\right] = \int_0^x \int_0^x \frac{s\chi_1(s) + t\chi_2(t)}{\chi_1(s) + \chi_2(t)} f(s)f(t)dsdt \tag{6.18}$$

*There are four ($2^2$) possible scenarios for the two independent random variables $X_1, X_2$.*

- $X_1 < x, X_2 < x \quad \mapsto \frac{s\chi_1(s)+t\chi_2(t)}{\chi_1(s)+\chi_2(t)} = \frac{s+t}{2}$,
- $X_1 < x, X_2 > x \quad \mapsto \frac{s\chi_1(s)+t\chi_2(t)}{\chi_1(s)+\chi_2(t)} = s$,
- $X_1 > x, X_2 > x \quad \mapsto \frac{s\chi_1(s)+t\chi_2(t)}{\chi_1(s)+\chi_2(t)} = 0 \quad$ *by definition,*
- $X_1 > x, X_2 < x \quad \mapsto \frac{s\chi_1(s)+t\chi_2(t)}{\chi_1(s)+\chi_2(t)} = t$.

*So the integral (6.18) becomes*

$$\begin{aligned} = & \int_0^x \int_0^x \frac{s+t}{2} f(s)f(t)dsdt + \int_0^x \int_x^\infty sf(s)f(t)dsdt + \int_x^\infty \int_x^\infty 0f(s)f(t)dsdt \\ & + \int_x^\infty \int_0^x tf(s)f(t)dsdt \\ = & F(x)\int_0^x tf(t)dt + (1-F(x))\int_0^x tf(t)dt + 0 + (1-F(x))\int_0^x tf(t)dt \\ = & [F(x) + 2(1-F(x))]\int_0^x tf(t)dt. \end{aligned} \tag{6.19}$$

*We can calculate this value also for $n = 3$ and $n = 4$, this gives for $n = 3$*

$$\mathbb{E}\left[\frac{X_1\chi_1 + X_2\chi_2 + X_3\chi_3}{\chi_1 + \chi_2 + \chi_3}\right] = [F(x)^2 + 3(1-F(x))^2 + 3F(x)(1-F(x))]\int_0^x tf(t)dt \tag{6.20}$$

*and for $n = 4$*

$$\begin{aligned} \mathbb{E}\left[\frac{X_1\chi_1 + X_2\chi_2 + X_3\chi_3 + X_4\chi_4}{\chi_1 + \chi_2 + \chi_3 + \chi_4}\right] = & [F(x)^3 + 4(1-F(x))^3 + 4F(x)^2(1-F(x)) \\ & + 6F(x)(1-F(x))^2]\int_0^x tf(t)dt. \end{aligned} \tag{6.21}$$

When we study these examples we can extract the general formula for the expected value (6.16) by counting the number of times a case occurs and what its contribution to the coefficient before $\int_0^x tf(t)dt$ is.

Hence, suppose we have $n$ random variables $X_i$ then we can distinguish the following $2^n$ possible cases. Firstly consider the case where for all $i$ holds that $X_i < x$, this case contributes the term $F(x)^{n-1}$ to the sum and the case occurs $1 = \binom{n}{n}$ times. The following case is the

case where all $X_i$ are smaller than $x$, so $X_i < x$ except 1, i.e. $X_j > 1$. This case occurs $\binom{n}{n-1}$ times and contributes each time $F(x)^{n-2}(1-F(x))$ to the sum. In the third case all $X_i > x$ except two, this case occurs $\binom{n}{n-2}$ and contributes each time the term $F(x)^{n-3}(1-F(x))^2$. This way we can construct all cases. The last two cases will be the case where for all $i$ except 1 holds that $X_i > x$, this case occurs $\binom{n}{1}$ times and contributes $(1-F(x))^{n-1}$, and the case where for all $i$ holds that $X_i > 0$ and this case contributes by definition 0. This leads us to the following theorem.

**Theorem 6.1** *Given a continuous random variable $X_i$ with distribution function $F$ and probability density function $f$, $x \geq 0$ and an indicator function $\chi_i$ as in (6.15), the following holds*

$$\mathbb{E}\left[\frac{\sum_{i=1}^{n} X_i\chi_i}{\sum_{i=1}^{n}\chi_i}\right] = \left[\sum_{i=0}^{n-1}\binom{n}{n-i}F(x)^{n-i-1}(1-F(x))^i\right]\int_0^x tf(t)dt. \tag{6.22}$$

Now if we use Newton's Binomial Formula, i.e. $\sum_{i=0}^{n}\binom{n}{i}k^i p^{n-i} = (k+p)^n$ the coefficient of the integral in (6.22) becomes

$$\begin{aligned}
\sum_{i=0}^{n-1}\binom{n}{n-i}F(x)^{n-i-1}(1-F(x))^i &= \frac{1}{F(x)}\left[\sum_{i=0}^{n-1}\binom{n}{n-i}F(x)^{n-i}(1-F(x))^i\right] \\
&= \frac{1}{F(x)}((F(x)+(1-F(x)))^n - (1-F(x))^n) \\
&= \frac{1}{F(x)}(1-(1-F(x))^n) \\
&= \frac{1}{F(x)} \qquad \text{as} \qquad n\to\infty.
\end{aligned} \tag{6.23}$$

On the other hand we can calculate $\mathbb{E}[X|X \leq x]$, with $x \geq 0$ is fixed, the following way. By definition the conditional probability of a random variable is the following.

$$\mathbb{P}(X \geq t|X \leq x) = \frac{\mathbb{P}(X \geq t \cap X \leq x)}{\mathbb{P}(X \leq x)} = \frac{\mathbb{P}(t \leq X \leq x)}{\mathbb{P}(X \leq x)}. \tag{6.24}$$

The expected value of a continuous random variable is by definition,

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x dF_X(x) = \int_{-\infty}^{\infty} x f_X(x)dx \tag{6.25}$$

$$\text{with} \qquad F(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^{x} f_X(t)dt. \tag{6.26}$$

Hence, (6.24) and the fact that $x \geq 0$ gives

$$\mathbb{P}(X \geq t|X \leq x) = \frac{\int_t^x f_X(y)dy}{\int_0^x f_X(y)dy}. \tag{6.27}$$

The expected value $\mathbb{E}[X|X \leq x]$ is rewritten as follows,

$$\begin{aligned}
\mathbb{E}[X|X \leq x] &= \mathbb{E}[Z] \qquad\qquad \text{with} \qquad Z = X|X \leq x \\
&= \int_{-\infty}^{\infty} z dF_Z(z).
\end{aligned} \tag{6.28}$$

To evaluate (6.28), $F_Z(z)$ has to be calculated

$$F_Z(z) \quad = \quad \mathbb{P}(Z \leq z) = \mathbb{P}(X \leq z | X \leq x) = 1 - \mathbb{P}(X \geq z | X \leq x). \tag{6.29}$$

The last step is valid because $A = (A \cap B) \cup (A \cap B^c) \Rightarrow \mathbb{P}(A \cap B) = \mathbb{P}(A) - \mathbb{P}(A \cap B^c)$. Hence, $\mathbb{P}(X \leq t | X \leq x) = 1 - \mathbb{P}(X \geq t | X \leq x)$ holds. Now we can calculate $\frac{d}{dz} F_Z(z) = f_Z(z)$

$$\begin{aligned}
\frac{d}{dz} F_Z(z) &= \frac{d}{dz}(1 - \mathbb{P}(X \geq z | X \leq x)) \\
&= \frac{d}{dz}\left(1 - \frac{\int_z^x f_X(y)dy}{\int_0^x f_X(y)dy}\right) = \frac{d}{dz}(1) - \frac{d}{dz}\left(\frac{\int_z^x f(y)dy}{\int_0^x f(y)dy}\right) \\
&= 0 - \frac{1}{\int_0^x f_X(y)dy} \frac{d}{dz}\left(\int_z^x f_X(y)dy\right) \\
&= \frac{f_X(z)}{\int_0^x f_X(y)dy} = \begin{cases} \frac{f_X(z)}{\mathbb{P}(X \leq x)}, & z \leq x; \\ 0, & z > x. \end{cases}
\end{aligned} \tag{6.30}$$

Substituting (6.30) into (6.28) gives the following

$$\mathbb{E}[X | X \leq x] \quad = \quad \frac{\int_0^x z f_X(z)dz}{\mathbb{P}(X \leq x)}. \tag{6.31}$$

So because $\mathbb{E}\left[\frac{\sum_{i=1}^n X_i \chi_i}{\sum_{i=1}^n \chi_i}\right] = \mathbb{E}[X | X < x]$ for large $n$, we can say that $\frac{\sum_{i=1}^n X_i \chi_i}{\sum_{i=1}^n \chi_i}$ is an asymptotically unbiased estimator for $\mathbb{E}[X | X < x]$.

We now can finish calculation (6.13).

$$\begin{aligned}
\mathbb{E}[B] &= \mathbb{E}\left[\frac{\sum_{i=1}^n C_i^e \chi_i}{\sum_{i=1}^n \chi_i} - C^t\right] \\
&= LB(J) e^{\beta_0 + \ldots + \beta_6 x_6 - \frac{\sigma^2}{2}} \frac{1}{F_{e^{\varepsilon_i}}(x)} \int_0^x t f_{e^{\varepsilon_i}}(t)dt - LB(J) e^{\beta_0 + \ldots + \beta_6 x_6} \\
&= LB(J) e^{\beta_0 + \ldots + \beta_6 x_6} \frac{e^{-\frac{\sigma^2}{2}}}{F_{e^{\varepsilon_i}}(x)} \int_0^x t f_{e^{\varepsilon_i}}(t)dt - LB(J) e^{\beta_0 + \ldots + \beta_6 x_6}.
\end{aligned} \tag{6.32}$$

Hence,

$$\frac{1}{F_{e^{\varepsilon_i}}(x)} \int_0^x t f_{e^{\varepsilon_i}}(t)dt \tag{6.33}$$

is the bias that occurs if we use the regression policy.

## 6.3 Description of this bias

We know that $e^{\varepsilon_i}$ has a lognormal distribution and $\varepsilon_i \sim N(0, \sigma^2)$. This lognormal distribution has the following probability density function

$$f_{e^{\varepsilon_i}}(x) = \begin{cases} \frac{e^{-1/2\left(\frac{\ln(x)}{\sigma}\right)^2}}{x\sigma\sqrt{2\pi}}, & x > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{6.34}$$

We now look at the distribution function of $e^{\varepsilon_i}$. In general $F(x) = \int_{-\infty}^{x} f_X(x)dx$ hence,

$$F_{e^{\varepsilon_i}}(x) = \begin{cases} \displaystyle\int_0^x \frac{e^{-1/2\left(\frac{\ln(x)}{\sigma}\right)^2}}{x\sigma\sqrt{2\pi}}dx, & x > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{6.35}$$

In our case $x = \left(\frac{T}{LB(J)} - 1\right) / e^{\beta_0 + \ldots + \beta_6 x_6 + t_{\alpha,df}\sigma}$, and because $LB(J)$ is the lower bound of the makespan of the job set $J$ it seems reasonable to avoid choosing a planning period $T$ that is smaller than the lower bound in our simulation. In real life if at some point the lower bound of a job set is larger than the length of the planning period the job set wil not be accepted, so the $\varepsilon_i$ of that job set will not be considered. Therefore we will not have a negative $x$.
The bias we found is a function of $x$

$$b(x) = \frac{1}{F_{e^{\varepsilon_i}}(x)} \int_0^x t f_{e^{\varepsilon_i}}(t)dt. \tag{6.36}$$

If we plot $b(x)$ we get Figure 6.2. Remark that the origin of the axes is not $(0, 0)$. As can

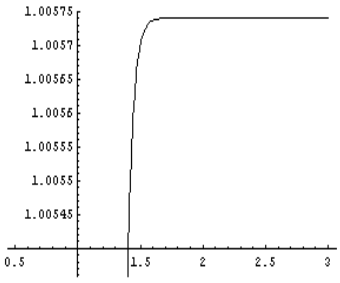


Figure 6.2: The function $b(x)$.

be seen converges the function $b(x)$ to a constant value, this value is $1.00574 = e^{\frac{\sigma^2}{2}}$, where $\sigma = 0.107$. This can be explained as follows.

$$\begin{aligned} b(x) &= \frac{1}{F_{e^{\varepsilon_i}}(x)} \int_0^x t f_{e^{\varepsilon_i}}(t)dt \\ &= \frac{1}{F_{e^{\varepsilon_i}}(x)} \int_0^x \int_0^t 1du f_{e^{\varepsilon_i}}(t)dt \\ &= \frac{1}{F_{e^{\varepsilon_i}}(x)} \int_0^x \int_0^u f_{e^{\varepsilon_i}}(t)dtdu \\ &= \frac{\int_0^x F_{e^{\varepsilon_i}}(u)du}{F_{e^{\varepsilon_i}}(x)}. \end{aligned} \tag{6.37}$$

Hence,

$$\begin{aligned}\lim_{x\to\infty} b(x) &= \lim_{x\to\infty} \frac{\int_0^x F_{e^{\varepsilon_i}}(u)du}{F_{e^{\varepsilon_i}}(x)} \\ &= \frac{\lim_{x\to\infty} \int_0^x F_{e^{\varepsilon_i}}(u)du}{\lim_{x\to\infty} F_{e^{\varepsilon_i}}(x)} \\ &= \frac{\mathbb{E}[e^{\varepsilon_i}]}{1} = e^{\frac{\sigma^2}{2}}. \end{aligned} \tag{6.38}$$

In this chapter the bias, $b(x)$, that occurs when the regression policy is used is investigated and identified. In the following chapter two possible solutions to solve the problem of not reaching the predefined service level due to this bias are given.

# Chapter 7

# Possible Solutions

In this chapter two possible approaches of how the problem of not reaching the service level can be solved are given. The first solution has to be considered at the beginning of the process and the second has to act each time the regression policy is used.

## 7.1 Adapting the $\alpha$

In this section we will give a solution that tries to correct the problem at the beginning of the process.

The regression policy has a predefined job set service level, in Ivanescu (2004) a job set service level of 95% is defined. We saw in Chapter 5 that this level is not reached when using the policy, the random order arrival data give a $PCME$ of 90.96. A possible solution of this problem is to give the regression policy a predefined service level that is higher than the service level that should be reached in the end, so the decrease in service level is less of a problem. It would be interesting to find how much this decrease is, there is probably a certain connection between this decrease and the predefined service level. Hence, the research objective is to find a relation between the service level that is reached and the predefined service level, given to the regression policy.

### 7.1.1 Approach

We can deal with this the following way, repeat the simulation of the random order arrival data with several different predefined service levels (e.g. $(1-\alpha)100\% = 5, 10, \ldots, 90, 95, 100$) and calculate for every of these levels the percentage of job sets on time, $PCME$. Now plot the $\alpha$ against the $PCME$ and try to extract a relationship. It would be very interesting if we could find a relation of the type $f(PCME) = \alpha$. Hence, if we want to reach a certain job set service level ($PCME$) we can calculate which $\alpha$ we should give the regression policy. To find such a relation between the $\alpha$ and the $PCME$ the whole simulation of the random order arrival data has to be repeated several times. This is a very elaborate job. So as we can see is this solution not very difficult but it is a lot of work to finally find a relation between $PCME$ and $\alpha$. At this point there is no guarantee that there is a nice relationship between $\alpha$ and the $PCME$, and finding this relation takes a long time (as already said). Of course we could also just take a larger service level at the beginning and hope that we reach the right service level (e.g. choose 97% to reach 95%) but that is of course not very nice (mathematically).

Therefore an other method is explained to correct the bias in the remainder of this chapter.

## 7.2 Correcting for the bias

It would be interesting if we could correct the bias that occurs when we use the regression policy and so obtain an unbiased estimator for the makespan of a job set. Firstly we remember what the bias was

$$b(x) = \frac{1}{F_{e^{\varepsilon_i}}} \int_0^x t f_{e^{\varepsilon_i}}(t)dt \qquad \text{where} \qquad x = \left(\frac{T}{LB(J)}\right) / e^{\beta_0+\ldots+\beta_6 x_6 + t_{\alpha,df}\sigma}. \tag{7.1}$$

Now if the job set that we are considering is not accepted, the incoming order is rejected, so we do not have to worry about underestimating the makespan of the job set because we do not accept the job set. On the other hand if the incoming order is accepted, it could be possible that we do not have an unbiased estimation of the makespan. To investigate whether or not the estimation is unbiased the value $b(x)$ of the job set is calculated using (7.1). The bias converges to the limit $e^{\frac{\sigma^2}{2}}$, as can be seen in the previous chapter. Therefore the following two cases are distinguished.

### 7.2.1 $b(x) = e^{\frac{\sigma^2}{2}}$

When this is the case there is no problem. The estimation of the expected makespan of the job set is unbiased, as can be seen with the calculation of $\mathbb{E}[B]$,

$$\begin{aligned}
\mathbb{E}[B] &= \mathbb{E}\left[\frac{\sum_{i=1}^n C_i^e \chi_i}{\sum_{i=1}^n \chi_i} - C^t\right] \\
&= LB(J)e^{\beta_0+\ldots+\beta_6 x_6 - \frac{\sigma^2}{2}} b(x) - LB(J)e^{\beta_0+\ldots+\beta_6 x_6} \\
&= LB(J)e^{\beta_0+\ldots+\beta_6 x_6 - \frac{\sigma^2}{2}} e^{\frac{\sigma^2}{2}} - LB(J)e^{\beta_0+\ldots+\beta_6 x_6} \\
&= 0.
\end{aligned} \tag{7.2}$$

Therefore the calculation of the estimation of the expected makespan is unbiased so we can accept the incoming order.

### 7.2.2 $b(x) < e^{\frac{\sigma^2}{2}}$

On the other hand it can occur that the bias is smaller than $e^{\frac{\sigma^2}{2}}$. If this is the case, the makespan is underestimated, hence, the estimation of the makespan is biased. This also explains why the predefined service level is not reached. When this is the case we propose to multiply $e^{\ln I^e(J)}$ with the correction factor $\mathbf{z} = \dfrac{e^{\frac{\sigma^2}{2}}}{b(x)}$. The estimation indeed becomes

unbiased with this correction, according to (6.12).

$$\begin{aligned} B &= \frac{\sum_{i=1}^{n}(1+\mathbf{z}e^{\ln I_i^e(J)-\frac{\sigma^2}{2}})LB(J)\chi i}{\sum_{i=1}^{n}\chi i} - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\ &= LB(J)\frac{\sum_{i=1}^{n}\chi_i + \mathbf{z}e^{\beta_0+\ldots+\beta_6 x_6+\varepsilon_i-\frac{\sigma^2}{2}}\chi i}{\sum_{i=1}^{n}\chi i} - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\ &= LB(J)\left(1+\mathbf{z}e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}\frac{\sum_{i=1}^{n}e^{\varepsilon_i}\chi i}{\sum_{i=1}^{n}\chi i}\right) - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J). \end{aligned} \tag{7.3}$$

Hence,

$$\begin{aligned} \mathbb{E}[B] &= LB(J)\left(1+\mathbf{z}e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}\mathbb{E}\left[\frac{\sum_{i=1}^{n}e^{\varepsilon_i}\chi i}{\sum_{i=1}^{n}\chi i}\right]\right) - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\ &= LB(J)\left(1+\frac{e^{\frac{\sigma^2}{2}}}{b(x)}e^{\beta_0+\ldots+\beta_6 x_6-\frac{\sigma^2}{2}}b(x)\right) - (1+e^{\beta_0+\ldots+\beta_6 x_6})LB(J) \\ &= 0. \end{aligned} \tag{7.4}$$

Hence, $(1+\mathbf{z}e^{\ln I^e(J)-\frac{\sigma^2}{2}})LB(J)$ gives an unbiased estimator for the makespan and because of $\mathbf{z} = \dfrac{e^{\frac{\sigma^2}{2}}}{b(x)} > 1$, this new value of the makespan of the job set $J$ larger than the old value. Now we again consider whether or not our job set will be accepted by the regression policy, i.e. check

$$\left(1+\mathbf{z}e^{\ln I^e(J)+t_{\alpha,df}\sigma}\right)LB(J) \leq T. \tag{7.5}$$

If the above equation holds, the incoming order can be accepted. If (7.5) does not hold, the incoming order should not be accepted. At every step of the process the jobs in the job set have a different configuration due to timing and no wait constraints, therefore the correction factor does not have to be remembered. To make the correcting method more clear we consider the following numerical example.

**Example 7.1** *Suppose* $T = 1200, LB(J) = 531$. *When the order* $j$ *enters the system the job set* $J$*, that is then constructed, has the following six job set characteristics*
$(x_1, x_2, x_3, x_4, x_5, x_6) = (6, 0.53, 0.2585, 0.4524, 0.87, 30)$.
*The estimated makespan, using the regression model, of this job set is* 1068.88.
*The value that is used to test the policy is* 1177.45*, this value is below* 1200 *so we would initially accept the incoming job* $j$.
*We now calculate the value for* $b(x)$ *this gives* $0.939556 < e^{\frac{\sigma^2}{2}} = 1.00574$ *so we should correct the bias.*
*After the correction, the estimation for the makespan is* 1106.77 *and the value for the policy is* 1222.99 *which is higher than* 1200.
*So based on the regression policy we should choose not to accept the incoming order* $j$.

So in the simulation whenever the regression policy is used, for the generation of the random order arrival job sets, the bias $b(x)$ should be calculated and compared with $e^{\frac{\sigma^2}{2}}$. After that the correction and the regression policy should again be considered. Correcting the bias each time it occurs is a more elegant method of dealing with the bias than the first method.

# Chapter 8

# Conclusion

In a multipurpose batch process regression analysis can be used to estimate the makespan of a job set. The estimation model is also employed when the orders enter the system. The regression policy chooses which order should be accepted. When using the regression policy in multipurpose batch process industries an interesting effect can be observed. The mix of orders changes in such a way that the predefined service level is no longer met. The difference in service level can be demonstrated by calculating the values for the $PCME$. This is a strange effect because in regression policy it is assumed that when an estimation model is well defined the expected values of the response and the estimation of the response are the same for all the data points in the experimental region. A possible explanation to the problem is that at some point the regression policy chooses orders that are all in a certain region of the experimental region and therefore the model that is fitted on the whole range does not describe the data as well as it should.

The aim of this thesis was to investigate the selectivity effect that can be observed in multipurpose batch processes when using the regression policy. We have approached the problem from two different directions. Firstly, maybe it would be possible to find a better regression model to use in the regression policy. Hence, the building and validation of the regression model, build in Ivanescu (2004), is examined closely in this thesis. The statistical package `R` was applied to do this examination. We also investigated the transformation that was made by Ivanescu to transform the response variabele $\ln I(J)$ back to the estimated value for the makespan of a job set $J$. After that we looked closer at how the data was split in construction and prediction data (in Ivanescu (2004) this is done in a random way). This has brought our attention to a very interesting algorithm to split data sets into two parts with the same statistical properties, i.e. the DUPLEX-algorithm. We investigated this algorithm and programmed it in `Mathematica`. Because the data set from Ivanescu is very large, 100 000 data points and there were no outliers, a random split of the data is reliable, but nevertheless the DUPLEX-algorithm is an interesting algorithm to consider when working with smaller data sets with possible outliers. From the calculations and investigation we could conclude that the regression model from Ivanescu is correct.

Our other approach involves conditional expectations. We have built a simulation to get a better understanding of the expected value of the estimated makespan of a job set given a policy. This expected value was compared with the expected value of the makespan of the job set. In theory we assumed that these two expected values would give the same value. But according to our simulation this was only the case when there is 100% acceptance, i.e. no

policy is present. However in the case of an acceptance of less than 100%, i.e. the policy is present, a difference (bias) between the two expected values can be observed. The calculation of this bias and, which we call $b(x)$, can be found in section 6.2.2. The expression of $b(x)$ is given by (6.36).

After identifying this biasing factor we proposed two possible solutions to the problem of not reaching the predefined service level, i.e. selectivity. The first solution consist of just starting with a higher service level than what is aimed for in the end. The second solution makes a correction for the bias each time the regression policy is used. Further research on this topic could include working out the possible solutions mentioned in this thesis. To execute these solutions, new simulation experiments should be set up to generate new data and then the $PCME$ values can be compared. On the other hand there could be looked into the fact, as mentioned earlier, that the random order arrival data should be described by a regression model that is fitted exactly on this data.

Order acceptance in multipurpose batch process industries, and especially the selectivity effect, is a very interesting subject to work on. With this thesis we hope to have given a insight into the existence of the bias that occurs when the regression policy is used. Also we wanted to hand some possible solutions of how the problem can be solved, which can be seen as a start of further research. In this thesis we only considered the regression policy but there are also other order acceptance policies,e.g. the scheduling policy that suffer from the problem of selectivity. Hence, the research on this fascinating subject can certainly be elaborated further.

# Appendix A

# Probability Distributions

In this appendix the probability distributions and their characteristics that are used in the thesis can be found.

**A.1 (Normal distribution)** *The random variable $X$ is normally distributed if the following holds, $X \sim N(\mu, \sigma^2)$*

- *Parameters :* $-\infty < \mu < \infty$, $\sigma > 0$
- *Values :* $(-\infty, \infty)$
- *Density function :* $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- *Expectation :* $\mu$
- *Variance :* $\sigma^2$
- *Characteristic function :* $e^{i\mu t - (t^2\sigma^2/2)}$

**A.2 (Lognormal distribution)** *The random variable $X$ has a lognormal distribution if* $\ln X \sim N(\mu, \sigma^2)$

- *Parameters :* $-\infty < \mu < \infty$, $\sigma > 0$
- *Values :* $(0, \infty)$
- *Density function :* $\frac{1}{\sigma x\sqrt{2\pi}} e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}}$
- *Expectation :* $e^{\mu+\sigma^2/2}$
- *Variance :* $e^{2\mu+2\sigma^2} - e^{2\mu+\sigma^2}$
- *Characteristic function : no closed expression*
- *Probability density function :* $f(x) = \begin{cases} \dfrac{e^{-\frac{1}{2}\left(\frac{\ln x-\mu}{\sigma}\right)^2}}{x\sigma\sqrt{2\pi}}, & x > 0; \\ 0, & \text{otherwise.} \end{cases}$

**A.3 (Multivariate normal distribution)** *A set of continuous random variables $X_1, \ldots, X_k$ are said to have a multivariate normal or k-variate normal distribution if the following holds*

- *Parameters :* $-\infty < \mu_i < \infty$ *for i=1, . . . , k*
- *Values :* $(-\infty, \infty)$
- *Density function :* $\frac{1}{\sqrt{(2\pi)^k |V|}} e^{-(x-\mu)^T V^{-1}(x-\mu)}$ *with* $x^T = (x_1, \ldots, x_k), \mu^T = (\mu_1, \ldots, \mu_k)$ *and* $V = \{Cov(X_i, X_j)\}$
- *Expectation :* $\mu$
- *Variance :* $V$
- *Characteristic function :* $e^{i\mu^T \mu - \mu^T V \mu}$ *with* $V$ *a positive semi-definite matrix*

**A.4 ($\chi^2$-distribution)** *The random variable $X$ is $\chi^2$ distributed if the following holds,* $X \sim \chi^2(\nu)$

- *Parameters :* $\nu = 1, 2, \ldots$
- *Values :* $(0, \infty)$
- *Density function :* $\frac{e^{-x/2} x^{(\nu-2)/2}}{2^{\nu/2}\Gamma(\nu/2)}$
- *Expectation :* $\nu$
- *Variance :* $2\nu$
- *Characteristic function :* $(1 - 2it)^{-\nu/2}$

**A.5 (Uniform distribution (continuous))** *The random variable $X$ is uniform (continuous) distributed if the following holds,* $X \sim U(a, b)$

- *Parameters :* $-\infty < a < b < \infty$
- *Values :* $(a, b)$
- *Density function :* $\frac{1}{b-a}$
- *Expectation :* $\frac{a+b}{2}$
- *Variance :* $\frac{(b-a)^2}{12}$
- *Characteristic function :* $\frac{e^{itb} - e^{ita}}{it(b-a)}$

**A.6 (Student $t$-distribution)** *The random variable $X$ is Student t distributed if the following holds*

- *Parameters :* $n = 1, 2, \ldots$
- *Values :* $(-\infty, \infty)$

- *Density function :* $\dfrac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\Gamma\left(\frac{n}{2}\right)\left(1+\frac{x^2}{2}\right)^{(n+1)/2}}$
  *with* $\Gamma(x) = \displaystyle\int_0^\infty e^{-t}t^{x-1}dt$
- *Expectation :* $0$ *for* $n \geq 2$
- *Variance :* $\dfrac{n}{n-2}$ *for* $n \geq 3$
- *Characteristic function :* $\dfrac{1}{\int_0^1 y^{1/2}(1-y)^{n/2-1}dy}\displaystyle\int_{-\infty}^{\infty}\frac{e^{itz\sqrt{n}}}{(1+z^2)^{(n+1)/2}}dz$

# Appendix B

# Multiple Linear Regression Analysis

In this appendix we want to give an overview of the most important definitions and notions concerning regression analysis. In the first part of this appendix some results about linear regression will be stated, in the second part we elaborate on the methods used to build and validate a regression model.

## B.1 Regression analysis

The results stated in this section can all be found in Van Berkum (2003).

### B.1.1 Properties of random vectors

Let $Y$ be a random vector with expectation $\mu$ and covariance matrix $V$. Let $A$ be a deterministic matrix and $a$ a deterministic vector. The following holds

$$\begin{aligned} \mathrm{Cov}[Y] &= \mathbb{E}\left[(Y-\mathbb{E}[Y])(Y-\mathbb{E}[Y])^T\right], \\ \mathbb{E}[a^TY] &= a^T\mu, \\ \mathbb{E}[AY] &= A\mu, \\ \mathrm{Cov}[a^TY] &= a^TVa, \\ \mathrm{Cov}[AY] &= AVA^T, \\ \mathbb{E}[Y^TAY] &= \mathrm{tr}[AV]+\mu^TA\mu. \end{aligned} \tag{B.1}$$

For an $n \times n$ matrix it holds that $tr(A) = \sum_{i=1}^{n} a_{ii}$.

### B.1.2 Models and estimation

First we will introduce the linear regression model. This model will be

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon, \tag{B.2}$$

where $\varepsilon$ is a random variable with expectation $\mathbb{E}[\varepsilon] = 0$ and variance $\mathrm{Var}[\varepsilon] = \sigma^2$.

We assume that $n$ observation are made. For each of these observation the model holds, so

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \ldots + \beta_k x_{1k} + \varepsilon_1, \\ Y_2 &= \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \ldots + \beta_k x_{2k} + \varepsilon_2, \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \ldots + \beta_k x_{nk} + \varepsilon_n, \end{aligned} \tag{B.3}$$

where we assume that all $\varepsilon_i$ are independent.

We simplify this by using a different notation, using matrices. In this matrix notation the model becomes

$$Y = X\beta + \varepsilon, \tag{B.4}$$

with

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} \quad \text{and} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

If we want to estimate $\beta$ and we use least squares estimators, we get

$$\widehat{\beta} = (X^T X)^{-1} X^T Y. \tag{B.5}$$

This estimator for $\beta$ is unbiased. This can be shown easily using one of the properties of (B.1),

$$\mathbb{E}[\widehat{\beta}] = \mathbb{E}[(X^T X)^{-1} X^T Y] = (X^T X)^{-1} X^T \, \mathbb{E}[Y] = (X^T X)^{-1} X^T X \beta = \beta.$$

The covariance matrix of $\widehat{\beta}$ is

$$\begin{aligned} \mathrm{Cov}[\widehat{\beta}] &= \mathbb{E}\left[(\widehat{\beta} - \mathbb{E}[\widehat{\beta}])(\widehat{\beta} - \mathbb{E}[\widehat{\beta}])^T\right] \\ &= \mathbb{E}\left[\left((X^T X)^{-1} X^T Y - \mathbb{E}[(X^T X)^{-1} X^T Y]\right)\left((X^T X)^{-1} X^T Y - \mathbb{E}[(X^T X)^{-1} X^T Y]\right)^T\right] \\ &= (X^T X)^{-1} X^T \cdot \mathbb{E}\left[(Y - \mathbb{E}[Y])(Y - \mathbb{E}[Y])^T\right] \cdot ((X^T X)^{-1} X^T)^T \\ &= (X^T X)^{-1} X^T \cdot \sigma^2 I \cdot ((X^T X)^{-1} X^T)^T \\ &= (X^T X)^{-1} \sigma^2 \cdot X^T X (X^T X)^{-1} \\ &= (X^T X)^{-1} \sigma^2. \end{aligned} \tag{B.6}$$

The maximum likelihood estimators of $\widehat{\beta}$ and $\widehat{\sigma}^2$ are the same as the ordinary least squares estimators under the assumption that $\varepsilon \sim N(0, \sigma^2)$.

## B.2 Building the model

In this section we look at the methods used to build a multiple regression model. The information in the following sections can be found in Montgomery et al. (2001)

### B.2.1 Choosing initial models

Firstly we have to choose possible models. The most commonly used method is looking at the model with only the regressor variables $(x_1, ..., x_k)$ and the models containing also second order interactions (e.g. $x_1x_3$ is a second order interaction). Next we check for significance of the regressors using forward, backward or stepwise regression. Regressors that are not significant will be eliminated from the model. For more details we refer to Montgomery et al. (2001). After doing this we end up with a certain number of possible models. From these models we have to choose the models that fits our wishes the best. This can be done using the following techniques.

### B.2.2 Multicollinearity

A serious problem that may have great impact on the usefulness of a regression model is multicollinearity among the regressor variables. An important multicollinearity diagnostic are the variance inflation factors ($VIF$'s). In general the variance inflation factor for the $j$th regression can be written as

$$VIF_j = \frac{1}{1 - R_j^2}, \tag{B.7}$$

where $R_j^2$ is the coefficient of multiple determination obtained from regressing $x_j$ on the other variables. The calculation of the coefficient of multiple determination will be given later on. If $x_j$ is nearly linear dependent on some of the other regressors, so multicollinearity occurs, then $R_j^2$ will be almost 1 and $VIF_j$ will be large. Hence, we can eliminate models where the $VIF$'s are large.

This step is often considered a model validation technique, we already perform it here because it reduces the number of regression models we have to consider.

### B.2.3 Model adequacy checking

Before we can investigate which of the remaining models is the best, it is useful to remember what the major assumptions are in our regression analysis.

1. The relationship between the response $y$ and the regressors is linear, at least approximately.
2. The error term $\varepsilon$ has zero mean.
3. The error term $\varepsilon$ has constant variance $\sigma^2$
4. The errors are uncorrelated.
5. The errors are normally distributed.

Taken together, assumptions 4 and 5 imply that the errors are independent random variables and assumptions 5 is required for hypothesis testing and interval estimation. We should not assume that our model does satisfy the above assumptions. Therefore we do the following tests.

**Plot of the residuals against the fitted values**

It is useful to plot the residuals, $e_i = y_i - \widehat{y}_i$, against the corresponding fitted values. If the residuals can be contained in a horizontal band, then there are no obvious model defects.

**Normal probability plot**

Big violations of the normality assumption (assumption 5) are very serious because the $t$ and $F$ statistics (see B.2.6) and confidence and prediction intervals depend on the normality assumption. We can use the normal probability plot (also called quantile-quantile plot) to check the normality assumption. This is a plot designed so that the cumulative normal distribution will plot a straight line. Let $e_{[1]} < e_{[2]} < ... < e_{[n]}$ be the residuals in increasing order. If we plot $e_{[i]}$ against the cumulative probability $P_i = (i - \frac{1}{2})/n$, $i = 1, 2, ..., n$, on the normal probability plot, the resulting points should lie approximately on a straight line.

### B.2.4 Overall adequacy of the model

By means of $R^2$ and $R^2_{adj}$ we can asses the overall adequacy of the model. We call $R^2$ also the coefficient of determination and compute it as follows

$$R^2 = \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{y})}{\sum_{i=1}^{n}(y_i - \overline{y})}. \tag{B.8}$$

$R^2$ is often referred to as the proportion of variation explained by the regressors. Because of the computation of $R^2$, $0 \leq R^2 \leq 1$. Most of the variability of $y$ is explained by the regression model if $R^2$ is close to 1. The quantity $R^2$ always increases if a regressor is added to the model, so it is difficult to judge whether one model is better than another. Hence, $R^2 \approx 1$ is a necessary condition but not sufficient. Therefore some regression model builders prefer to use the adjusted $R^2$

$$R^2_{adj} = 1 - \frac{SS_{Res}/(n-p)}{SS_T/(n-1)} \tag{B.9}$$

with $SS_{Res} = \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$, $SS_T = \sum_{i=1}^{n} y_i^2 - \frac{(\sum_{i=1}^{n} y_i)^2}{n}$, $n$ is the number of observations and $p$ is the number of regressors + 1.

When choosing the best model it is wise to choose the model with an $R^2_{adj}$ that is closest to 1.

### B.2.5 Estimation of $\sigma$

The estimator of $\sigma^2$ is $\widehat{\sigma}^2$. This estimator is model dependent. We can compute $\widehat{\sigma}^2$ as follows

$$\widehat{\sigma}^2 = \frac{SS_{Res}}{n-p}. \tag{B.10}$$

Because we wish to have a small $SS_{Res}$, a model with a small value for $\widehat{\sigma}^2$ so also for $\widehat{\sigma}$ is preferable.

### B.2.6 $F$-test and $t$-test

Next we are going to test the significance of regression of the model and look at which specific regressors seem important. We may use the $F$ and $t$ statistics because we already checked the normality assumption.

#### Test for significance of regression

The test for significance of regression is a test to determine if there is a linear relationship between the response $y$ and any of the regressor variables $x_1, x_2, ..., x_k$. This procedure is often thought of as an overall test of model adequacy. The hypotheses are

$$\begin{aligned} H_0 &: \beta_0 = \beta_1 = ... = \beta_k = 0 \\ H_1 &: \beta_j \neq 0 \qquad \text{for at least one } j. \end{aligned}$$

Rejection of the null hypothesis implies that at least one of the regressors $x_1, x_2, ..., x_k$ contributes significantly to the model. To test these hypotheses we use the following test statistic

$$F_0 = \frac{SS_R/k}{SS_{Res}/(n-p)} \tag{B.11}$$

with $SS_R = SS_T - SS_{Res}$ the regression sum of squares. The test statistic $F_0$ follows a $F_{k,n-p}$ distribution under $H_0$. We can reject $H_0$ if $F_0 > F_{\alpha,k,n-p}$, or when the $P$-value is very small.

#### Test on individual regression coefficients

Now we have determined that at least one of the regressors in our models is important, the question is now which one(s). The hypotheses for testing the significance of an individual regression coefficient are

$$\begin{aligned} H_0 &: \beta_j = 0 \\ H_1 &: \beta_j \neq 0. \end{aligned}$$

If $H_0$ is not rejected, then this indicates that the regressor $x_j$ can be removed from the regression model. To test this hypothesis we use the following test statistic

$$t_0 = \frac{\widehat{\beta}_j}{\sqrt{\widehat{\sigma}^2 C_{ij}}} \tag{B.12}$$

with $C_{jj}$ the diagonal element of $(X^T X)^{-1}$. $H_0$ is rejected if $|t_0| > t_{\alpha/2,n-k-1}$.

## B.3 Model validation

When we have build the model it seems wise to test whether or not the model does what it is built for. The goal of our research is making a prediction model. To test the predictive performance we have to use new data, so not the data that was used for building the model, we call this data the prediction data. The data used to build the model is called the construction data.

The quality of the the model(s) is evaluated by means of the mean prediction error ($ME$), the square root of the mean square prediction error ($\sqrt{MSE}$) and the percentage of variability

in the new data explained by the model ($R^2_{pred}$). These three statistics are calculated using the prediction data. Before we can compute these statistics we have to calculate the estimated $y_i$ using the prediction data and the models that we have fitted.

### B.3.1 Mean Prediction Error

Obviously we would like that the mean prediction error ($ME$) is very close to zero. We define $ME$ as follows

$$ME = \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)}{n-p}. \tag{B.13}$$

A very large mean prediction error means that the prediction errors are large, so the model does not give good prediction for the new data.

### B.3.2 Square root of the mean square prediction error

The square root of the mean square prediction error is computed as follows

$$\sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{n-p}}. \tag{B.14}$$

This calculation is very much the same as the calculation of $\widehat{\sigma}$ with the construction data. Hence, we compare the value af $\sqrt{MSE}$ with the value of $\widehat{\sigma}$.

### B.3.3 Percentage of variability explained by the model

This statistic shows how much this model is expected to explain of the variability in predicting new observations. $R^2_{pred}$ can be computed as follows

$$R^2_{pred} = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{B.15}$$

where $n$ = the number of observations.
It is desirable that the $R^2_{pred}$ of the testing data does not vary a lot with the $R^2_{adj}$.

# Appendix C

# R Manual

We used the computer software program R to obtain the results in the chapter about the building of the regression model. R can be freely downloaded on www.r-project.org. In the first part of this appendix we give some information on the use of R and in the second part some of the code that was used for calculations in this report can be found.

## C.1 Packages

R contains several packages. Loading a package can be done the following way:

Packages → Load package → Select one → OK

It can also be the case that a package that is not contained standard in R is needed. Then the following should be done:

Packages → Install package(s) → Select one → OK

We can now choose a package that we want to be contained standard in R. If a command is not contained in a standard package in R the help function will not find it, therefore it is wise to also scan the internet for the command.

## C.2 The help function

There are two ways of finding information about a subject in R:

- by typing a question mark before the command that is wanted, e.g. ?pnorm
- by typing the term that is wanted between brackets after help.search, e.g. help.search("normal distribution")

## C.3 Reading data

Data can be read from file and from the internet.

- from file : read.table("file.txt", header = TRUE)
- from internet: read.table("URL", header = TRUE)

When we want R to use a data set, we save this set in a file with .txt extention (e.g. winedt, notepad). Also in the name of the file every \ should be replaced by three \'s. Attention should be paid on how the numbers are written, R does recognized only a point in a number (e.g. $\pi = 3.14$ and not 3,14). With the command str(data) one can check whether or not the data is imported correctly.

## C.4 Regression analysis in R

A few commands in R that can be useful for regression analysis in R. Between brackets the package in which the functions can be found is given.

- lm: is used to fit linear regression models (stats)
- summary: summary method for the class lm (stats)
- anova: computes an analysis of variance table for one or more linear model fits (stats)
- predict: predicted values based on linear model object (stats)
- qqnorm: quantile-quantile plot= normal probability plot (stats)
- qqline: adds a line to a quantile-quantile plot which passes through the first and third quartiles (stats)
- vif: variance inflation factor (car)

## C.5 Working with matrices in R

Before starting to work with matrices in R, the package "base" should be loaded. Most of the commands that can be applied on matrices run in this package. Some useful commands are the following.

- matrix: creates a matrix from the given set of values (base)
- as.matrix: attempts to turn its argument into a matrix, very useful when reading data from a txt-file (base)
- %*%: matrix multiplication (base)
- t: transpose (base)
- solve: gives the inverse of a matrix (base)
- det: gives the determinant of a matrix (base)

## C.6 R code

In this section there will be some examples of code that were used to obtain results in this report.

## C.7 Building the model

Firstly the data has to be imported into R. It is always useful to check with str whether this is done correctly (Figure C.1). After that the we fit the model, i.e model $A$ (Figure C.2). In

```
> datac<-read.table("D:\\\My Documents\\\\\\afstuderen\\\\\\R\\\\\\constr.txt", header=TRUE)
> str(datac)
`data.frame':   79750 obs. of  7 variables:
 $ inter: num  0.959 0.872 0.936 0.883 0.921 ...
 $ no   : num  6 6 6 6 6 6 6 6 6 6 ...
 $ ol   : num  0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 ...
 $ icv  : num  0.259 0.259 0.259 0.259 0.259 ...
 $ cv   : num  0.452 0.452 0.452 0.452 0.452 ...
 $ rho  : num  0.87 0.87 0.87 0.87 0.87 0.87 0.87 0.87 0.87 0.87 ...
 $ nj   : int  30 30 30 30 30 30 30 30 30 30 ...
```

Figure C.1: Importing the data points and checking dimensions.

```
> fitA<-lm(inter~rho+no+ol+icv+nj+cv, data=datac)
```

Figure C.2: Fitting the model to the data.

order to calculate the $VIF$'s we have to import the package car before asking R for the values (Figure C.3). In Figure C.4 the code to make a residual plot is given and in Figure C.5 the

```
> local({pkg <- select.list(sort(.packages(all.available = TRUE)))
+ if(nchar(pkg)) library(pkg, character.only=TRUE)})
> vif(fitA)
     rho       no       ol      icv       nj       cv
1.105049 1.128058 1.144591 2.224798 1.041008 2.203275
```

Figure C.3: Variance inflation factors of model $A$.

```
> plot(fitted(fitA),resid(fitA))
> lines(lowess(fitted(fitA),resid(fitA)))
> abline(h=0)
```

Figure C.4: Plot of the residuals.

transformation of the response variable is shown. After that the regression model has to be

```
> datac$inter<-log(datac$inter)
```

Figure C.5: Transforming the response variable.

fitted again using the transformed data and then the ANOVA-table (analysis of variance) can

```
> summary(fitA)

Call:
lm(formula = inter ~ rho + no + ol + icv + nj + cv, data = datac)

Residuals:
      Min         1Q     Median         3Q        Max
-0.517683 -0.069385  0.001694  0.071322  0.520278

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.984e+00  9.411e-03 -210.85   <2e-16 ***
rho          1.466e+00  5.195e-03  282.28   <2e-16 ***
no           1.109e-01  1.114e-03   99.59   <2e-16 ***
ol          -1.298e-01  1.030e-02  -12.60   <2e-16 ***
icv         -8.832e-01  4.237e-03 -208.43   <2e-16 ***
nj          -2.604e-03  3.277e-05  -79.48   <2e-16 ***
cv           9.111e-01  2.619e-03  347.90   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1071 on 79743 degrees of freedom
Multiple R-Squared: 0.7653,     Adjusted R-squared: 0.7653
F-statistic: 4.334e+04 on 6 and 79743 DF,  p-value: < 2.2e-16
```

Figure C.6: The analysis of variance table of model *A*.

be made (Figure C.6). Finally the command to generate the normal probability plot is given in Figure C.7.

```
> qqnorm(resid(fitA))
> qqline(resid(fitA))
```

Figure C.7: The normal probability plot.

# Appendix D

# Mathematica Code

In this appendix the Mathematica code that is made and used during this thesis can be found.

## D.1 Data-splitting: DUPLEX

In Chapter 4, the DUPLEX-algorithm, a tool to split data into two parts, is explained. In this section the Mathematica code used to run this algorithm and the explanation of the code can be found.

The DUPLEX-algorithm is implemented in the following two modules duplex1 and duplex2.

```
\!\(duplex1[X_] :=
    Module[{i, j, k, Z, T, B,
        W}, \[IndentingNewLine]s[
          j_] := \@\(\[Sum]\+\(i = \
1\)\%\(Length[X]\)\(\(\(X[\([i]\)]\)[\([j]\)] - \(\[Sum]\+\(k = \
1\)\%\(Length[X]\)\(X[\([k]\)]\)[\([j]\)]\)\/Length[X]\)\^2\); \
\[IndentingNewLine]z[i_,
          j_] := \(\(X[\([i]\)]\)[\([j]\)] - \(\[Sum]\+\(k = 1\)\%\(Length[X]\
\)\(X[\([k]\)]\)[\([j]\)]\)\/Length[X]\)\/s[j]; \[IndentingNewLine]For[
        Z = {}; i = 1, i <= Length[X], \(i++\),
        Z = Join[
            Z, {Table[
                z[i, j], {j, 1,
                  Length[Transpose[X]]}]}]]; \[IndentingNewLine]B =
        Dot[Transpose[Z], Z]; \[IndentingNewLine]T =
        CholeskyDecomposition[B]; \[IndentingNewLine]W =
        Dot[Z, Inverse[T]]; \[IndentingNewLine]W]\)

duplex2[Y_] :=
    Module[{n = Length[Y], afstanden, v1, v2, overig, nieuweafstanden,
        nieuweoverig, afstandenrestmet1, afstandenrestmet2, minpos, naarv1,
        naarv2}, afstanden = Outer[(#1 - #2).(#1 - #2) &, Y, Y, 1];
      v1 = Position[afstanden, Max[afstanden]][[1]];
      overig = Complement[Range[n], v1];
```

```
nieuweafstanden = afstanden[[overig, overig]];
v2 = overig[[Position[nieuweafstanden, Max[nieuweafstanden]][[1]]]];
nieuweoverig = Complement[overig, v2];
While[nieuweoverig =!= {},
  afstandenrestmet1 =
    Outer[(#1 - #2).(#1 - #2) &, Y[[nieuweoverig]], Y[[v1]], 1];
  minpos = Ordering[Min /@ afstandenrestmet1, -1][[1]];
  naarv1 = nieuweoverig[[minpos]];
  nieuweoverig = Drop[nieuweoverig, {minpos}];
  v1 = Append[v1, naarv1];
  If[nieuweoverig =!= {},
    afstandenrestmet2 =
      Outer[(#1 - #2).(#1 - #2) &, Y[[nieuweoverig]], Y[[v2]], 1];
    minpos = Ordering[Min /@ afstandenrestmet2, -1][[1]];
    naarv2 = nieuweoverig[[minpos]];
    nieuweoverig = Drop[nieuweoverig, {minpos}];
    v2 = Append[v2, naarv2]]];
{v1, v2}];
```

### D.1.1 Explanation of the code

The DUPLEX-algorithm has two main parts, we have chosen to make a module for every part. In duplex1 the matrix $X$ with the data points is orthonormalized to a matrix $W$. In duplex2 the points from $W$ are assigned into two sets.

**Input duplex1:**
The list $X$ of data points.
**Output duplex1:**
The list $W$ of orthonormalized data points

**Input duplex2:**
The list $W$ of orthonormalized data points.
**Output duplex2:**
Two lists of the labels of the points, one list for the construction data and one list for the prediction data.

To run the DUPLEX-algorithm on a matrix $X$, use the following

```
duplex2[duplex1[X]]
```

## D.2 Selectivity in the regression policy

In this section the program we used to identify the selectivity is shown.

```
<< Statistics`ContinuousDistributions`

selectivity[z_, \[Sigma]_, T_, rep_] :=
  Module[{x, yest, ndist, \[CurlyEpsilon], interest, intertrue, makespanest,
```

```
  makespantrue, geschat, echt, dummylijst, \[Mu], i, y, difference = {},
  accept = {}, makespansa, kwantiel, acceptance},
For[i = 1, i <= Length[z], i++, x = z[[i]];
  ytrue = -1.984 + 0.111x[[3]] - 0.130x[[4]] - 0.883x[[5]] +
      0.991x[[6]] + 1.466x[[7]] - 0.003x[[8]];
  ndist = NormalDistribution[0, \[Sigma]];
  \[CurlyEpsilon] = RandomArray[ndist, rep];
  yest = Map[# + ytrue &, \[CurlyEpsilon]];
  interest = Exp[yest - (\[Sigma]^2)/2];
  intertrue = Exp[ytrue];
  lowerb = x[[2]];
  makespanest = (1 + interest)*lowerb;
  makespantrue = (1 + intertrue)*lowerb;
  kwantiel = Exp[yest + 1.644873*\[Sigma]];
  geschat = {};
  echt = {};
  If[makespantrue < T, echt = Append[echt, makespantrue]];
  geschat =
    Complement[
      Table[If[((kwantiel[[i]] + 1)*lowerb) < T, makespanest[[i]], y], {i,
           1, Length[makespanest]}], {y}];
  acceptatie = N[Length[geschat]/rep];
  \[Mu] = Mean[geschat];
  accept = Join[accept, {acceptatie}];
  difference =
    If[geschat == {}, biastrue = {no value},
      Join[difference, {\[Mu] - makespantrue}]]
  ];
Print["difference = ", difference];
Print["acceptance = ", accept]]
```

### D.2.1 Explanation of the code

**Input:**

- $z = ((x^1), (x^2), \ldots, (x^k))$ with $x^i = (x_1^i, x_2^i, \ldots, x_8^i)$ where $(x_3^i, \ldots, x_8^i) = (\mu_s, \mu_g, cv^2_{\mathbb{E}[p]}, cv_p^2, \rho_{\max}, n_J)$ are the job set characteristics of the job set $J^i$, $x_1^i$ is the makespan of the job set found by the SA-algorithm and $x_2^i$ is the workload of the job set.
- $\sigma$ = the estimated value of $\sigma$ from the regression model.
- $T$ = the length of the planning period.
- *rep* = the number of times the estimated value for the makespan is generated.

**Output:**

- *difference*: the difference between the mean of the accepted estimated makespans and the true makespan.

- *acceptance*: the number of estimated makespans that are accepted by the regression policy, the acceptance rate.

For every set of job set characteristics $x^i, i = 1, ..., k$ the above values are calculated. Therefore if for example $k = 5$ (see input) the algorithm gives a list of two tables of each five values.

# Index


$F$-test, 63
$R^2$, 62
$R^2_{adj}$, 62
$R^2_{pred}$, 64
$SS_T$, 62
$SS_{Res}$, 62
$t$-test, 63

accuracy of errors, 26
asymptotically unbiased estimator, 45
average number of processing steps, 20
average overlap of processing steps , 20

batch process, 11
bias, 46
biascorrection, 50
bottleneck, 17

Cholesky-decomposition, 28
construction data, 21, 27

data-splitting, 27
Duplex, 28
dynamic, 33

ECU, 34
estimation errors, 25, 26
expected value, 44

flow process, 11

interaction margin, 17
Invariance property, 18

job, 15
job characteristics, 15, 16

lognormal distribution, 45

makespan, 15, 17, 46
maximum likelihood estimator, 18
mean prediction error, 64
model adequacy checking, 23, 61
multicollinearity, 61
multiproduct, 11
multipurpose, 12
multivariate normal distribution, 18

Newton's Binomial Formula, 44
normal probability plot, 24, 62
number of jobs in the job set, 20

order, 15
order acceptance, 15
ordinary least squares estimator, 18
orthonormalizing, 28
overall adequacy of the model, 62

PCME, 34
planning period, 15, 33
plot of the residuals against the fitted values, 62
precision of errors, 26
prediction data, 21, 27

regression analysis, 17, 59
regression policy, 33
routing, 12

selectivity, 34
simulated annealing algorithm, 13
square root of the mean square prediction error, 64
squared coefficient of variation, 21
standardizing, 28

upper prediction bound, 34

variation indicated by the dissimilarity of the processing steps, 20

waiting time restrictions, 12
workload, 21

# Bibliography


L.J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics*. Duxbury, 1992.

V.C. Ivanescu. *Order Acceptance Under Uncertainty in Batch Process Industries*. PhD thesis, Technische Universiteit Eindhoven, 2004.

R.W. Kennard and L.A. Stone. Computer aided design of experiments. *Technometrics*, 11 (1):137–148, 1969.

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

D.C. Montgomery, E.A. Peck, and G.G. Vining. *Introduction to Linear Regression Analysis*. Wiley, 2001.

W.H.M. Raaymakers. *Order Acceptance and Capacity Loading in Batch Process Industries*. PhD thesis, Technische Universiteit Eindhoven, 1999.

W.H.M. Raaymakers and J.A. Hoogeveen. Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing. *European Journal of Operational Research*, 126:131–151, 2000.

R.D. Snee. Computer aided design of experiments. *Technometrics*, 19(4):415–428, 1977.

E.E.M. Van Berkum. Regressie en variantie analyse. Lecture notes, 2003.