Methodology development for identifying improvement opportunities using discrete simulation modelling

LA Haasbroek

orcid.org/ 0000-0002-3076-8964

Dissertation accepted in fulfilment of the requirements for the degree *Master of Engineering in Industrial Engineering* at the North-West University

Supervisor: Dr PvZ Venter

Co-supervisor: Dr SC van Niekerk

Graduation: August 2024

“Ke Nako!”

- A

Acknowledgements

The following institutions and individuals are acknowledged for their various contributions and support towards the completion of this dissertation:

- Dr. PvZ Venter, at the North-West University, and Dr. SC van Niekerk, for their continuous assistance and guidance with the completion of this dissertation.
- The steel manufacturing company in which the study was conducted not only offered the author valuable insights into the dynamics regarding the research but also facilitated access to the Simio Simulation Software. Furthermore, the personnel at the Central Workshops, their willingness to assist is highly appreciated.
- Marno Haasbroek, my amazing husband, for his unconditional love and continued support. His patience and encouragement made it possible to attempt and overcome this daunting task.
- Johan and Marié Ackerman, my two wonderful parents, for their ongoing love and motivation. They provided a significant support structure which assisted me in completing this task.
- Individuals and friends who gave insightful comments, my twin sister and older brother who know how to motivate in their unique way and any other relevant party who guided and assisted me.

Finally and most importantly, I thank God for the ability He bestowed upon me to develop, learn and grow.

Abstract

The Central Workshops' main responsibility, within an integrated steel manufacturer, is to manufacture and recondition plant-related equipment. In recent years, the Central Workshops have been reorganised and down-scaled on various occasions. This was done without adequate change management, leading to operational deficiencies with long-term and immediate consequences. These consequences include increased manufacturing costs, not adhering to deadlines and not being profitable, which could lead to the closure of the Central Workshops.

Improvement opportunities must be identified to ensure the Central Workshops' profitability. Due to the unpredictability of job shop operations, and the fluctuations in production demand, this task is challenging. The unpredictability stems from the frequent arrival of varied brake-down orders from integral plants within the steel manufacturer. These challenges complicate the identification of improvement opportunities. A strategic approach to addressing this complexity entails initially focusing on one specific job.

A methodology is developed that aims at identifying improvement opportunities. By addressing bottlenecks and inefficiencies with the use of the developed methodology, operations can be enhanced, enabling effective operations and increasing customer satisfaction, which will in turn have a positive financial impact.

The developed methodology is built on a sequence of steps, commencing with an understanding of the system under consideration. The required data is identified, cleansed, and subjected to analysis through the use of relevant statistical analysis tools. Distribution functions are identified that reflect the real-world operations of the system under consideration. The distribution fits are evaluated by utilising Exploratory Data Analysis (EDA) techniques to establish a hypothesis and evaluate possible distribution fits through the use of informative graphs. The relevant hypothesis testing methods are used to formalise the interpretation of the 'ideal' distribution fits.

Thereafter, discrete simulation modelling software is used to develop models that reflect the operation under consideration. Firstly, the current state is modelled, whereafter an iterative process is followed to identify bottlenecks within the system and consequently identify improvement opportunities. These bottlenecks are identified through the application of the Theory of Constraints (TOC) principles and a thorough assessment of workstation utilisation. The utilisation result not only provides insights into the system's operational efficiency but also serves as a fundamental performance measure. To further evaluate the iterative model development, additional key performance measures, namely throughput and production costs, were employed. The iterative process continued until satisfactory results were achieved. With effective and adequate results obtained, the proposed methodology is identified as correct and adequate. Although the methodology is developed to analyse a single job within the job shop environment, there is a promising avenue for future research to extend its applicability.

Keywords: Methodology development; Improvement opportunities; Data analysis; Discrete simulation modelling; Simio

Contents

Acknowledgements i

Abstract ii

List of Figures vi

List of Tables ix

Acronyms xi

1 Research Overview 1
- 1.1 Background and Rationale . . . 1
  - 1.1.1 Central Workshops . . . 2
- 1.2 Research Scope . . . 2
  - 1.2.1 Problem Statement . . . 2
  - 1.2.2 The Aim . . . 2
  - 1.2.3 Objectives . . . 2
- 1.3 Research Approach . . . 2
- 1.4 Chapter Division . . . 3

2 System Background 4
- 2.1 Mechanical Fabrication Job Shop . . . 4
  - 2.1.1 Floor Layout . . . 4
  - 2.1.2 Machine Allocations . . . 5
  - 2.1.3 Workforce . . . 5
- 2.2 Pipe Manufacturing . . . 6
  - 2.2.1 Decision to Consider a Specific Job . . . 7
  - 2.2.2 Manufactured Pipes Overview . . . 7
  - 2.2.3 The Process . . . 7
  - 2.2.4 Flow of Materials . . . 16
- 2.3 Concluding Remarks: System Background . . . 18

3 Literature Review 19
- 3.1 Process Heterogeneity . . . 19
- 3.2 Applicability of Modelling Approaches . . . 19
- 3.3 Systems and Models . . . 20
- 3.4 Simulation Modelling . . . 21
  - 3.4.1 Simulation Modelling Approaches . . . 22
  - 3.4.2 Simio Simulation Software . . . 23
  - 3.4.3 The Simulation Process . . . 23
- 3.5 Probability Distribution Functions and Their Properties . . . 24
  - 3.5.1 Normal Distribution . . . 25
  - 3.5.2 Uniform Distribution . . . 25
  - 3.5.3 Exponential Distribution . . . 26

3.5.4 Logistic Distribution . . . 26
3.5.5 Beta Distribution . . . 26
3.5.6 Lognormal Distribution . . . 27
3.5.7 Gamma Distribution . . . 28
3.5.8 Weibull Distribution . . . 28
3.6 Randomness in Simulation and Random Variables . . . 29
3.7 Theory of Constraints . . . 29
3.8 Concluding Remarks: Literature Review . . . 30

4 Data Analysis **32**
4.1 The Raw Data . . . 32
4.1.1 "Pre-allocated" and "Booked" Processing Times . . . 32
4.2 Exploratory Data Analysis . . . 33
4.2.1 No Distribution Fit . . . 38
4.2.2 Distribution Data . . . 39
4.3 Concluding Remarks: Data Analysis . . . 40

5 Model Development **42**
5.1 Fundamentals of Model Development in Simio . . . 42
5.1.1 Entities and Attributes . . . 42
5.1.2 Object Libraries . . . 42
5.2 Model-0.0: Current State . . . 43
5.2.1 Floor Layout Visualisation . . . 44
5.2.2 Workstations' Properties . . . 44
5.2.3 Machine Reliability Metrics . . . 53
5.2.4 Networks within Simio . . . 53
5.2.5 Symbol Transformation . . . 54
5.2.6 Material Handling: Workers . . . 55
5.2.7 Material Handling: Transferring of Entities . . . 57
5.2.8 Simulation Run-Time . . . 61
5.2.9 Object Flow Diagram . . . 62
5.3 Model Verification and Validation . . . 64
5.3.1 Verification . . . 64
5.3.2 Validation . . . 65
5.4 Concluding Remarks: Model Development . . . 70

6 Improvement Opportunities Analysis **71**
6.1 Improved Process Flow . . . 71
6.1.1 From-to-Chart . . . 72
6.1.2 Evaluating the Current State and Identifying a Proposed State . . . 74
6.1.3 Transportation Distance between Consecutive Workstations . . . 77
6.1.4 Implementation of Improved Process Flow . . . 78
6.2 Improvement Opportunities Analysis . . . 79
6.2.1 Theory of Constraints . . . 79
6.2.2 Performance Measures . . . 79
6.2.3 Model-1.0 . . . 80
6.2.4 Model-2.0 . . . 84
6.2.5 Model-3.0 . . . 93
6.2.6 Model-4.0 . . . 100
6.3 Concluding Remarks: Improvement Opportunities Analysis . . . 101

7 Conclusion and Recommendations **104**
7.1 Study Overview . . . 104
7.2 Future Research . . . 105

Bibliography **106**

Appendices **110**

A Validate Pipe Manufacturing Operation **110**
A.1 Throughput . . . 110
A.2 Cost Comparison . . . 110

B Mechanical Fabrication Job Shop **112**
B.1 Floor Layouts . . . 112
B.2 Sections in Job Shop . . . 114
B.3 Machine Allocations . . . 115
B.4 Machine Visualisation . . . 117
B.5 Workforce Allocations . . . 117

C Plate Specifications **119**

D Details Regarding the Pipe Manufacturing Process **122**
D.1 Straight Line Cutting . . . 122
D.2 Deburring of Plates . . . 123
D.3 Roll . . . 124
D.4 Long-weld . . . 125
D.5 Inspection . . . 126

E Transportation Methods **127**

F Probability Distribution Functions **128**

G Data Analysis **131**
G.1 Exploratory Data Analysis Plots . . . 131
G.2 Coding Done in RStudio . . . 137

H Model Development **181**
H.1 Simio Library Objects . . . 181

I Process Development within Simio **183**

J Validate Model-0.0 **189**

K Opening in Wall **190**

L Developed Model-1.0 **191**

M Plagiarism Declaration **192**

List of Figures

2.1 Schematic Floor Layout of Mechanical Fabrication Job Shop . . . 4
2.2 Symbols that Reflect Types of Machines . . . 5
2.3 Schematic Floor Layout of Mechanical Fabrication Job Shop Depicting Allocation of Machine Types . . . 5
2.4 Workforce within Mechanical Fabrication (MechFab) . . . 6
2.5 Demonstration of 12-metre, 1000 Nominal Bore (NB) pipe . . . 7
2.6 Pipe Manufacturing Flowchart . . . 8
2.7 Schematic Representation with Work-Area Allocations . . . 8
2.8 Transportation Mode Between Workstations . . . 9
2.9 Material Transferred from Plate Store to Bay 1 . . . 10
2.10 Material transferred from trolley to straight line cutting workstation . . . 10
2.11 Material Transferred from *s_line* workstation to *dress* workstation . . . 11
2.12 Material Transferred from *dress* workstation to *roll* workstation . . . 11
2.13 Material Transferred from *roll* workstation to *l_weld* workstation . . . 12
2.14 Material Flow within Elektriska Svetsnings-Aktiebolaget (ESAB)-room . . . 13
2.15 Decision Representation after *l_weld* Workstation . . . 13
2.16 Material transferred to *assem* workstation . . . 14
2.17 Placement of Canisters to Weld Seams Accordingly . . . 15
2.18 12-Metre Pipe is Transferred to *inspect* Workstation . . . 15
2.19 12-Metre Pipe is Transferred to *disp* Workstation . . . 16
2.20 Flow of Materials . . . 17
2.21 Improved Flow Presumed Possibility . . . 17

3.1 The Basic Simulation Process [1] . . . 23

4.1 Histogram of *s_line* Operation's Processing Time . . . 34
4.2 Empirical density- and cumulative distribution plot of *s_line* Operation's Processing Time 35
4.3 Cullen and Frey graph of Straight line cutting . . . 36
4.4 Histogram and theoretical densities plot for *s_line* Operation's Processing Times . . . 36
4.5 Q-Q plot for Straight Line Cutting Operation's Processing Times . . . 37
4.6 Fitted Distribution of Processing Time for the *s_line* Workstation . . . 38

5.1 Process Related to Inventory Replenishment - *PlatesInventory_OnReplenishmentOrder* 46
5.2 Process that Determines Roll or Re-roll Operation at *roll* Workstation - *roll_Processing* 46
5.3 Schematic Representation of Process that Determines Roll or Re-roll Operation at *roll* Workstation . . . 47
5.4 assem and c_weld Setup within Simio . . . 48
5.5 Combiner Object's Node Clarification . . . 49
5.6 Schematic Representation of Operations at *assem-* and *c_weld* Workstations . . . 49
5.7 Representation of Process at *assem_transnode* - *assem_transnode_Entered* Process . . . 50
5.8 Process Defined as *assem_Processing* . . . 51
5.9 Process Defined as *c_weld_Processing* . . . 52
5.10 Process on Exiting assem workstation . . . 55

5.11 Lunchroom Position . . . 56
5.12 Worker's Work Schedule . . . 57
5.13 Movement of Vehicle Object . . . 58
5.14 Identification of 'Home' nodes for Vehicle Objects . . . 59
5.15 Movement of Crane 1 . . . 60
5.16 Movement of Crane 2 . . . 60
5.17 Movement of Crane 3 . . . 61
5.18 Movement of Crane 4 . . . 61
5.19 OFD of Current State . . . 63
5.20 Validating *dress* Workstation's Processing Time at Input Node . . . 68
5.21 Validating *dress* Workstation's Processing Time at Output Node . . . 68
5.22 Section of Gantt Chart . . . 68
5.23 Representation of Alternative *assem* and *c_weld* Workstations . . . 69
5.24 Original Setup of *assem* and *c_wled* Workstations . . . 69

6.1 Activity Relationship Chart . . . 74
6.2 Path Numbering Associated with Distance Travelled . . . 75
6.3 Current and Proposed Workstation Allocations' Spaghetti Diagrams . . . 75
6.4 Improved State Modelled in Simio . . . 76
6.5 Allocation of Wall-Restriction . . . 78
6.6 Wall Restriction . . . 78
6.7 Opening in Wall Visualised from *ESAB-room* . . . 78
6.8 Representation of a Status Pie with States . . . 80
6.9 Status Pie Charts for Model-1.0 . . . 81
6.10 Representation of Workstations' Utilisation After *l_weld*'s Processing Time is Decreased by 70% . . . 86
6.11 Placement of Additional *roll* Machine . . . 87
6.12 Comparison Between Model-2.1's and Model-2.2's *roll* workstation(s) . . . 87
6.13 Results Obtained for Adding Additional *roll* Workstation . . . 88
6.14 Results Obtained for Decreasing *dress* Workstation's Processing Time by 30% . . . 88
6.15 Identification of Additional *dress* Workstation . . . 89
6.16 Demonstration of Additional *dress* Workstation . . . 89
6.17 Results Obtained for Adding an Additional *dress* Workstation . . . 90
6.18 Results Obtained after re-roll Operation is Reduced by 10% . . . 91
6.19 Representation of Workstations' Utilisation After the *c_weld* Workstation's Processing Time is Decreased by 52% . . . 93
6.20 Representation of Workstations' Utilisation After the *assem* Workstation's Processing Time is Decreased by 22% . . . 94
6.21 Location of Additional *assem* Workstation . . . 94
6.22 Representation of Additional *assem* Workstation within Simio . . . 95
6.23 Representation of Workstations' Utilisation After an Additional *assem* Workstation is Added to System . . . 96
6.24 Representation of Workstations' Utilisation After *l_weld* Workstation's Processing Time is Decreased by 68% . . . 97
6.25 Representation of Workstations' Utilisation After *c_weld* Workstation's Processing Time is Decreased by 40% . . . 98
6.26 Representation of Workstations' Utilisation After *inspect* Workstation's Processing Time is Decreased by 85% . . . 100

B.1 Mechanical Fabrication Floor Layout - with Relevant Dimensions . . . 112
B.2 Mechanical Fabrication Floor Layout . . . 113
B.3 Schematic Floor Layout of Mechanical Fabrication Job Shop with Area Numbering . . . 114
B.4 Visual Demonstration of Machine Types in MechFab . . . 117

C.1 Material Certificate (1) . . . 120
C.2 Material Certificate (2) . . . 121

D.1 Measuring and Marking Tools for *s_line* Workstation . . . 122
D.2 Representation of *s_line* workstation . . . 123
D.3 Representation of *dress* workstation . . . 123
D.4 Representation of *roll* workstation . . . 124
D.5 Assessment of Canister's Shape . . . 124
D.6 Brief demonstration Regarding the Use of *dog and wedge* [2] . . . 125
D.7 Visual Demonstration of the *dog and wedge* Tools Being Used . . . 125
D.8 Representation of *l_weld* workstation . . . 125

G.1 Histograms of processing times' data . . . 131
G.2 Empirical Density and Cumulative Distribution of Operations' Process Times . . . 132
G.3 Cullen and Frey Graphs of processing times' data . . . 133
G.4 Histogram and Theoretical Densities of Operations' Process Times . . . 134
G.5 Q-Q Plots of Operations' Process Times . . . 135

I.1 Roll Workstation - Two Operations Process . . . 183
I.2 Process at *assem_transnode*, First bundle of steps . . . 184
I.3 Process at *assem_transnode*, Second bundle of steps . . . 185
I.4 Process at *assem_transnode*, Third bundle of steps . . . 186
I.5 Process defined as *assem_Processing* . . . 187
I.6 Process defined as *assem_Processing* . . . 188

J.1 Gantt Chart Developed for Validating Model-0.0 . . . 189

K.1 Civil Drawing Regarding Opening in Wall to Improve Pipe Manufacturing Process Flow 190

L.1 Developed Model-1.0 . . . 191

List of Tables

2.1 Designations within Job Shop . . . 5
2.2 Work Schedule Pattern 1 . . . 6
2.3 Work Schedule Pattern 2 . . . 6
2.4 Workstations' Numbering, Abbreviation and Dimension . . . 9

4.1 *Pre-allocated* - and *booked* Mean Processing Times of Workstations . . . 33
4.2 Distribution Fits Related to Workstations' Processing Time . . . 40

5.1 Simio Standard Library Objects . . . 43
5.2 Standard Library and Processing Time Properties of Workstations . . . 44
5.3 Reliability Logic of Machines . . . 53
5.4 Symbol Transformation . . . 54
5.5 Vehicles' Ride Capacity . . . 58
5.6 Vehicle Objects Specifications . . . 59
5.7 OFD Notations . . . 62
5.8 Utilising Animation to Verify Model . . . 65
5.9 Mean Processing Times of Workstations . . . 66
5.10 Expected Travelling Times of Vehicles Between Workstations . . . 67

6.1 Number Allocation to Workstations for From-to-Chart . . . 72
6.2 Colour Codes Evaluation Criteria for From-to-Chart . . . 72
6.3 From-to-Chart depicting Current State . . . 73
6.4 From-to-Chart depicting Proposed Improved Process Flow . . . 73
6.5 Distance Travelled on Paths for Current and Proposed Workstation Allocations . . . 77
6.6 Rand value per hour Associated with Discipline . . . 82
6.7 Discipline Associated with Specific Workstation with Corresponding Rand value per Hour 82
6.8 Cost Factors Related to Model-1.0 . . . 84
6.9 Utilisation of *assem* Workstation for Specific Decrease in Processing Time of *l_weld* Workstations . . . 85
6.10 Mean values (hours) of roll Workstation's Two Tasks . . . 86
6.11 Throughput Results for Different Versions of Model-2.0 . . . 92
6.12 Model-2.0's Versions Cost Results . . . 92
6.13 Utilisation Results After Incremental Decrease in c_weld Workstation's Processing Time 98
6.14 Throughput Results for Different Versions of Model-3.0 . . . 99
6.15 Model-3.0's Versions' Cost Results . . . 99
6.16 Model-4.0's Version's Cost Results . . . 101
6.17 Summary of Developed Models . . . 102
6.18 Utilisation Results - Model-0.0 till Model-2.4 . . . 102
6.19 Utilisation Results - Model-2.5 till Model-4.1 . . . 103
6.20 Throughput Results . . . 103
6.21 Cost Factors - Model-0.0 till Model-2.4 . . . 103
6.22 Cost Factors - Model-2.5 till Model-4.1 . . . 103

A.1 Additional Supply of Pipes Needed 110
A.2 Cost Comparison 110

B.1 Brief Discussion of Machine Types in Job Shop 116
B.2 Information Regarding Workforce 118

C.1 Chemical Elements of Plates 119

E.1 Information Regarding Transportation Methods 127

F.1 Distribution Information (1) [3] [4] 128
F.2 Distribution Information (2) [5] [6] [7] 129
F.3 Distribution Information (3) [8] [9] [10] 130

G.1 Distribution Identification for Workstations' Processing Time 136

Acronyms

CDF Cumulative Distribution Function.

EDA Exploratory Data Analysis.

ERP Enterprise Resource Planning.

ESAB Elektriska Svetsnings-Aktiebolaget.

MechFab Mechanical Fabrication.

ML Machine Learning.

MPT Magnetic Particle Testing.

MTBF Mean Time Between Failure.

MTTR Mean Time To Repair.

NB Nominal Bore.

NDT Non-destructive testing.

OFD Object Flow Diagram.

PDF Probability Distribution Function.

Q-Q quantile-quantile.

SAP Systems Applications and Products.

TOC Theory of Constraints.

Chapter 1

Research Overview

1.1 Background and Rationale

The Central Workshops manufacture and recondition plant-related equipment within an integrated steel manufacturer. The steel manufacturer encompasses all the necessary processes needed to transform raw materials into domestic and global market-grade steel. The manufacturing process is an intricate and continuous operation, involving a series of thermal, chemical and physical processes across various plants. The integration of these processes is essential to ensure the effective flow of production, meeting both domestic and international market demands.

Situated within the premises of the steel manufacturer, hereinafter referred to as 'The Company', the Central Workshops offer a wide range of essential services to support various plants. The Central Workshops are typically equipped with various machining equipment, offering services such as turning, milling, drilling and grinding to fabricate and repair steel components and machinery. These services not only support steel production and maintenance processes but also ensure the smooth and efficient operation of The Company.

In recent years, the Central Workshops have encountered operational setbacks. Some of these setbacks include not adhering to deadlines, increased production costs, and decreased customer satisfaction. These setbacks can be attributed, in part, to challenges in management practices. Management decisions, at times, have lacked strategic foresight, impacting the Central Workshops' overall efficiency and effectiveness. Furthermore, The Company's recent retrenchment decisions, while aiming at cost reduction and operational streamlining, had unintended consequences for the Central Workshops. The reduction in skilled personnel resulted in diminished capacity and efficiency, leading to longer downtimes and delays in adhering to production demands.

These operational deficiencies in the Central Workshops not only result in immediate setbacks but also have long-term consequences. Such consequences include increased production costs, production demands not being met, and if deemed not profitable, it will lead to the closing down of the Central Workshops.

The closure of the Central Workshops can have significant consequences for The Company as a whole. First, it can disrupt The Company's production, leading to delays and potentially affecting product quality and safety. Maintenance costs may rise as external service providers become necessary, increasing operational expenses. Job losses among Central Workshops' employees can harm workforce morale and potentially result in labour disputes. Increased lead times regarding the repair of critical plant equipment, decreased sustainability, and lower profitability of The Company are among the other consequences. Hence, it is clear that the inefficiencies within the Central Workshops need to be addressed.

Although it is clear that the need is there to identify improvement opportunities, it must be noted that this task, especially in a job shop environment, is challenging. Firstly, the difficulty arises through

the inherent complexity of job shop operations. Unlike more repetitive manufacturing settings, job shops produce a wide array of custom-made products, each with unique requirements and processes [11]. This heterogeneity, which is further elaborated on in the literature, makes it difficult to identify areas for improvement, as solutions must often be suited for each specific job. Secondly, the limited predictability of job shop workflows due to the constant arrival of diverse orders and variations in production demands further complicates the identification process. The dynamic nature of job shops, where priorities can change instantly due to unscheduled work, makes it difficult to establish clear benchmarks and identify bottlenecks or inefficiencies.

1.1.1 Central Workshops

The Central Workshops' main responsibility includes the reduction of the total maintenance cost of The Company by providing manufacturing, reconditioning, and a 24-hour standby service. The Central Workshops consist of four job shops, each embodying unique tasks, responsibilities, and operational characteristics, collectively contributing to a versatile and customer-centric manufacturing and reconditioning environment. Due to the available resources and required data available for conducting a comprehensive study, this dissertation will narrow its focus to a Mechanical Fabrication job shop that forms part of Central Workshops. This Mechanical Fabrication workshop will herein further be referred to as MechFab.

1.2 Research Scope

Within this section, the research problem is stated, and the objectives of the dissertation's completion are discussed.

1.2.1 Problem Statement

There is no methodology to effectively identify and evaluate improvement opportunities within the job shop environment.

1.2.2 The Aim

In light of the existing inefficiencies within the Central Workshops, focusing on the MechFab job shop, there is a need to develop a methodology for the systematic identification and exploitation of improvement opportunities. Due to the process heterogeneity within the MechFab job shop, it is recommended to initially concentrate on a single operation, specifically the pipe manufacturing process. More details on the pipe manufacturing operation and the associated decision are discussed in Section 2.2.

The methodology to be developed is envisioned as a versatile tool that can not only benefit jobs in the MechFab job shop but can also be extended to similar operations.

1.2.3 Objectives

The objectives of the study entail the following:

1. **Develop an effective methodology**: The first objective is to identify and/or develop a suitable technique or approach to adequately identify improvement opportunities within a system. Given the intricate and unpredictable dynamics inherent to the job shop environment, the methodology must take uncertainty and complexity into account.

2. **Identify improvement opportunities:** The research aims to identify opportunities for improvement by systematically evaluating existing processes.

1.3 Research Approach

To effectively address the research's objectives as outlined above, the following research approach is applied:

1. It is important to obtain the necessary knowledge regarding the environment under consideration. Therefore, a thorough analysis is first conducted to evaluate the system under consideration.

2. A comprehensive literature study is performed regarding techniques that can be used to address the study's problem.

3. After identifying the solution technique to be applied to identify improvement opportunities within the system, it is required that the relevant data is extracted and adequately evaluated.

4. Once an appropriate technique is identified and the necessary data analysis is done, the next step involves constructing the proceedings of the system. The primary objective of this stage is to construct a comprehensive model that has undergone thorough verification and validation procedures, thereby ensuring its ability to accurately portray the system under consideration.

5. Upon completion of the original model, which reflects the current proceedings of the system, an exploration regarding possible improvement initiatives is to take place. This involves a systematic approach where a series of models is consecutively developed. Each iterative phase encompasses the development of a new model, followed by an analysis of its performance to identify potential areas for improvement. These identified improvement opportunities then inform the subsequent iteration's model development.

6. The results obtained from each model scenario will be compared by evaluating applicable performance measures. Subsequently, a conclusive assessment will be drawn that evaluates the identified improvement initiatives.

1.4 Chapter Division

The first chapter, namely Research Overview, which is the current chapter, is followed by six chapters, a bibliography, and several appendixes. In Chapter 2, a brief discussion regarding the MechFab job shop takes place. Thereafter the main objective of Chapter 2 is to provide sufficient information to understand the relevant operations and procedures of the system under consideration.

The literature review, which takes place in Chapter 3, serves as a tool to present sufficient literature related to the achievement of the study's goals. Initially, the chapter introduces the concept of systems and models. Subsequently, it assesses three modelling techniques and identifies the most appropriate approach for further investigation. Next, the chapter delves into the significance of using Probability Distribution Function (PDF)s to represent real-world variability. The chapter explores the relevant PDFs and evaluates their role in integrating randomness into the forthcoming models.

With the appropriate modelling methodology identified, through the guidance of literature done in the previous chapter, the subsequent step involves extracting, refining, and interpreting the data required for the model. This step forms the focus of Chapter 4, where a systematic approach is outlined to explore the relevant data. This data is then ready to be incorporated into the developed model.

After the required data is identified and relevant observations are made, a model is developed that adequately reflects the proceedings of the system under consideration. The construction of this model takes place in Chapter 5. To ensure reliability, the model is validated and verified, confirming, with sufficient accuracy, the reflection of the actual system.

Chapter 6 employs the Theory of Constraints (TOC) principles to subsequently identify bottlenecks within the system. The various iterative modelling results are evaluated using three distinct performance measures, namely *Utilisation*, *Throughput*, and *Production Cost*.

A final summary of the work completed, a conclusion of important findings, and an overview of the contributions made to the field of study are discussed in Chapter 7 of the dissertation.

Chapter 2

System Background

To provide a logical flow of the study, it is important to first provide an understanding of the system before the literature review takes place, which is discussed in the next chapter.

Within this chapter, an overview regarding the MechFab job shop is first provided. Thereafter, an applicable job within the MechFab job shop is identified on which further evaluation is done to consequently develop a comprehensive methodology, which will in turn address the study's objectives. Thus, after the overview of the MechFab job shop is provided, sufficient information regarding the identified job is elaborated on to ensure an understanding of the system.

2.1 Mechanical Fabrication Job Shop

The purpose of this section is to provide an overview of MechFab job shop. Firstly, the floor layout is discussed. Thereafter, a visual demonstration depicts the placement of the various machinery in the job shop, and lastly, a discussion regarding the workforce takes place.

2.1.1 Floor Layout

The MechFab job shop is approximately 10 234 m^2. The detailed floor layout with main dimensions is depicted in Figure B.1 in Appendix B.1, where Figure B.2 reflects the detailed floor layout without dimensions.

A schematic diagram of the job shop is developed for demonstration purposes, as depicted in Figure 2.1 below. The representation provides an overview of the stations that are present within the job shop. A more comprehensive explanation of each station is provided in Appendix B.

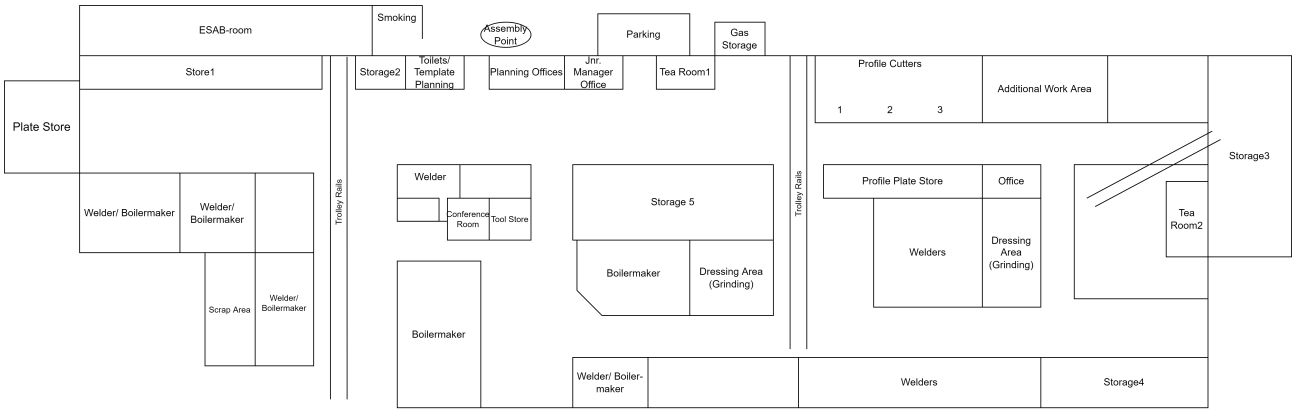


Figure 2.1: Schematic Floor Layout of Mechanical Fabrication Job Shop

The schematic demonstration serves two important purposes. Firstly, the demonstration offers a comprehensive overview of the MechFab job shop. Secondly, this visual representation is utilised during the explanation and elaboration of the job/ system under consideration, to be discussed further on in the dissertation.

2.1.2 Machine Allocations

Within MechFab job shop, various machines are utilised to attend to the required and different tasks or jobs. Each machine type is allocated a symbol, as can be observed in Figure 2.2 below, where the locations of the relevant machine types are displayed in Figure 2.3.

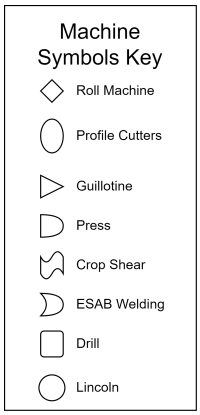


Figure 2.2: Symbols that Reflect Types of Machines

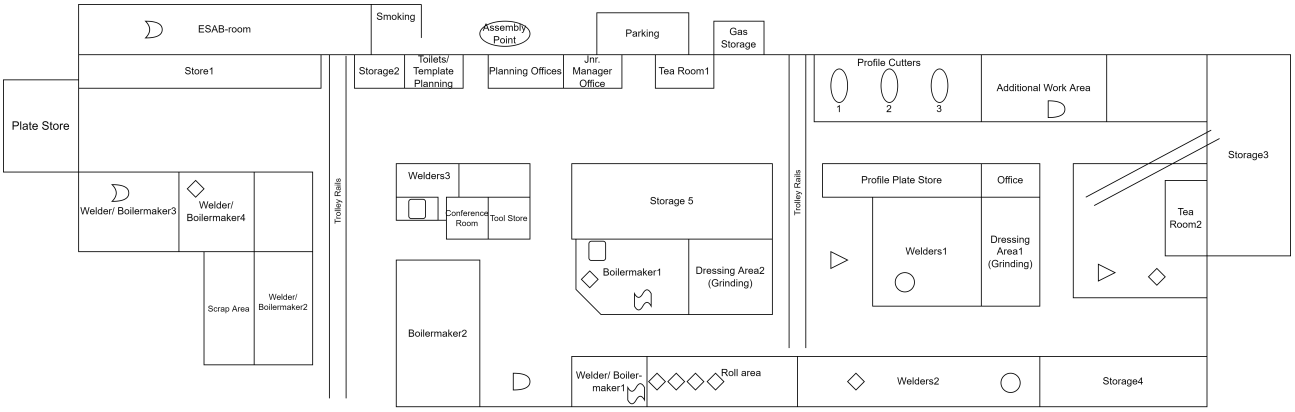


Figure 2.3: Schematic Floor Layout of Mechanical Fabrication Job Shop Depicting Allocation of Machine Types

Improving machine allocations across the entire MechFab job shop is an important factor to consider when assessing and exploring avenues for improvement initiatives. A concise description of the various machine types is provided in Table B.1 within Appendix B, where a visual demonstration is provided in Figure B.4 in the same appendix.

2.1.3 Workforce

The workforce is an important factor to consider, as tasks and operations are dependent on the workforce's contributions. The designations that are present within the job shop are listed in Table 2.1 below, where further information regarding the workforce's designations are provided in Table B.2 which are present within Appendix B.

Table 2.1: Designations within Job Shop

Designation	Description	Related Icon
Boilermaker	Fabricates steel structures.	B
Welder	Attends to welding operations.	W
Operator dresser	Ground edges of plates.	D
Profile operator	Operates profile cutting machines.	P

From Table 2.1 above, it can be observed that each designation is allocated a colour icon with a corresponding letter. For example, the "Boilermaker" designation is assigned the colour purple with the letter "B". The purpose of this icon allocation is evident in Figure 2.4 below, where the locations of each designation can be observed.

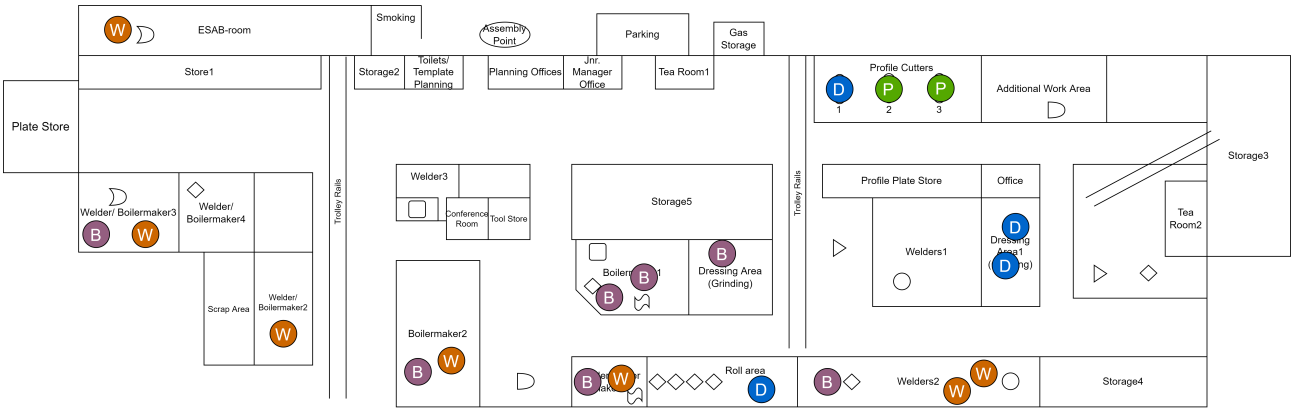


Figure 2.4: Workforce within MechFab

Work Schedule

An employee's or a group of employees' specific hours and days of work are outlined in a work schedule, which is a predefined plan or timetable [12]. The work schedule that is outlined for the pipe manufacturing operation is an important factor to consider as the schedule defines when operations at the various workstations take place. The evaluation regarding the pipe manufacturing operation is further elaborated on in the next section.

There are two work schedules which the employees follow weekly, as outlined in Table 2.2 and Table 2.3 below.

Table 2.2: Work Schedule Pattern 1

Activity	Time
Work period 1:	06:20 - 09:00
Tea break 1:	09:00 - 09:15
Work period 2:	09:15 - 12:00
Lunch break:	12:00 - 12:30
Work period 3:	12:30 - 14:00
Tea break 2:	14:00 - 14:15
Work period 4:	14:15 - 16:00

Table 2.3: Work Schedule Pattern 2

Activity	Time
Work period 1:	06:20 - 09:30
Lunch Break 1:	09:30 - 10:00
Work period 2:	10:00- 13:00

The initial work pattern, presented in Table 2.2 above, is implemented from Monday through Thursday. In contrast, the second pattern, as outlined in Table 2.3 above, is exclusive to Fridays. No operational tasks take place on Saturdays and Sundays.

2.2 Pipe Manufacturing

In this section, a discussion is first centred on the rationale behind selecting the pipe manufacturing operation as the most suitable system for the development of the proposed methodology. Subsequently,

a discussion is initiated on the pipe manufacturing operation, presenting important details to adhere to the study's objectives.

2.2.1 Decision to Consider a Specific Job

As already mentioned in section 1.1, due to a job shop's formality and inconsistency, it is difficult to develop an effective methodology to adequately address the inefficiencies of all the jobs within the MechFab job shop environment. To address this challenge, a strategic approach involves initially focusing the methodology development on a single job. The developed methodology will then serve as a versatile tool that can be used to further identify improvement opportunities within other jobs.

Introduction to the Pipe Manufacturing Operation

Within The Company, there are various processes and operations in the plants that require the relevant gasses to enable production to take place. Gasses are important elements used in diverse stages of the steel manufacturing process, ranging from heating to cooling, to chemical reactions and power generation. The effective distribution of these gasses between different plants is facilitated through an extensive network of pipes. The utilised pipes are important to ensure the efficient flow of gas, thus directly impacting the overall productivity and effectiveness of the steel manufacturing company.

These essential gas-transporting pipes are manufactured within the MechFab job shop. The pipe manufacturing operation is identified as the ideal job to evaluate, to consequently adhere to the study's objectives. The decision to focus on the pipe manufacturing operation stems from the continuity within the job shop. In contrast to intermittent tasks, where there is uncertainty regarding their recurrence, pipe production is a job that always needs to be adhered to. The reason for the continuity of the pipe manufacturing job is correlated to the necessity of the pipes in the plants. Further elaboration regarding the requirements and demand of the pipes is provided in Appendix A.

2.2.2 Manufactured Pipes Overview

The system under investigation manufactures 1000 Nominal Bore (NB) pipes with a total length of 12-metres. A visual demonstration of one manufactured pipe is displayed in Figure 2.5 below.

Figure 2.5: Demonstration of 12-metre, 1000 NB pipe

Plates are the primary input material for pipe manufacturing, characterised by dimensions of approximately 6,500 mm in length, 3,000 mm in width, and 10 mm in thickness, with an approximate weight of 1,575 kg. The job card with relevant dimensions and information that are used for this study's analysis is displayed in Figure C.1 in Appendix C.

2.2.3 The Process

An overview of the pipe manufacturing process is provided in the flowchart, in Figure 2.6 below.

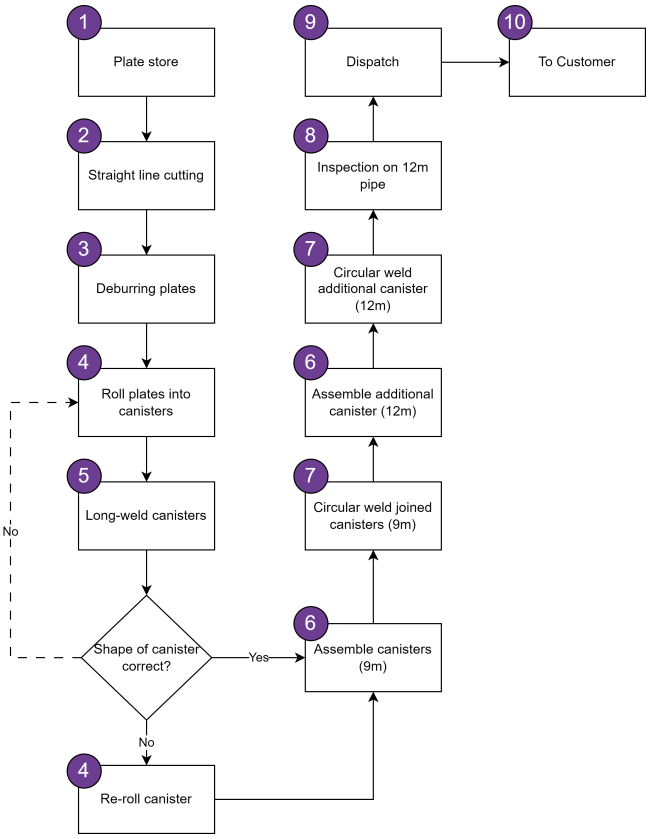


Figure 2.6: Pipe Manufacturing Flowchart

From the flowchart, it is observed that each process step is assigned a unique number, which correlates with the numbers that are displayed in the schematic floor layout in Figure 2.7 below. Hence, an overview of the positioning of workstations is provided, where a discussion regarding the information of the individual workstations takes place in the subsequent subsections.

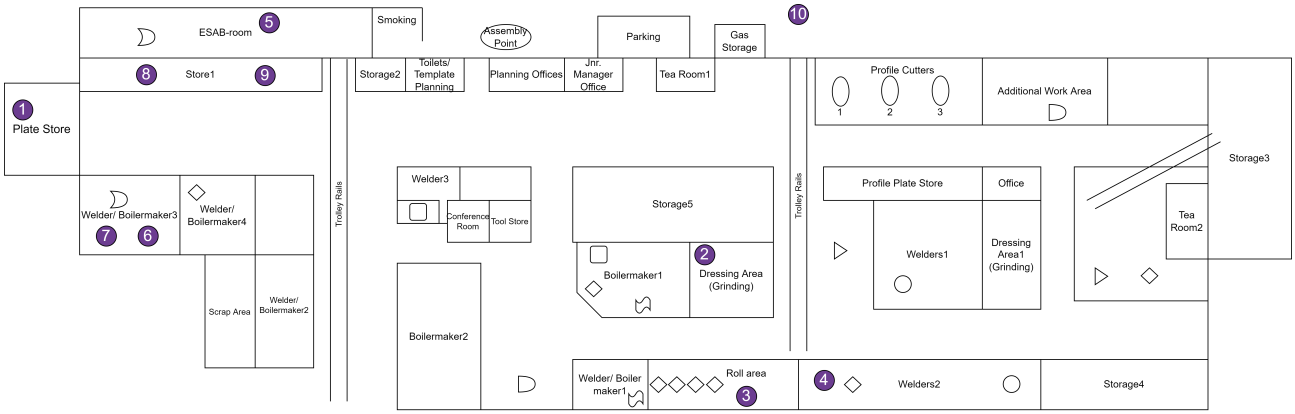


Figure 2.7: Schematic Representation with Work-Area Allocations

To enhance clarity and facilitate effective demonstration further on in the dissertation, an abbreviation has been assigned to each workstation, as listed in Table 2.4 below. Throughout the remainder of the study, the workstations will be consistently referenced using the respective abbreviation. Furthermore, the workstations' space requirements are also listed. The dimensions serve as an important factor to consider later on in the study as the dimensions need to be considered when evaluating possible improvement initiatives.

Table 2.4: Workstations' Numbering, Abbreviation and Dimension

Workstation Number	Workstation	Workstation's Abbreviation	Length (m)	Width (m)	Square Metres (m^2)
1	Plate Store	plate_store	58	22	1300
2	Straight Line Cutting	s_line	13	11	149
3	Deburring of Plates	dress	12	11	125
4	Roll	roll	9	7	63
5	Long-weld	l_weld	73	12	869
6	Assembly	assem	18	5	94
7	Circular-weld	c_weld	18	10	189
8	Inspect	inspect	15	5	76
9	Dispatch	disp	11	3	33
10	To Customer	Finished	-	-	-

Before a discussion is provided regarding the operations at each workstation, it is beneficial to first obtain an overview of the transportation methods used during the pipe manufacturing operation.

Transportation Methods

The material is transferred to the required workstations using different transportation methods. The areas in which the various transportation methods are utilised are illustrated in Figure 2.8 below. From the illustration, it can be observed that five transportation methods are utilised. These transportation methods include overhead cranes, a trolley, a vehicle, rolling the canisters on the floor, and a low-bed. The transportation methods used in the designated areas are highlighted with the use of different colours, as depicted in the *Key* of Figure 2.8. Further information regarding the various transportation methods is provided in Table E.1 which is present in Appendix E.

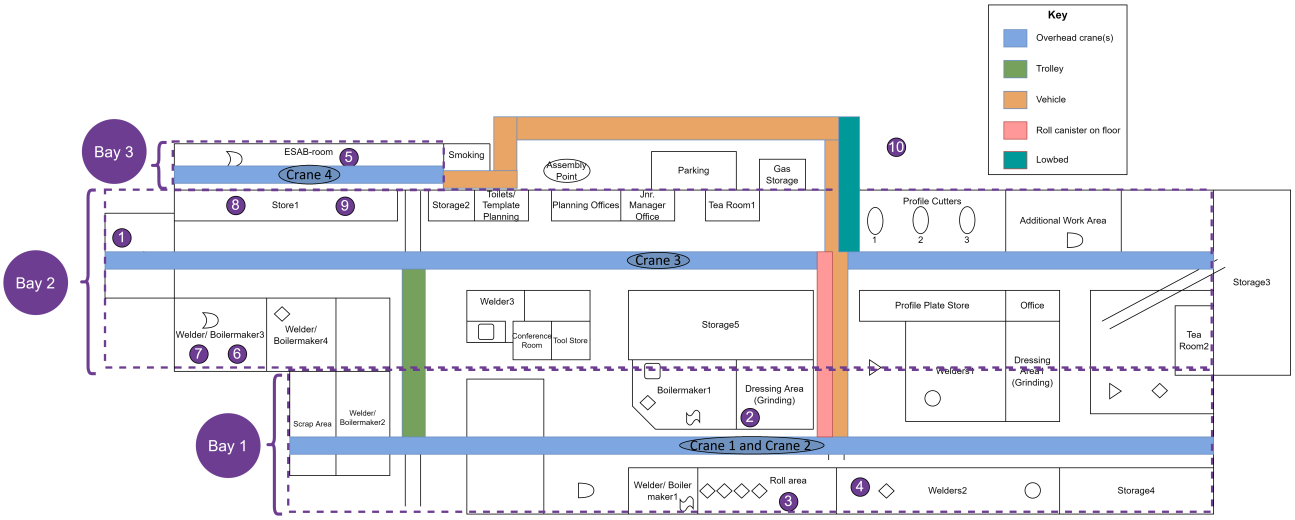


Figure 2.8: Transportation Mode Between Workstations

To further enhance the proceedings of the process flow, it is important to note that three distinct bays are present within the MechFab job shop, as depicted in Figure 2.8 above. These bays are primarily distinguished by the presence of the rails that facilitate the movement of the overhead cranes. As a result, the demarcated area, as indicated by the purple-striped blocks in Figure 2.8, identifies the operational zone where the cranes can be used.

From Figure 2.8 above, it can be observed that two overhead cranes, namely Crane 1 and Crane 2 are present within Bay 1. The usage of Crane 1 and/ or Crane 2 depends on the availability of the crane. Whereas for Bay 2 and Bay 3, only one overhead crane is present, Crane 3 and Crane 4, respectively. Since overhead cranes are exclusively used in the pipe manufacturing process, it is important to clarify

that the term 'crane' hereinafter pertains to overhead cranes.

1) Plate Store

The *plate_store* is a dedicated area in which the plates are stored. For the study, it is assumed that there is a sufficient amount of plates. The plates move through the system, where the plates are rolled into canisters and thereafter assembled into 12-metre pipes. Further explanations regarding canisters and the assembly of the canisters are elaborated on in the below subsections.

The flow of material is displayed in Figure 2.9 below, where the crane-operator uses Crane 3 to transfer the adequate amount of plates from the *plate_store* to the trolley where the plates are then transferred to Bay 1.

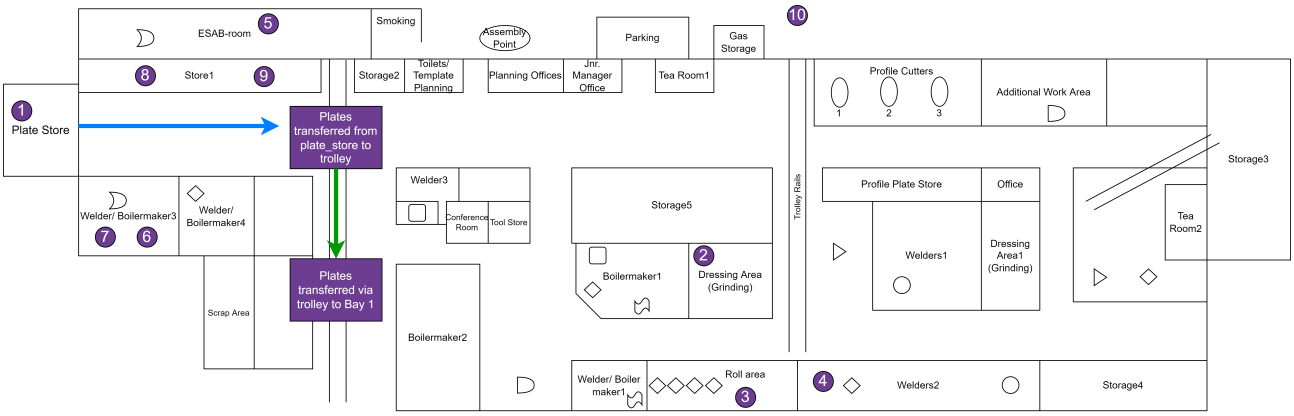


Figure 2.9: Material Transferred from Plate Store to Bay 1

For this study, it is assumed that a batch of twenty plates is transferred from *plate_store* to the trolley. The transferring of plates out of the *plate_store* commences when the stock of material reaches five plates at the subsequent workstation, namely the *s_line* workstation.

2.) Straight Line Cutting

After the plates are transferred to Bay 1 via the trolley, the crane-operator uses the adequate crane, namely either Crane 1 or Crane 2, to transfer the plates to the *s_line* workstation, as demonstrated in Figure 2.10 below. The usage of the specific crane depends on the availability between the two cranes.

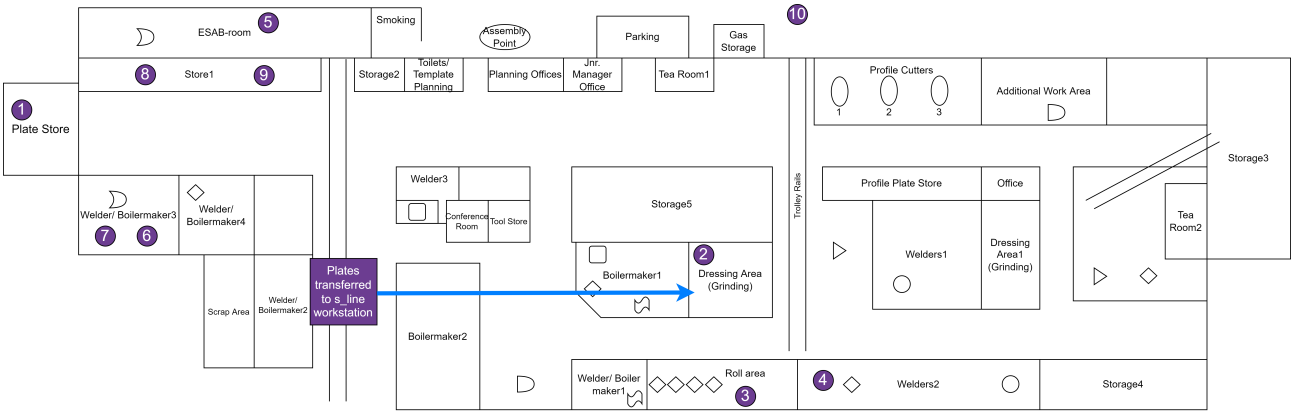


Figure 2.10: Material transferred from trolley to straight line cutting workstation

The *s_line* workstation's purpose is to cut one plate in half to deliver two smaller plates. Therefore, two plates are retrieved as output from the *s_line* workstation, with approximate dimensions of 3,174 mm x 3,000 mm x 10 mm. It is required that the plate is cut to the aforementioned dimensions to

ensure compliance with the following workstations' constraints, as well as to adhere to secondary equipment limitations, crane-weight limits, and overall handling of the plate. Further information regarding preparing and cutting the plate is provided in Appendix D.

3.) Deburring Plates

After the plates are cut to size at the *s_line* workstation, the plates are transferred to the area where the plates are ground. The transferring of the plates is demonstrated in Figure 2.11 below.

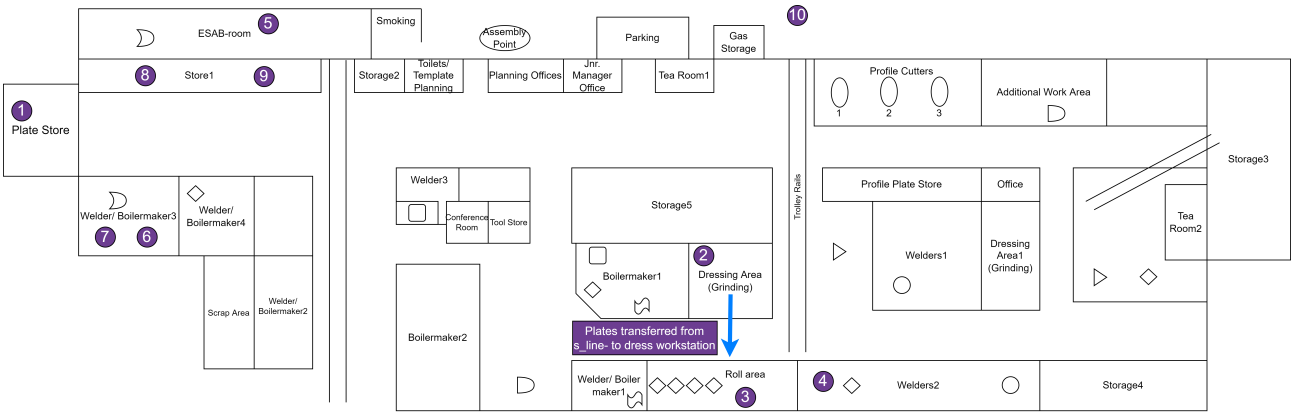


Figure 2.11: Material Transferred from *s_line* workstation to *dress* workstation

Burr removal by ground of the plates' edges ensures that the edges are smooth, promoting uniform and strong pipe connections for enhanced structural integrity. This process also improves welding quality by reducing the risk of flaws, such as cracks or porosity, and assists in achieving deeper weld penetration for increased strength.

4.) Roll

After the edges of the plates are ground at the *dress* workstation, the plates are transferred with a crane to the *roll* workstation. The transportation of the plates from the *dress* workstation to the *roll* workstation is demonstrated in Figure 2.12 below.

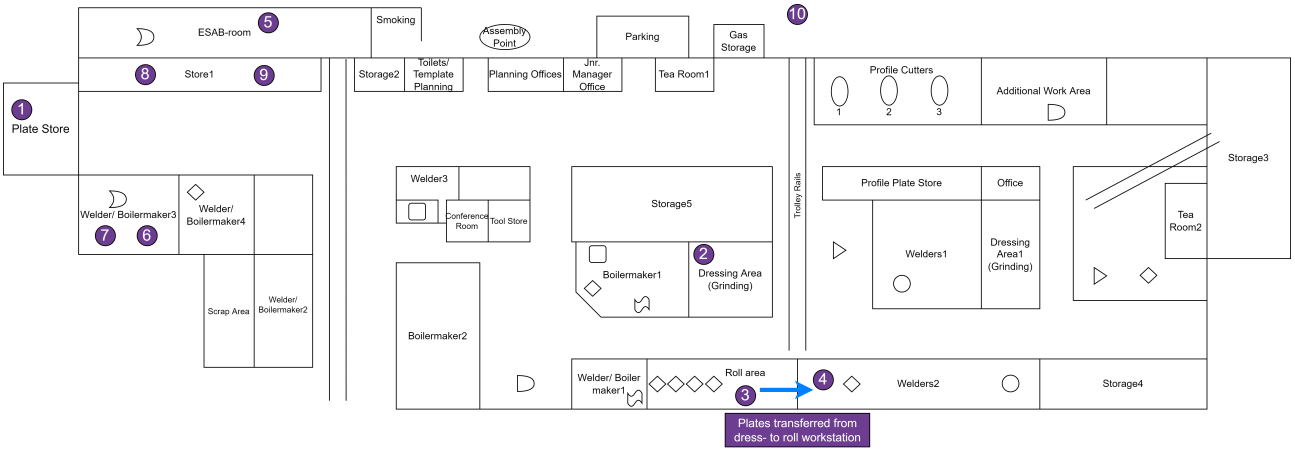


Figure 2.12: Material Transferred from *dress* workstation to *roll* workstation

At the *roll* workstation, a *roll press machine* is used to roll the plates into 'canisters', as well as re-rolling the 'canisters', if deemed necessary. A 'canister' is a term used to refer to a plate that has been formed into a cylindrical shape.

5.) Long-weld

Following the rolling process, the canisters are transported to the *l_weld* workstation using a flatbed mini delivery truck, referred to as 'vehicle'. A visual demonstration, as well as additional information,

regarding the 'vehicle', can be observed in Table E.1, which are presented in Appendix E. The transferring of canisters from the *roll* workstation to the *l_weld* workstation is illustrated in Figure 2.13.

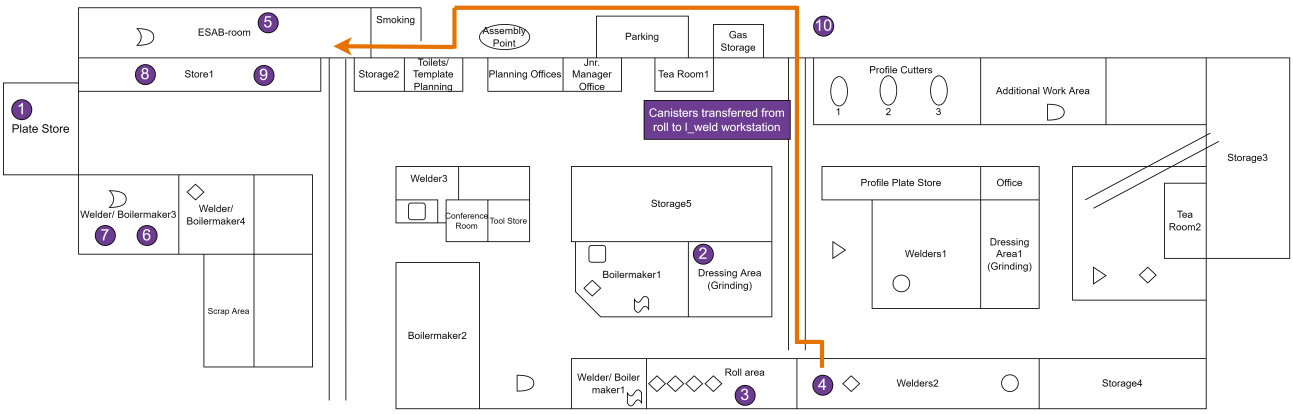


Figure 2.13: Material Transferred from *roll* workstation to *l_weld* workstation

The purpose of this workstation is to weld the seam of the canister which ensures a strong and secure bond, enhancing structural integrity and preventing any potential leaks or openings. An ESAB welding machine is used to acquire the workstation's objective. The term ESAB refers to a global manufacturer of welding and cutting equipment, consumables, and automation solutions [13]. Henceforth, the machine will be referred to as ESAB#1. To adequately reflect the operations related to the *l_weld* workstation, it is important to obtain an idea regarding the flow of material within the ESAB-room, which is the room in which the *l_weld* workstation is positioned:

1. The canisters arrive via the vehicle.
2. The operator uses Crane 4 to transfer the canisters from the vehicle onto the floor and in the designated area.
3. With the use of the Crane 4 the operator transfers the canisters to the ESAB#1 welding machine.
4. After welding takes place, the canisters are transferred via the crane to the designated area, where the canisters are inspected for any defects.
5. Thereafter the welded canisters are transferred via Crane 4 onto the vehicle, to be transferred to the next workstation.

A visual demonstration regarding the flow within the ESAB-room is depicted in Figure 2.14 below, where the allocated numbering correlates to the above explanation.

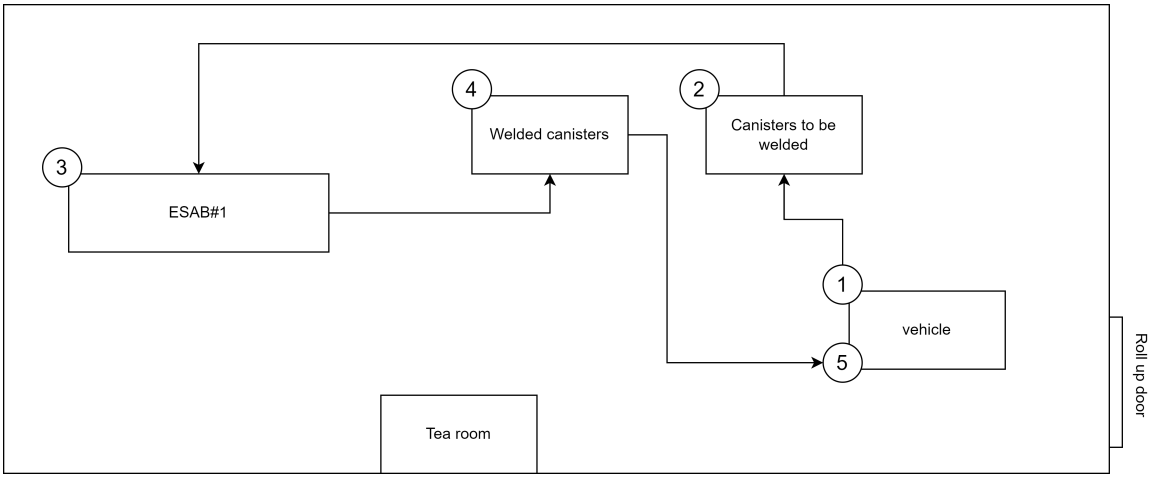


Figure 2.14: Material Flow within ESAB-room

6.) and 7.) Assemble and Circular-weld

Following the operations at the *l_weld* workstation, there is a possibility that the canisters may undergo slight deformation. A canister can deform during seam welding due to the high heat causing localised expansion and contraction of the canister. When a canister deforms, the canister must be re-rolled at the *roll* workstation. The purpose of re-rolling the deformed canister is to ensure that it acquires the necessary shape for efficient utilisation at subsequent workstations. Once the canisters have achieved the required shape and form, the canisters are transferred to the *assem* workstation where further operations take place.

It is identified that approximately 40% of canisters are re-rolled whereas approximately 60% of canisters proceed directly to the *assem* workstation. Figure 2.15 provides a graphical example of the above statement.

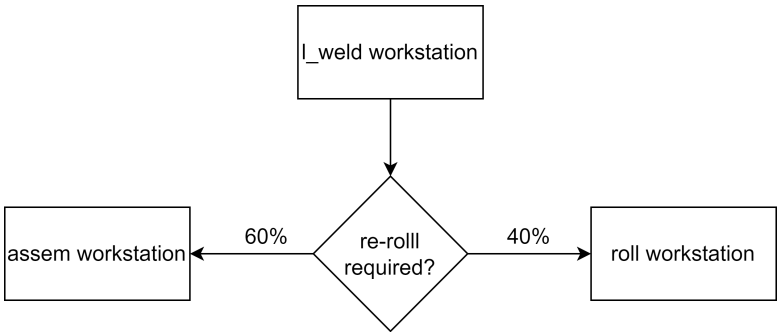


Figure 2.15: Decision Representation after *l_weld* Workstation

To further assist with the above statement, the flow of canisters from and to the *l_weld*, *roll*, and *assem* workstations are demonstrated in Figure 2.16 below.

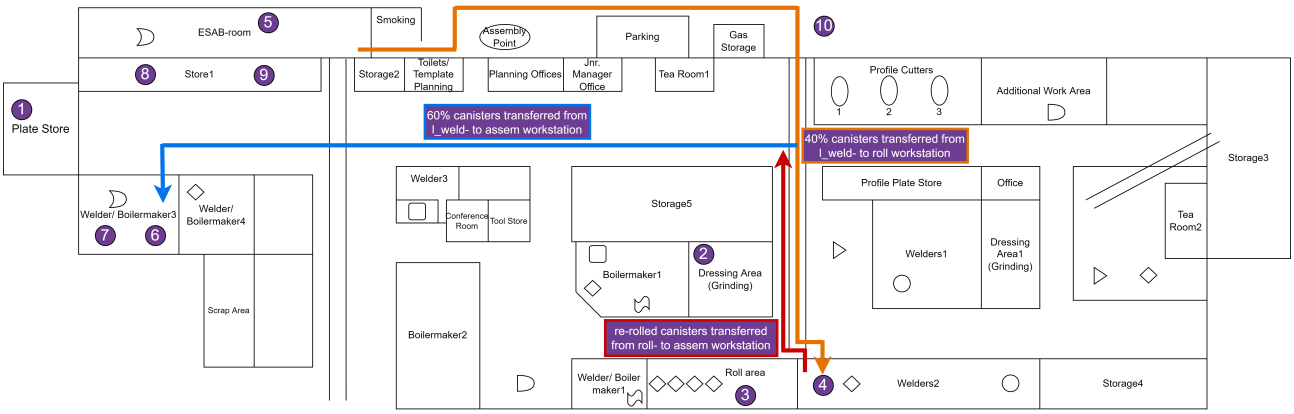


Figure 2.16: Material transferred to *assem* workstation

Based on Figure 2.16, it can be observed that after the operations at the *l_weld* workstation took place, approximately 40% of the canisters are returned to the *roll* workstation, as indicated by the orange line. Around 60% of the canisters are transferred to the *assem* workstation, as represented by the blue line. After the canisters are re-rolled at the *roll* workstation, the canisters are transferred to the required bay, as identified with the red line, where-after the canisters are transferred to the *assem* workstation with the use of the relevant crane.

The *assem* and *c_weld* workstations' operations are simultaneously explained due to the deliverables of the workstations being dependent on one another. To provide an understanding of the two workstations' operations, a discussion on the process of delivering a single 12-metre pipe is provided. It is important to note that during 'real life' operation, the interconnected manufacturing of multiple 12-metre pipes occurs. Furthermore, it is also important to note that a total of four canisters, which have an approximate length of 3 metres each, is required to deliver a one 12-metre pipe. The required operations to manufacture a single 12-metre pipe are as follows:

1. First, the *assem* workstation requires three canisters. As already mentioned, canisters are transferred from either the *roll* or the *l_weld* workstation to the *assem* workstation. Note that the fourth canister, to consequently deliver a 12-metre pipe, is joined later on.

2. The three canisters are stacked on two I-beams to axially align via the use of Crane 3.

3. By rotating each canister clockwise, a 90-degree offset between seam welds is achieved, resulting in improved structural integrity, leak prevention, weld quality, and manufacturing efficiency.

4. The sides of the corresponding canisters are joined together by tack welding the canisters manually. Thus, 3 canisters are joined, resulting in a 9-metre joined assembly.

5. Thereafter, the operator transfers the 9-metre assembly, with the use of Crane 3, to the *c_weld* workstation. The machine utilised at the *c_weld* workstation is an ESAB welding machine, similar to the ESAB machine that is used at the *l_weld* workstation. The machine is further referred to as ESAB#2 welding machine.

6. The 9-metre assembled pipe is placed on supports in front of the machine. The supports enable the pipe to rotate on its axis. This rotation is necessary to ensure that circular welding on the seams of the attached canisters can take place while the welding head of the ESAB#2 machine stays in a fixed position. After completing the necessary operations at the *c_weld* workstation, a 9-metre pipe is complete.

7. The 9-metre pipe is transferred back to the *assem* workstation with the use of Crane 3. Similar steps are followed as outlined in step 2 till step 4, however, only one additional canister is joined.

8. After the required operations are done to attach the fourth canister, the 12-metre assembly is transferred back to the *c_weld* workstation. Similar operations, as mentioned in step 6 and step 7, takes place, however, only one seam is welded.

9. After circular welding the fourth canister's seam, a 12-metre pipe is delivered.

10. The 12-metre pipe is transferred to the next workstation, namely *inspect*.

The reason for initially assembling three canisters and subsequently adding the fourth canister, instead of directly assembling all four, is due to the limited reach of the ESAB#2 machine's arm length. A practical approach is used to first weld the two seams of the three canisters. After adding the fourth canister, the assembled pipe is rotated to ensure that the machine's arm can reach the final seam to be welded. A graphical example is provided in Figure 2.17 below, where the far right rectangle represents the ESAB#2 machine's arm, on which the welder-head is positioned. The numbered rectangles represent the canisters that are joined to consequently deliver a 12-metre pipe.

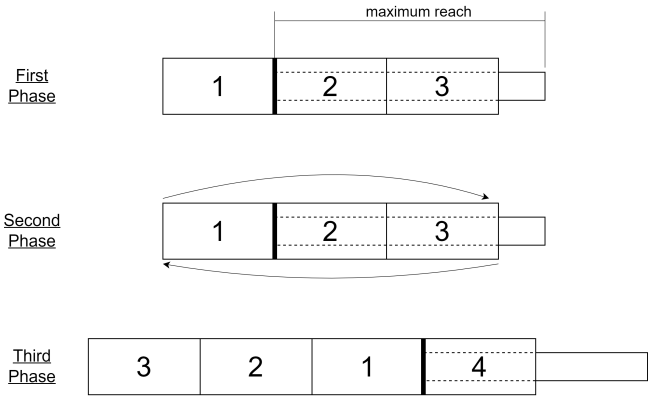


Figure 2.17: Placement of Canisters to Weld Seams Accordingly

To aid in transferring the manufactured pipes, lifting lugs are used. Lifting lugs are mechanical components used to facilitate the handling and lifting of the pipes. At the *assem* workstation, the lifting lugs are affixed after the 12-metre pipe is assembled. Note that specific manufacturing and assembly details regarding the lifting lugs are beyond the scope of this study.

8.) Inspect

After the operations are complete at the *c_weld* workstation, the 12-metre pipe is transferred to the *inspect* workstation, as can be seen in Figure 2.18 below.

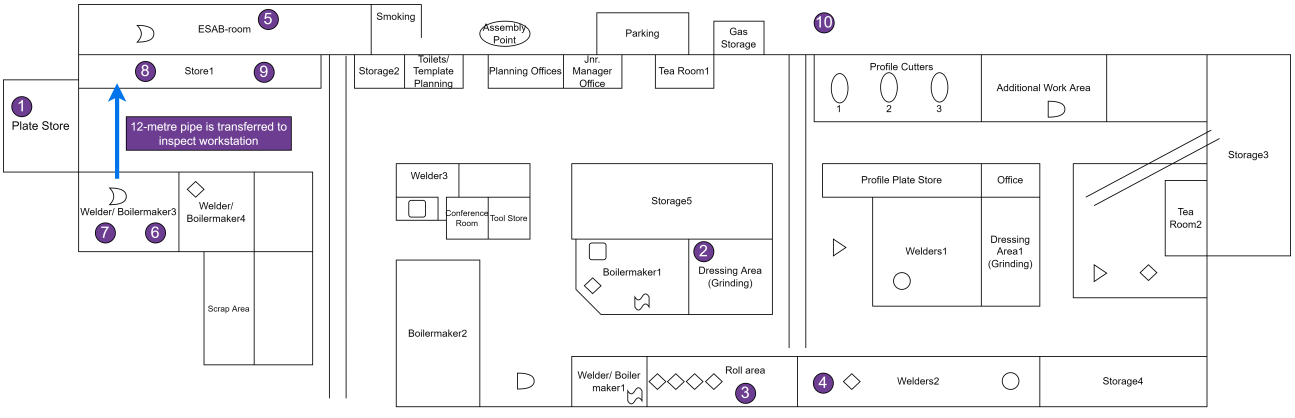


Figure 2.18: 12-Metre Pipe is Transferred to *inspect* Workstation

At the *inspect* workstation, the 12-metre pipe goes through an inspection and Non-destructive testing (NDT) before being delivered to the customer. Doing a comprehensive inspection is important to guarantee adherence to quality standards, safety regulations, and customer specifications. Further details regarding the inspection of the pipes are elaborated on in Appendix D.

9.) Dispatch

After the required inspection is conducted on the 12-metre pipe, the pipe is transferred, with the use of Crane 3, to the *disp* workstation, as can be seen in Figure 2.19 below.

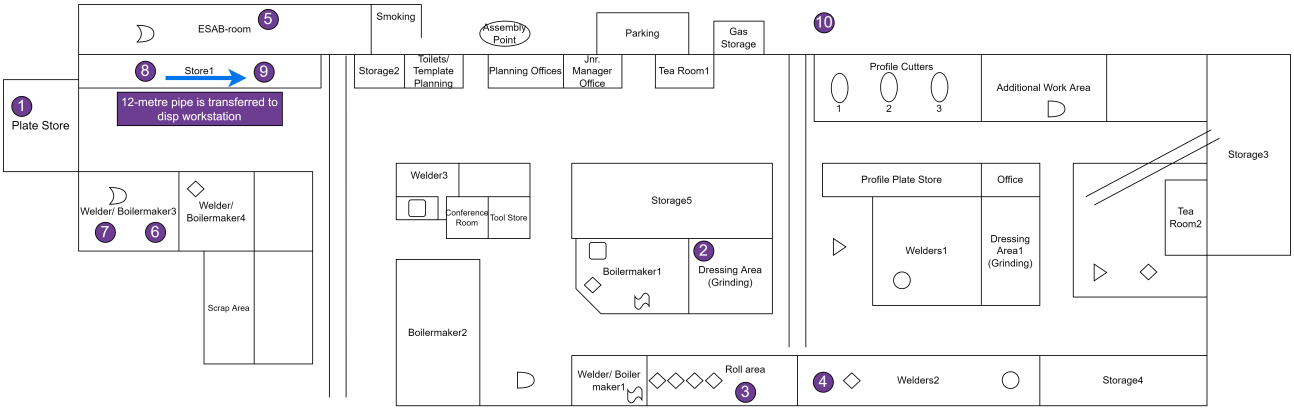


Figure 2.19: 12-Metre Pipe is Transferred to *disp* Workstation

The purpose of this station is to store the manufactured 12-metre pipes, awaiting the arrival of a 30-ton low bed to consequently transfer the pipes to the customer. A total of two pipes can be placed onto the low bed.

10.) To Customer

The conclusive stage, denoted as *Finished* signifies the departure of the 12-metre pipes from MechFab job shop to the customer. It is important to clarify that the study does not focus on activities that occur after this stage.

Chapter 4 will provide a discussion regarding the expected/planned processing time at each workstation, in addition to relevant data. The current focus was on providing sufficient information regarding the process flow and the pertinent information associated with each workstation.

2.2.4 Flow of Materials

The flow of materials is an important factor to consider to effectively identify improvement opportunities within the pipe manufacturing system. Hence, the total flow is depicted in Figure 2.20 below.

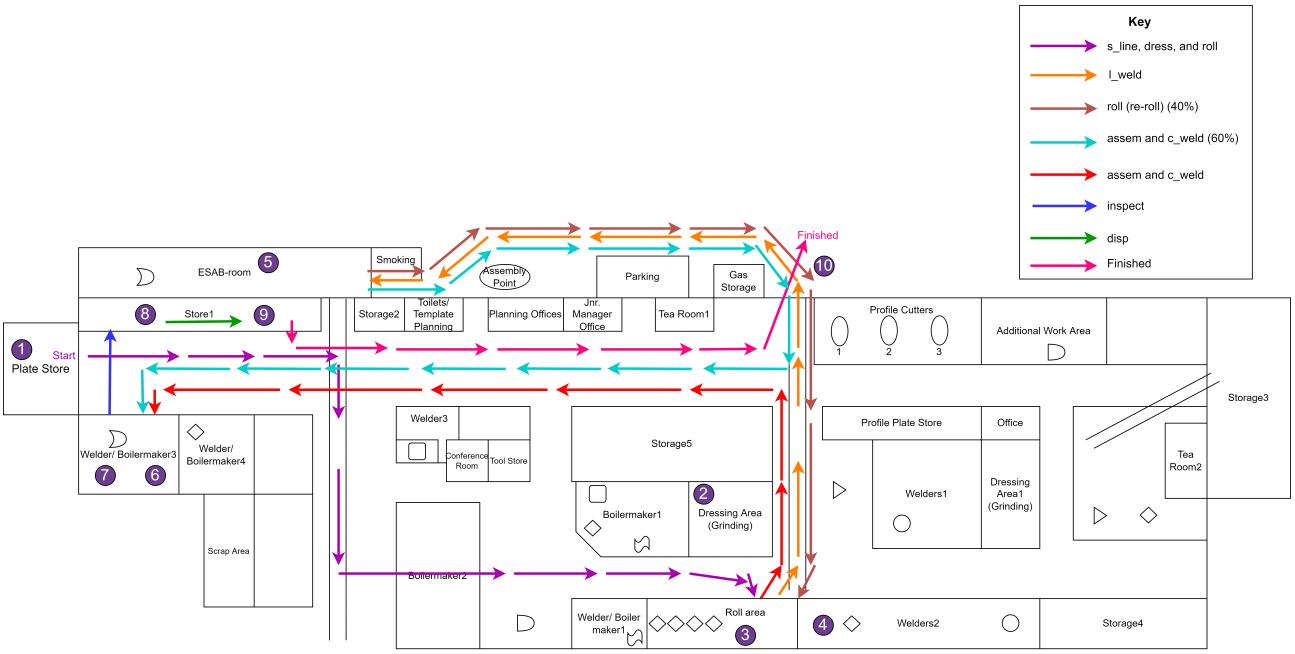


Figure 2.20: Flow of Materials

Within the demonstration, the arrows are colour-coded, which illustrates the flow of material to the relevant workstation, as identified in the *Key*. For example, the flow of material that is transferred from the plate store area to the work areas where the material is cut according to size, ground, and rolled into canisters, is identified by purple arrows, whereas the flow of material to the *l_weld* workstation is identified with orange arrows, and so forth.

The illustration presented in Figure 2.20 unmistakably illustrates a disorderly and unorganised material flow. At a glance, it can be assumed that the process flow can be improved by eliminating the need to travel outside the job shop to get to the *l_weld* workstation. Due to 60% of the canisters that need to be transferred from the *l_weld* workstation to the *assem* workstation, as identified by the black arrow illustrated in Figure 2.21 below, it would be ideal to investigate the possibility of shortening the travel distance. However, restrictions, such as a wall, are in the way of ensuring a shorter travel distance between relevant workstations. The identification of the improved process flow is further elaborated on in Chapter 6 where the From-to-Chart technique is used to elaborate on the positioning of workstations.

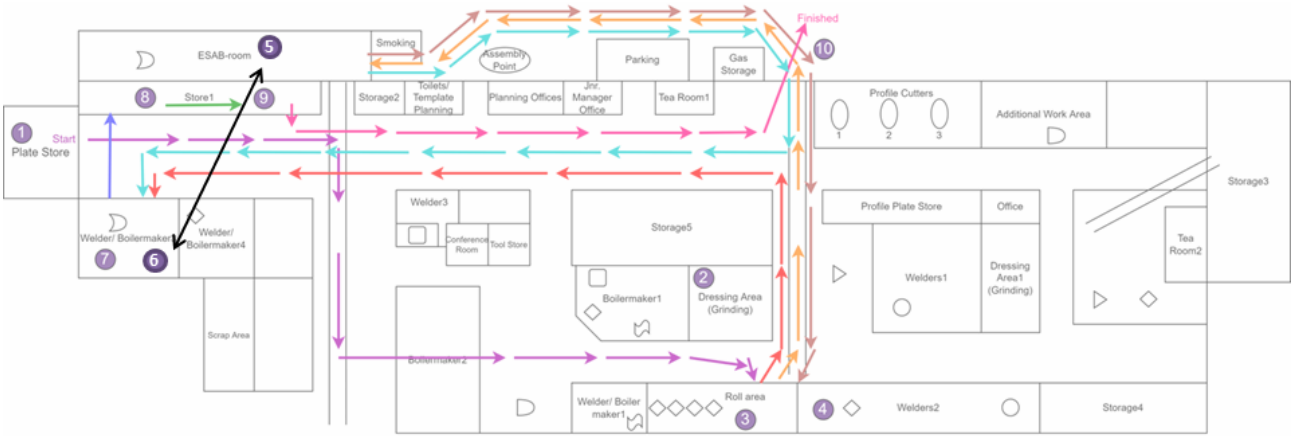


Figure 2.21: Improved Flow Presumed Possibility

2.3 Concluding Remarks: System Background

First, the chapter provided sufficient information regarding the layout of the MechFab job shop. Further, the relevant information, such as machine allocations and workforce discipline utilisation, was identified. It was established that the operational processes can only take place when the employees are on shift. Therefore, it was imperative to establish and consider their work schedules in the analysis.

The chapter further revolved around the core of the study: the pipe manufacturing operation. An introductory overview, with allocated abbreviations related to the workstations as well as space requirements of the workstations, was provided. The identification of the workstations' space requirements serves as important information necessary to sufficiently and effectively identify improvement opportunities. The depicted process flow was used as a reference to explain each workstation's purpose, while simultaneously emphasising the flow between workstations. The essential information for each workstation was diligently supplied, creating a robust foundation for conducting a thorough analysis. The understanding of the process flow was further enhanced by providing sufficient information to grasp an overview of the disorderly and unorganised flow of materials between the various workstations.

Each step in the analysis, from understanding the process flow to examining the transportation methods, considering work schedules, and identifying workstation dimensions, provided a comprehensive framework to effectively understand the process under consideration. Having established the necessary groundwork, the next logical step is to explore similar studies or problems. Therefore, the next chapter draws insights from past research and identifies potential solutions or methodologies employed in similar contexts that will enable the study to address its research problem.

Chapter 3

Literature Review

This chapter provides the relevant and adequate literature about addressing the study's objectives. First, a discussion regarding models that were used in literature to address problems relevant to the study takes place. Thereafter, an introduction regarding systems and models is elaborated on, emphasising the limitations of relying solely on observations to understand complexity. Three intricate models are evaluated and the most suitable approach is identified and further explored. Thereafter the importance of using PDFs in representing uncertainty in the identified modelling approach is addressed. The relevant PDFs are explored, as well as their role in incorporating randomness into the models to be developed. Finally, the chapter concludes with a brief discussion regarding TOC principles and the applicability thereof.

3.1 Process Heterogeneity

The literature surrounding methodologies for addressing process heterogeneity in manufacturing environments, particularly in job shop settings like MechFab, is rich and diverse. Scholars such as Deming, Juran, and Taguchi have extensively explored the consequences of process variation, emphasising its potential to lead to defects, rework, and operational inefficiencies [14] [15] [16]. Their work underscores the importance of comprehending and managing process heterogeneity for operational improvements.

In the field of utilising a methodology to identify improvement opportunities, researchers have developed innovative approaches tailored to address the challenges present within process heterogeneity. For example, J. McDonnell et al. conducted a study where they identified process heterogeneity within a manufacturing system and focused on a single process to effectively address this variability [17]. By concentrating on one process within a complex system, organisations can gain insights into its specific nuances and implement targeted improvements [17].

Lee et al.'s work introduces a methodological framework that utilises simulation modelling within improvement initiatives for heterogeneous manufacturing processes. This integrated approach not only facilitates a comprehensive understanding of the complexities inherent in process heterogeneity but also empowers organisations to devise strategies for enhancing operational performance [18].

The multifaceted nature of process heterogeneity emphasises the need for a focused approach to addressing variation. Hence, within the context of the study, pipe manufacturing was identified as the ideal operation to initially focus on, as detailed in section 2.2. This targeted focus allows for a deeper understanding of the specific challenges and opportunities within that process, paving the way for effective improvement initiatives.

3.2 Applicability of Modelling Approaches

The strategic application of models to identify improvement opportunities in complex environments, similar to the complexity of a job shop environment, has gained attention in the literature. The foundational works of Deming [19], Juran [20], and Shewhart [21] have identified the systemic use of models across diverse domains, providing a theoretical basis for addressing complexities inherent in job

shop operations.

Research has extended the principles of utilising modelling techniques, and investigating statistical and mathematical models in the context of job shops and other intricate operational settings. Scholars, such as Montgomery [22] and Grant and Leavenworth [23], have explored the efficacy of statistical process control models, identifying patterns and anomalies within the intricate processes similar to a job shop's setting. Additionally, analytical modelling, including linear programming, integer programming, and dynamic programming, provides avenues to mathematically formulate and solve complex decision-making problems [24] [25]. These techniques were used in practice to improve resources, effectively schedule tasks, and reduce costs while adhering to constraints. As stated by Bellman [24] [26], these analytical techniques aided in streamlining operations within various systems.

In the contemporary landscape, the integration of advanced modelling techniques, including Machine Learning (ML) and predictive modelling, has become important in addressing the unique challenges within complex environments [27]. As demonstrated by Chen et al. [28], ML, coupled with predictive modelling, was used to identify areas for improvement within intricate organisational workflows. Further modelling approaches that were utilised to identify improvement opportunities include data mining, which involves extracting patterns and valuable information from large datasets [29]. Techniques such as rule mining, clustering, and classification algorithms were employed to identify hidden relationships, group similar operational processes, and predict outcomes within intricate operational settings [29] [30].

Additionally, simulation modelling is presented as a powerful tool for comprehensively grasping and refining intricate systems, such as a job shop environment. By developing virtual representations of real-world processes, simulation models allow for testing different scenarios, evaluating various strategies, and predicting the impact of changes before implementation [31]. This approach has been utilised by researchers such as Banks et al. [32] to improve efficiency and resource allocation in manufacturing settings.

It is evident from the literature that there are various modelling techniques and approaches that can be utilised to effectively address the study's requirements. The exploration of various modelling frameworks underscores the versatility and adaptability of models in addressing the intricate challenges within a job shop environment. However, it is important to assess and evaluate the models' capabilities and formalities to identify the ideal modelling technique to effectively address the study's objectives.

3.3 Systems and Models

A system encompasses a collection of interconnected components that work together toward some purpose [33]. The process of gaining necessary insights into an existing system through observation has its limits. These limits may include requiring prolonged observations to encounter specific conditions of interest and to obtain sufficient data for reliable conclusions. Moreover, studying system changes can be challenging, as alterations can be hindered by practical constraints or costly implementations of evaluations [33].

In addition to the mentioned limitations, when observing systems without a modelling approach, there are various scenarios from the literature where the application of models proves to be invaluable in understanding and improving complex systems [34]. For instance, there are various reported cases where manufacturing plants aim to improve their production processes [34] [35]. It is identified that traditional observation-based methods necessitate extensive data collection over an extended period to identify bottlenecks and efficiency gaps accurately. Similarly, in the context of urban traffic management, it is identified to be impractical to rely solely on real-time observations to devise effective congestion reduction strategies [36]. Furthermore, models allow researchers to examine the effects of numerous causes and treatments within a controlled and ethical framework, which is advantageous in disciplines like epidemiology and climate science where conducting real-world scenarios is either infeasible or ethically challenging [34] [37] [38]. Models prove to provide a faster and more cost-effective means to

simulate different scenarios, evaluate potential interventions, and make data-driven decisions [36]. It is therefore proved by literature that models are an adequate approach to evaluate to consequently adhere to the study's objectives.

There are various types of modelling techniques, each with its own unique advantages and limitations [1]. *Physical models*, such as those representing objects like aeroplanes or cars, offer a tangible sense of reality and facilitate interactions with the physical environment. However, when studying processes or systems rather than physical components, the physical model approach may not be sufficient. Hence, the physical model approach will not be adequate to utilise for the pipe manufacturing system. *Analytical models*, on the other hand, employ mathematical representations, providing structured and deterministic solutions that are valuable within specific problem domains. Nevertheless, the analytical modelling technique can face challenges in handling uncertainty and lacks the necessary flexibility for complex systems. The analytical modelling approach is therefore also not necessarily the ideal approach to address the study's problem. In contrast, *simulation models* serve as a versatile modelling approach, replicating system behaviours and internal processes over time, which allows for detailed insights into system behaviour. Simulation modelling stands out for its capability to address and evaluate both uncertainty and complexity within a system [1]. Therefore, this adaptability and capacity to model intricate systems make simulation modelling an ideal technique to use and address the study's objectives.

3.4 Simulation Modelling

Simulation modelling, a computational technique that replicates real-world systems in a controlled digital environment, has proved to be an important tool across numerous academic and practical domains [31]. As stated by Banks et al. [39], simulation modelling enables the representation of complex systems, allowing for experimentation and analysis while mitigating real-world risks.

Simulation modelling is a versatile tool extensively used across various fields to discern and enhance opportunities for improvement within complex systems [31]. The literature provides several examples where simulation modelling has been deemed important in this regard. Van Der Merwe et al. [40] used pharmaceutical supply chains to illustrate the value of operations management and supply chain enhancement by cutting lead times and improving overall efficiency. Likewise, with healthcare, Brailsford et al. [41] demonstrated how simulation modelling may dramatically enhance emergency department operations by improving patient flow and resource allocation to decrease wait times in the healthcare industry. Transportation and traffic management have also effectively utilised simulation modelling, as identified by Ben-Akiva and Bierlaire [42], who used this modelling approach to improve transportation networks, alleviate congestion and enhance traffic flow.

Due to simulation modelling's effective power-solving capabilities for complicated systems that cannot be simply studied using the standard study approaches available, simulation has developed to be one of the most extensively used study tools as present [43]. Simulation modelling's key advantages include [43]:

- Do not interfere with current operations, new policies, operating procedures, organisational structures, information flow, and similar operations.
- The viability of theories explaining how and why specific occurrences happen can be examined.
- It is possible to locate bottlenecks in material, product and information flow.
- Understanding how the system actually functions, as opposed to how everyone believes it to.
- Before investing funds in procurement and/ or execution, new hardware designs, physical layouts, software programs, transportation systems, etc., can be tested.

Even while simulation solutions provide numerous benefits, there are some drawbacks as well [43]:

- Model construction takes specialised skills. The calibre of the analysis is influenced by the model's calibre and the modeller's expertise.
- Interpreting simulation findings might occasionally be challenging.
- Simulator analysis can be costly and time-consuming.

It is clear from the literature that simulation modelling can be utilised for various functionalities. Consequently, the identification of an appropriate simulation modelling technique is important when addressing a specific problem. In this context, it becomes imperative to assess the applicable simulation modelling approaches and identify the most suitable approach to effectively address the study's objectives.

3.4.1 Simulation Modelling Approaches

According to Kelton et.al [1], traditional modelling approaches may be identified according to three main dimensions:

1. **Static vs. Dynamic Models:** A *static simulation model* is characterised by its independence from the passage of time, as time does not play a significant role in the model's operation and execution. For instance, this type of model might employ a random number generator to simulate activities like gambling games or lotteries, estimate integral values or matrix inverses, or evaluate financial profit-and-loss statements. Whereas a *dynamic simulation model* utilises a computer program to model the time-varying behaviour of a dynamic model. Unlike static models, the use of dynamic models accounts for the change in system variables over time in response to various inputs and feedback mechanisms [1].

2. **Continuous vs. Discrete Models:** *Continuous simulation models* represent systems where the variables change continuously over time, typically described by differential equations. These models allow for a continuous representation of the system's behaviour, making the model suitable for phenomena that have a continuous flow such as fluid dynamics or chemical reactions. On the other hand, *discrete simulation models* represent systems where the variables change in discrete steps or events. Discrete models capture the system's behaviour at specific points in time, which assists in making discrete models applicable to utilise in situations with distinct events or actions, such as manufacturing processes and queuing systems [1].

3. **Deterministic vs. Stochastic Models:** *Deterministic simulation models* do not consider randomness and assume that the model's input parameters and variables have fixed, precise values. As a result, running the simulation multiple times with the same inputs will produce the exact same results. Whereas *stochastic simulation models* explicitly account for uncertainty by introducing randomness into the model's processes or inputs. The stochastic simulation models use probability distributions to represent uncertain variables, allowing for the generation of different outcomes each time the simulation is run [1].

Based on the study's objectives, it becomes apparent that the system being examined embodies a dynamic model. This dynamic nature implies that the system's behaviour evolves over time. Moreover, the system's characteristics align with those of a discrete simulation model, where the variables change in distinct steps or events. Additionally, the system can be classified as stochastic since randomness is inherently present within the 'real-world' pipe manufacturing operation [1].

The identified simulation modelling approach, characterised by its dynamic, discrete, and stochastic attributes, offers a highly effective means of analysing the pipe manufacturing operation. The effective implementation of these approaches is emphasised by literature, as several studies have highlighted the advantages of utilising the identified approaches within various industrial contexts.

Such observed advantages include the work of Heyman et al. [44], where the synergy between

dynamic modelling and stochastic elements in enhancing production processes is demonstrated, identifying improvements in production efficiency and resource allocation. Additionally, Vidalakis et al. [45] explored the integration of discrete-event simulation with stochastic modelling to address supply chain uncertainties, resulting in enhanced supply chain resilience and reduced operational risks. Hence, it is identified, and emphasised from the literature, that it will be ideal to utilise the simulation modelling approaches, to effectively address the study's objectives.

To effectively leverage the simulation approaches within the constraints of available resources, the chosen methodology involves the utilisation of Simio Simulation Software [46] to address the study's objectives. This study identified the Simio Simulation Software as the ideal choice to adhere to the study's objectives. The software was selected for its existing availability within The Company and the provision of necessary resources to facilitate effective user proficiency.

3.4.2 Simio Simulation Software

Simio Simulation Software is a versatile and robust computer-based tool designed for modelling and analysing complex systems and processes [46]. Users of Simio can construct dynamic simulations that replicate real-world scenarios, allowing the user to investigate "what-if" scenarios, identify bottlenecks, assess resource utilisation, and make data-driven decisions [46] [1].

Key features of Simio include a user-friendly modelling environment, powerful experiment design capabilities, and the ability to integrate with external data sources. Simio's versatility extends to both discrete event simulation and agent-based modelling, making the software suitable for a variety of applications. This software has gained popularity for its intuitive interface and advanced analytical tools, making it an essential resource for professionals and researchers seeking to enhance system performance and efficiency [46].

To fully harness the potential of Simio Simulation Software, a well-structured and systematic approach is necessary for achieving the study's objectives. The relevant process is therefore discussed in the following subsection.

3.4.3 The Simulation Process

The basic simulation process is depicted in Figure 3.1 below, as stated by Kelton et.al [1].

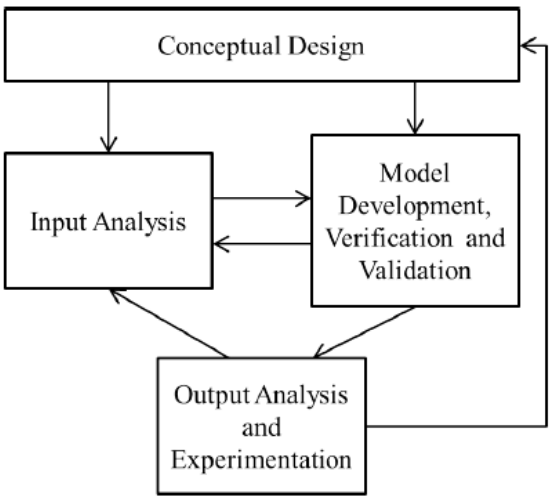


Figure 3.1: The Basic Simulation Process [1]

It is important to note that the basic simulation process is not strictly sequential: rather, the process frequently takes on an iterative nature. Two critical components must be adhered to during the *conceptual design* phase. Firstly, gaining an understanding of the system under consideration is essential.

This includes comprehending the required processes and relevant interactions. Secondly, adopting a basic modelling approach to develop the required simulation model. Therefore, the key elements and variables that influence the system's behaviour and relationships need to be identified. As evident from Chapter 2, this phase is already adhered to, where the outcomes and proceedings of each workstation, and consequently the system as a whole, were discussed.

Furthermore, *input analysis* entails characterising the system's inputs. The primary focus of the input-analysis is to characterise the input random variables and necessary data. Thereafter, the appropriate distribution for each phase is determined to consequently be employed within the developed model. Further literature regarding concepts that are applicable to this phase is discussed in the underneath Section 3.5, where the main analysis is done in Chapter 4.

The *model development* phase is where the process of converting the conceptual model into an executable simulation model takes place. It is required that the developed simulation model accurately represents the real-world system and can generate valuable insights for analysis and decision-making. The *verification* and *validation* steps ensure that the model is correct. Verification ensures that the model behaves as intended by the developer, while validation ensures that the model accurately represents the actual system being simulated. These two complementary processes provide confidence in the model's reliability and its ability to produce reliable results for decision-making and analysis purposes. The topics regarding model development, verification, and validation are covered in Chapter 5.

Once the model is verified and validated, the results of the model are gathered and evaluated; this involves *output analysis* and *experimentation*. The output analysis phase entails an analysis of the observations generated by the simulation and drawing meaningful inferences about the modelled system. On the other hand, experimentation involves systematically varying the model inputs and structures to explore and investigate alternative system configurations, enabling the assessment of various scenarios and the impact on system performance behaviour. Output analysis and experimentation topics are present in Chapter 6.

3.5 Probability Distribution Functions and Their Properties

One might question the rationale behind not opting for the more intuitive method of directly incorporating observed real-world data into the developed simulation. An indirect approach is employed, which entails the fitting of a probability distribution for each workstation. Subsequently, random variates are generated from the provided data to drive the simulation, a concept that will be elaborated upon in the following Section 3.6. Several factors contribute to the preference for this indirect approach of identifying distribution fits [1]:

1. Simulations often have extended run times and require numerous replications to ensure statistically valid and accurate conclusions from the results. However, this extensive process quickly depletes the available real-world data needed to support the simulations.

2. At different instances of time, different data is observed, possibly equally representative, leading to various ranges of values (minimum and maximum). If the data that is only extracted at one specific instance is used to drive and evaluate the simulation, then the analysis will be limited to those specific values, preventing the generation of results beyond the observed range. Therefore, leading to biased results.

3. If the sample size is not sufficiently large, the observed data may contain gaps in regions that should be theoretically possible but in which observations did not occur at the specific time. Consequently, these unobserved values would be excluded from the simulation and cannot occur in the model's outcomes.

Hence, it is stated from the literature that it is preferred to identify probability functions to drive the developed simulation, rather than utilising observed real-world data directly.

To effectively implement and utilise probability distribution functions within the developed model, it is important to have an understanding of the functions to be evaluated. This understanding is important to ensure that the data is incorporated correctly and that the identified distribution is relevant to the workstation under consideration [1]. The distribution functions to be evaluated include the normal, uniform, exponential, logistic, beta, lognormal, gamma, and Weibull distributions. The required data analysis and identification of the distribution functions are done in Chapter 4.

3.5.1 Normal Distribution

The normal distribution, also known as the Gaussian distribution is a continuous probability distribution with a symmetric bell-shaped curve [3]. The distribution is mathematically described by its PDF, denoted as $f(x;\mu,\sigma)$, where 'x' represents a random variable, and 'μ' and 'σ' represents the parameters of the distribution. The PDF of the normal distribution is provided by [3]:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where:

x The random variable for which the probability density is calculated.

μ The mean of the normal distribution, determining the centre of the curve.

σ The standard deviation of the distribution, indicating the spread of the data.

$\sqrt{2\pi}$ A constant term ensuring the area under the curve integrates to 1.

The normal distribution is a central concept in statistics, providing a powerful model for a wide range of phenomena [3]. Its symmetric and bell-shaped nature makes it an invaluable tool for data analysis, hypothesis testing, and decision-making across various disciplines [3].

3.5.2 Uniform Distribution

The uniform distribution is a continuous probability distribution that characterises random variables with an equal likelihood of falling within a specified interval [4]. The distribution is commonly denoted as $U(a,b)$, where 'a' and 'b' are the parameters representing the lower and upper bounds of the interval, respectively. This distribution assumes a constant probability density function over the interval [a,b], resulting in a flat, rectangular-shaped curve. The PDF of the uniform distribution $U(a,b)$ is defined as follows [4]:

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{if } a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}$$

where:

x The random variable for which the probability density is calculated.

a, b The lower and upper bounds of the interval, respectively.

$\frac{1}{b-a}$ The constant ensuring the total area under the curve equals 1, satisfying the requirement of a probability density function.

The uniform distribution is a fundamental concept in probability theory, representing random variables with constant probabilities over a defined interval [4].

3.5.3 Exponential Distribution

The exponential distribution is a continuous probability distribution that characterises the time elapsed between consecutive events in a Poisson process - a stochastic process with a constant event rate [5]. The PDF of the exponential distribution is defined as follows [5]:

$$f(x;\lambda) = \lambda e^{-\lambda x},$$

where:

- x The random variable used to determine probability density.
- λ The distribution's rate parameter, often known as the inverse of the mean, represents the typical number of occurrences per unit of time.

As presented in the exponential formulation above, the exponential distribution is characterised by a single parameter, namely the rate parameter (λ). The parameter represents the average rate at which events occur [5]. The exponential distribution's form is determined by the rate parameter, which has a positive value. The larger the parameter value, the more frequently events occur, leading to a steeper drop-off in the PDF and a shorter expected time between events. Conversely, a smaller λ value results in less frequent events, which causes a more progressive drop-off and a longer anticipated interval between events [5].

3.5.4 Logistic Distribution

The logistic distribution is a continuous probability distribution, well-known for its versatile usage in modelling various real-world phenomena [6]. The distribution is characterised by its S-shape curve, resembling the logistic function [6]. The logistic distribution's PDF is defined as follows [6]:

$$f(x;\mu,s) = \frac{1}{s} \cdot \frac{e^{-(x-\mu)/s}}{\left(1+e^{-(x-\mu)/s}\right)^2}$$

where:

- x The random variable for which the probability density is calculated.
- μ The location parameter, determining the centre of the distribution.
- s The scale parameter, governing the spread or width of the distribution.

As observed from the logistic distribution's PDF, the distribution is characterised by two parameters, namely the location (μ) and the scale (s) parameters. The *location parameter* represents the midpoint of the distribution, which is the value where the Cumulative Distribution Function (CDF) reaches 0.5 (50% probability). The parameter determined the horizontal shift of the S-shaped curve along the x-axis. The entire distribution is shifted to the left or right as the parameter alters. If the parameter is 0, the distribution is centred at the origin [6].

The logistic distribution's spread or width, is determined by the *scale parameter*. The parameter has a positive value, that affects the curve's steepness. A steeper curve is produced by a lower value of s, whereas a softer curve is produced by a larger number [6].

3.5.5 Beta Distribution

The beta distribution is a flexible and adaptable continuous probability distribution that may be used for a variety of purposes [7]. For data that is bounded between 0 and 1, such as proportions, probabilities, and percentages, the distribution function is frequently utilised in statistical modelling [7]. The PDF of the beta distribution is defined as follows [7]:

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} \cdot x^{\alpha-1} \cdot (1-x)^{\beta-1}$$

where:

x The random variable for which the probability density is calculated, restricted to a certain interval.

α, β The shape parameters, both greater than 0, which control the shape and skewness of the distribution.

$B(\alpha, \beta)$ The beta function, acting as a normalisation constant to ensure the total area under the curve is 1.

As observed from the beta distribution's PDF, the distribution is characterised by two shape parameters, namely 'α', and 'β'. The distribution's shape is determined by the α parameter, which has a positive value [7]. The α parameter has an impact on the beta distribution's skewness and peak placement. The distribution is skewed to the right and the peak is moved towards to lower end of the range when the parameter is smaller than 1. In contrast, the distribution is skewed to the left, and the peak is moved towards the upper end of the range when the parameter is greater than 1. When the parameter represents a value of 1, the distribution is uniform, with equal probability across the entire range [7].

Similar to α, the shape parameter 'β' has a positive value and affects the beta distribution's peak form, skewness, and placement [7]. The distribution is skewed to the left and the peak is closer to 1 when β is smaller than 1. The distribution is skewed to the right and the peak is closer to zero when β is larger than 1. Also similar to the α parameter, the distribution is uniform when the β value is 1 [7].

The beta distribution is a powerful and flexible probability distribution suitable for modelling continuous random variables with bounded support on the specified interval [7]. The shape parameters 'α' and 'β' allow precise customisation of its characteristics, making the distribution invaluable for statistical analysis and modelling in practical scenarios [7].

3.5.6 Lognormal Distribution

The continuous probability distribution, known as the lognormal distribution is frequently used to describe random variables where the logarithm reflects characteristics of the normal distribution [8]. The lognormal distribution's PDF is defined as follows [8]:

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \cdot e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$$

where:

x The random variable for which the probability density is calculated, restricted to the positive real numbers.

μ, σ The location and scale parameters, respectively, of the underlying normal distribution after taking the logarithm of x.

The two parameters, namely the location parameter 'μ' and the scale parameter 'σ', are necessary to shape the characteristics of the lognormal distribution [8]. The *location parameter* determines the position or location of the peak of the lognormal distribution. The parameter represents the median of the distribution and corresponds to the value of the random variable x at which the probability density is highest. By altering the value of μ, the lognormal distribution shifts along the x-axis without altering its shape. When the μ value increases, the peak of the distribution shifts to the right: alternatively, the peak shifts to the left when the μ parameter decreases [8].

The *scale parameter* governs the width or spread of the lognormal distribution. The parameter established the degree to which the values of the random variable x vary from the location parameter. Given that the parameter reflects the standard deviation of the logarithms of the random variable x, the parameter must be positive. The lognormal distribution takes on a different shape when the value of σ is changed. While a greater value of σ results in a wider and flatter distribution, a lower value of σ produces a narrow and more pronounced peak [8].

3.5.7 Gamma Distribution

Due to its adaptability and suitability for a wide range of real-world settings, the gamma distribution is a continuous probability distribution that is essential to statistical modelling [9]. The distribution is often used in a variety of industries where variables are constrained to be positive [9]. The PDF of the gamma distribution is defined as follows [9]:

$$f(x;k,\theta) = \frac{1}{\Gamma(k)\cdot\theta^k}\cdot x^{k-1}\cdot e^{-\frac{x}{\theta}}$$

where:

- x The random variable for which the probability density is calculated, restricted to the positive real numbers.
- k,θ The shape and scale parameters, respectively, of the distribution.
- $\Gamma(k)$ The gamma function, acting as a normalisation constant to ensure the total area under the curve is 1.

The gamma distribution is a continuous probability distribution that is commonly used to model positive, right-skewed data [9]. As observed from the provided PDF, the distribution is characterised by two shape parameters, 'k' and 'θ'. The gamma distribution's shape is determined by the shape parameter k. The parameter regulates the PDF curve of the distribution. When k is less than 1, the distribution has a long tail to the right and is substantially right-skewed. The distribution becomes more symmetric as k increases and approaches a normal distribution [9]. For integer values of k, the gamma distribution reduces to the Erlang distribution [9].

The scale parameter, denoted by θ, influences the dispersion or spread of the gamma distribution [9]. Similar to the k parameter, the θ parameter is also a positive value. The θ parameter determines the width of the distribution and affects the average value of the distribution. The distribution gets more spread out as the value of θ increases, whereas a smaller θ value produces a distribution that is more concentrated [9].

3.5.8 Weibull Distribution

The Weibull distribution is a continuous probability distribution that is mainly utilised to describe the time-to-failure of components or systems [10]. The distribution also depicts the distribution of lifetimes for goods with different failure rates [10]. The Weibull distribution is a cornerstone in reliability and survival analysis due to the distribution's adaptability and broad use in several sectors [10]. The PDF for the Weibull distribution are described as follows [10]:

$$f(x;k,\lambda) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$$

where:

- x The random variable for which the probability density is calculated, constrained to $x \geq 0$.
- λ The scale parameter, determining the characteristic lifetime of the system or component.
- k The shape parameter, influencing the shape and failure characteristics of the distribution.

As observed from the provided PDF for the Weibull distribution, the distribution is characterised by two parameters. The first parameter is the shape parameter, denoted by 'k'. The positive parameter k represents the shape parameter which affects how the Weibull distribution is represented. The parameter controls how the curve behaves and determines whether the distribution is approximately symmetric, right-skewed, or left-skewed. When k is greater than 1, then the distribution is right-skewed, with a decreasing rate. When k is less than 1, the distribution is left-skewed, with an increasing rate. When k is equal to 1, the Weibull distribution reduces to the exponential distribution, which exhibits a constant rate over time [10].

Furthermore, the second parameter utilised is the scale parameter, denoted by 'λ'. Similar to the k parameter, the λ parameter is a positive value. The parameter influences the spread and scale of the Weibull distribution. Also, the parameter controls the rate at which the values increase or decrease. A broader spread of the distribution, results from larger values of λ, whereas a narrower spread is produced by lower values [10].

It is important to note that the identified distribution functions cannot be used directly in the developed simulation modelling. This is because simulation models are formulated using random variables. Further elaboration on this concept is provided in the subsequent section.

3.6 Randomness in Simulation and Random Variables

Understanding and utilising random variables is necessary for simulation modelling and analysis because unpredictability in simulation models is expressed using random variables [1]. A random variable is a function that derives its value from the outcome of an experiment, where the value is only determined after conducting the experiment. In a simulation model, an experiment involves running the simulation model with specific inputs [1].

The behaviour of a random variable, denoted as X, is characterised by its PDF (or CDF) represented by $F(x) = Pr(X \leq x)$. This function describes the probability that the random variable X will have a value less than or equal to x. In the case of discrete random variables, the probability mass function, $p(x_i)$, is considered[1]:

$$p(x_i) = \Pr(X = x_i)$$
$$F(x) = \sum_{\substack{\forall i \ni \\ x_i \leq x}} p(x_i)$$

On the simulation input side, random variables are characterised, and samples are generated from the corresponding distributions. Further discussion regarding the input data, i.e. the distribution functions, takes place in the next chapter. On the output side, the characteristics of the distributions are analysed based on observations generated by the simulation [1].

3.7 Theory of Constraints

The TOC is a management philosophy and methodology that focuses on identifying and managing the limiting factors (or "constraints") that determine the performance of a system [47]. The goal of TOC is to improve the performance of the entire system by improving the performance of the constraints [47]. Hence, this technique serves as an invaluable tool for identifying opportunities for improvement within the system under investigation.

The TOC was developed by Dr. Eliyahu M. Goldratt in the late 20th century, and the TOC has been widely adopted in manufacturing, logistics, and other types of operations [47]. This philosophy is based on the principle that every system has at least one constraint that limits its performance and that the performance of the entire system is determined by the performance of the identified constraint [47]. The goal of TOC is to identify the constraint and thereafter manage the identified constraint in such a manner as to maximise the performance of the entire system [47].

The five steps of TOC for maximising a system's performance, are as follows [48]:

1. Identify the system's constraints: This involves determining which factor or factors limit the performance of the system. This is often done by analysing data and observing the system in operation.

2. Decide how to exploit the system's constraint: Once the constraint has been identified, the constraint must be managed in such a manner as to maximise the system's performance. This often involves investing in equipment, training, or other resources to increase the capacity or efficiency of the constraint.

3. Subordinate the system to the constraint: All other activities in the system must be aligned with the constraint to improve the performance of the entire system. This often involves investing in equipment, training, or other resources to increase the capacity or efficiency of the constraint.

4. Elevate the system's constraint: After the constraint has been exploited and subordinated, the constraint must be continually monitored and improved to maintain the performance of the entire system. This may involve developing new processes, technologies, or other innovations to further increase the capacity or efficiency of the constraint.

5. Repeating the process: The process of identifying, exploiting, subordinating and elevating the constraint must be repeated regularly to maintain the effective performance of the system.

The TOC principles presents itself as a methodological framework extensively substantiated by literature. This technique is renowned for its systematic approach to identifying and assessing improvement opportunities within systems. Hence TOC principles are to be used adaptively to identify bottlenecks within the system whereafter improvement opportunities can be identified.

3.8 Concluding Remarks: Literature Review

First, the chapter focused on the key concepts of systems and models. The limitations of relying solely on observations to understand complex systems have been recognised, and the practical benefits of using models have been highlighted. Three types of modelling techniques have been identified: physical, analytical, and simulation models. Among these, simulation models stand out as the most suitable approach for addressing the study's objectives, offering a versatile and adaptable means to achieve the research's objectives.

It is further identified that employing a simulation modelling technique that is characterised by dynamic, discrete, and stochastic attributes, will be ideal to effectively address the study's objectives. Hence, the simulation software identified to be utilised to effectively address the study's objectives, and conforms to the mentioned attributes, is Simio Simulation Software [46]. Once the suitable software is identified, a brief discussion unfolds regarding the iterative simulation process necessary for developing the required models effectively.

Thereafter, the applicability of following an intuitive method of utilising distribution functions, rather than using direct real-world data, is elaborated on. Hence, the importance of PDFs in simulation modelling is discussed. Several applicable PDF's are explored and elaborated on. These PDF's include the normal, uniform, exponential, logistic, beta, lognormal, gamma, and Weibull distribution.

Furthermore, the incorporation of randomness in the models was elaborated on. Random variables and the variable's role in expressing unpredictability in simulation modelling are evaluated, with PDFs representing the behaviour of random variables. The integration of PDFs and the introduction of randomness are important factors to ensure that the models accurately reflect real-world pipe manufacturing operations. Finally, the use of TOC principles is explored as an approach to effectively identify bottlenecks within the system in an iterative manner.

Before a discussion takes place regarding the development of the required model to consequently address the study's objectives, it is required that adequate data is identified and evaluated. Hence, the next chapter focuses on the exploration of the relevant data to consequently be used in the developed model.

Chapter 4

Data Analysis

The data that were used in developing solutions to the applicable simulation models are analysed and evaluated in this chapter. Examples and a summary of the analysis are provided along with a discussion of the techniques and methods used to analyse the data. The relevant data are adapted to prevent the dissemination of proprietary design information. However, care was taken to ensure the statistics accurately reflect reality.

4.1 The Raw Data

The raw data obtained from The Company included information from the year 2000 up until 2019. The data is extracted from the Systems Applications and Products (SAP). The SAP software is an extensively used Enterprise Resource Planning (ERP) software that provides organisations with a centralised system that allows each department to access and exchange common data [49]. The data that was extracted included relevant data regarding the manufacturing of 12-metre, 1000 NB pipes.

The data in its raw form was unorganised and inadequate to use. The initial step thus focused on ensuring that all the necessary observations were obtained and listed. Once all the relevant data were gathered, the identified observations were grouped, analysed, cleaned, and observed with the use of RStudio [50]. The RStudio software is an integrated statistical analysis tool, where the R programming language is used for statistical computing and data evaluation [50]. The relevant coding that is done in RStudio can be reviewed in Appendix G.

4.1.1 "Pre-allocated" and "Booked" Processing Times

To gain a comprehensive understanding of the provided data, it is necessary to assess both *pre-allocated* and *booked* processing times for the workstations. The pre-allocated processing time represents the time an employee is expected to complete a task at a specific workstation. In contrast, the booked processing times refer to the actual time taken by the employee to complete the task.

For each workstation, mean values for both pre-allocated and booked processing times are determined with the use of RStudio [50]. This involves evaluating the data for each workstation to determine their respective mean values. The resulting mean values for pre-allocated and booked times, relevant to the various workstations within the pipe manufacturing operation, are presented in Table 4.1 below.

Table 4.1: *Pre-allocated* - and *booked* Mean Processing Times of Workstations

Workstation	Mean Actual Work ('Booked') (hours)	Mean Work ('Pre-allocated') (hours)	Percentage Difference (%)
s_line	1.47	1.15	27.60
dress	1.08	0.82	32.58
roll	1.93	1.93	-0.08
l_weld	2.02	1.17	72.82
roll (re-roll)	1.06	0.99	7.39
assem (9 m)	6.92	5.66	22.34
c_weld (9 m)	8.35	5.49	52.17
assem (12 m)	2.31	1.89	22.34
c_weld (12 m)	2.78	1.83	52.17
inspect	4.92	2.65	85.66
disp	0.31	0.30	2.75

The percentage difference, as listed in Table 4.1, represents the percentage variance between the pre-allocated and booked times. This variance indicates the potential for improvement in the operation at each specific workstation. If the booked times deviate significantly from the pre-allocated processing times, it indicates an opportunity to implement strategies for improvement.

Due to the availability and nature of the data applicable to factors such as transportation times between workstations and the maintenance times on the machines, a more direct approach was followed. The relevant data applicable to the additional factors present within the system is further discussed in Chapter 5.

As stated in the literature (please refer to Section 3.5), it is beneficial to identify distribution functions to drive the developed simulation model. The parameters, of which the distribution functions must be determined, include the processing time of the relevant workstations. The relevant distribution functions are identified and determined with the use of Exploratory Data Analysis (EDA).

4.2 Exploratory Data Analysis

The EDA technique is the critical process of doing preliminary analyses on data to find patterns, identify anomalies, test hypotheses, and evaluate assumptions with the aid of summary statistics and graphical representations [51]. The EDA approach is used to analyse the applicable data, where the main goal is to estimate the probability distributions of the parameters [51].

During EDA, three necessary steps were followed: estimating the distributions' parameters in step one, hypothesising the distribution or its family in step two, and assessing the quality of the fit in step three using visualisation and the analyses of the goodness of fit test. In conjunction with developing programming code to adhere to the necessary data analysis, using RStudio's built-in functions [50], assisted in achieving the mentioned steps' goals.

To exemplify the process of performing EDA for obtaining required distribution fits the instance of identifying the distribution fit for the *s_line* workstation's processing time is elaborated on. It is imperative to note that the same analysis was conducted to determine all the workstations' required distribution fits. Furthermore, it is essential to emphasise that the data reflecting processing times at each workstation encompasses more than just the workstation's inherent processing duration. In essence, a "buffer" time has been incorporated to account for various activities such as waiting for the crane, tool retrieval, and similar factors.

Before data evaluation took place, the relevant datasets were cleaned, pre-processed, and formatted

accordingly to adhere to the specific evaluation task. As data visualisation is an essential step during EDA, the data related to the *s_line* workstation's processing time were first plotted as a histogram. This was done to obtain an overview of the data and to make a distributional hypothesis [52]. The developed histogram is displayed in Figure 4.1 below. From this developed graph it can be hypothesised that a normal or even a lognormal distribution can be a possible good fit for the data. Literature regarding the possible distribution fits, as well as the relevant parameters, was already discussed in Section 3.5, which is present in Chapter 3.

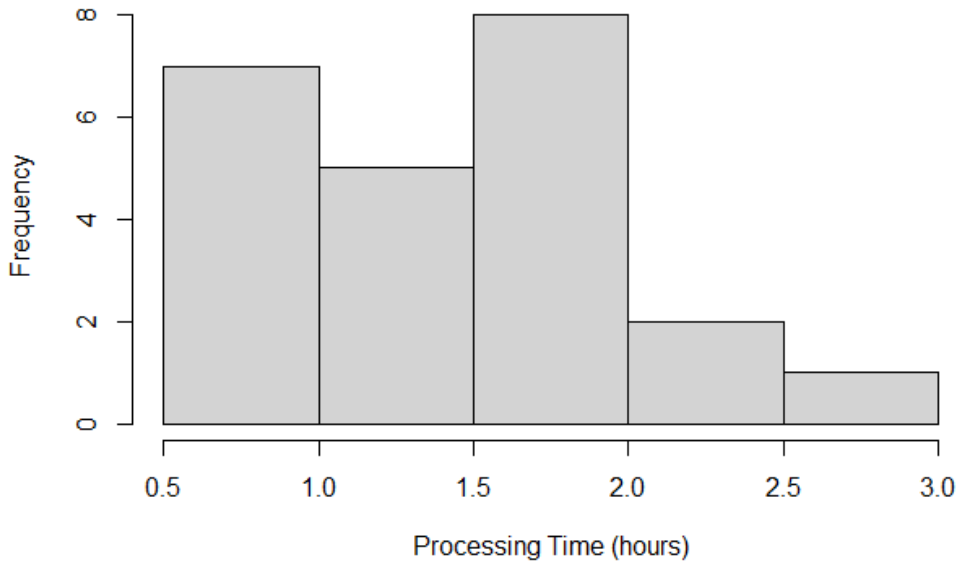


Figure 4.1: Histogram of *s_line* Operation's Processing Time

Further EDA took place by visualising the empirical density and cumulative distribution of the *s_line* workstation's processing time dataset, as can be seen in Figure 4.2 below. The empirical density graph is equivalent to the histogram, depicting the density of observations, whereas the cumulative distribution graph is the accumulation of these data points' probabilities [53] [54]. The dotted line on the empirical distribution graph refers to the distribution of observed data points in the provided dataset. As hypothesised above, it can be visualised that a normal or lognormal distribution may be a possible good fit for the dataset. Within the cumulative distribution graph, the added densities of the observations are displayed where a gradual change within the dataset is visualised.

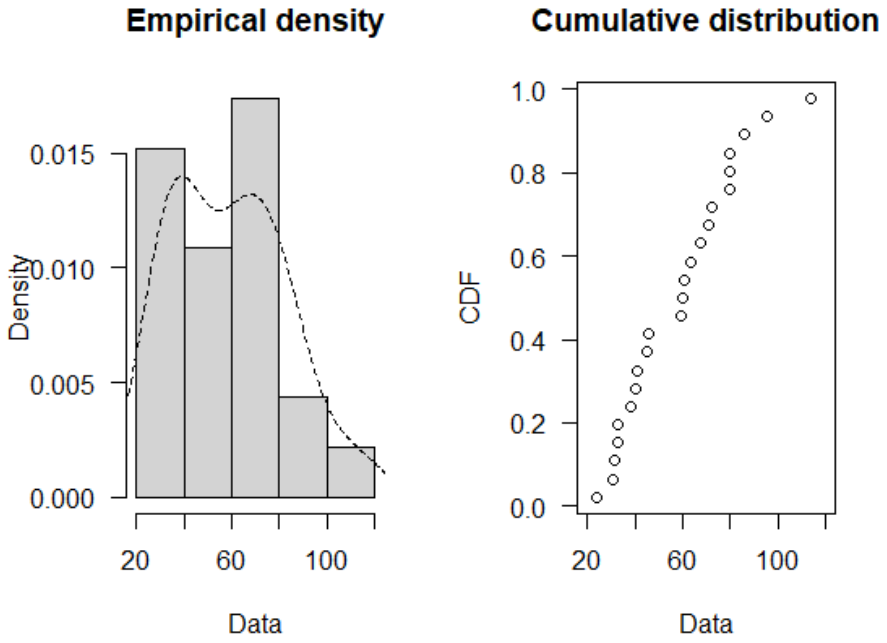


Figure 4.2: Empirical density- and cumulative distribution plot of *s_line* Operation's Processing Time

The data were further analysed by displaying the dataset on a Cullen and Frey graph, as seen in Figure 4.3 below. The Cullen and Frey graph is a common tool for data analysis, which is used to identify potential distributions for the data [55]. The theoretical distributions that are evaluated for the dataset, as presented in the top right corner of the graph, include the normal, uniform, exponential, logistic, beta, lognormal, gamma, and Weibull distributions. An elaboration regarding these datasets took place in Section 3.5.

The graph in Figure 4.3 is the square of the skewness of the data plotted against the kurtosis of the data. Skewness is a metric for the absence of symmetry, i.e. the skewness is a measure of how symmetrical the data are: the higher the square of the skewness, the less symmetrical the data is. Kurtosis is a measure of tailedness, i.e. the kurtosis indicates whether the data has a heavy or light tail. Data that are more heavily tailed are those with a greater kurtosis value [55].

The data being observed is displayed as the blue dot on the Cullen and Frey graph. The yellow dots on the graph are bootstrapped values. An estimation of the parameters of the data can be made by using the statistical analysis tool of bootstrapping, which resamples a single dataset to create many datasets [55]. Bootstrapping is used to improve distribution estimation, where 1000 bootstrapped values are illustrated on the Cullen and Frey graph.

The data observation displays an approximate square of skewness between 0 and 1, which indicates a reasonably symmetric distribution. Furthermore, the kurtosis is grouped between 1 and 4, with the observed dataset being present between 2 and 3. From the graph, it would appear that the normal distribution, or even the gamma distribution, may be a good fit. This observation is based on the observed dataset and the bootstrapped values.

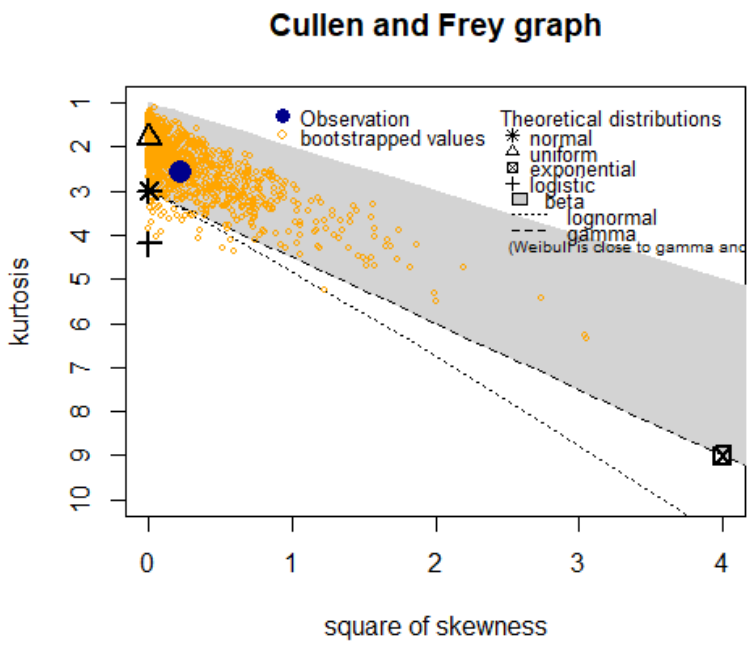


Figure 4.3: Cullen and Frey graph of Straight line cutting

From the Cullen and Frey graph, the three most likely distribution fits are identified. Further analysis is done on the three identified distributions to determine the 'best' distribution fit. The three identified distributions include the normal-, lognormal-, and gamma distributions.

Further EDA took place by evaluating a plot that reflects the combination of a histogram and a theoretical density plot, as can be seen in Figure 4.4 below. The purpose of this plot is to obtain an understanding of the possible distribution fits with the dataset under consideration. The three identified distributions are depicted in the plot. The histogram is displayed to illustrate how the numerical data is distributed. The theoretical density plot, for each distribution respectively, visualises a curve that depicts the PDF of a continuous random variable [56] [52].

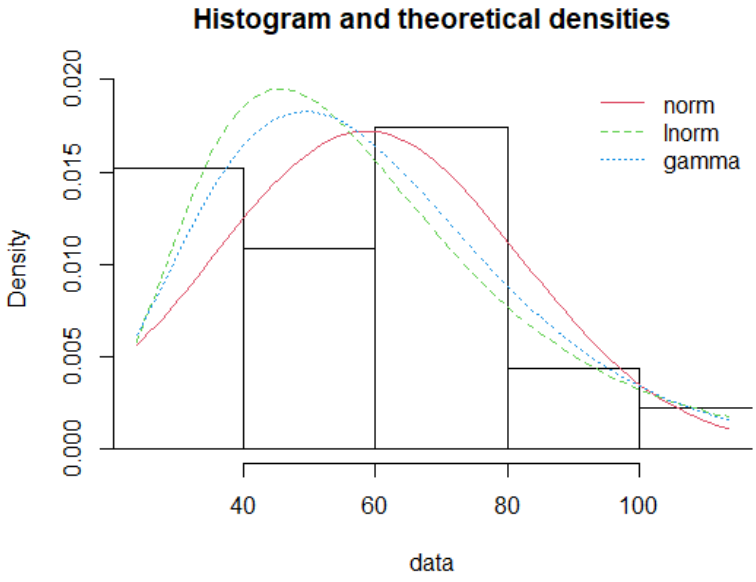


Figure 4.4: Histogram and theoretical densities plot for *s_line* Operation's Processing Times

By visually displaying the combination of a histogram and a theoretical density plot, insight can be obtained by comparing the dataset to the known theoretical distributions. Thus, the better the distribution curve follows the histogram's pattern, the better the data fits the distribution under evaluation. From Figure 4.4 above, it can be seen that the distributions under evaluation, follow the histogram-pattern accordingly. One might suggest that the normal or the gamma distribution is a 'better' possible distribution fit for the dataset.

The next plot used for EDA is a quantile-quantile (Q-Q) plot. A Q-Q plot is a graphical tool for determining if a group of observations can be fairly described as coming from a certain distribution [57]. From Figure 4.5 below, and determined from the Cullen and Frey graph in Figure 4.3 above, it can be seen that the distributions under investigation include the normal, lognormal and gamma distribution. The quantiles of the observed data are compared to the quantiles of the predicted distributions [57]. The closer the specific distribution fits the dataset under consideration, the closer the distribution's colour point will be to the diagonal black line [57].

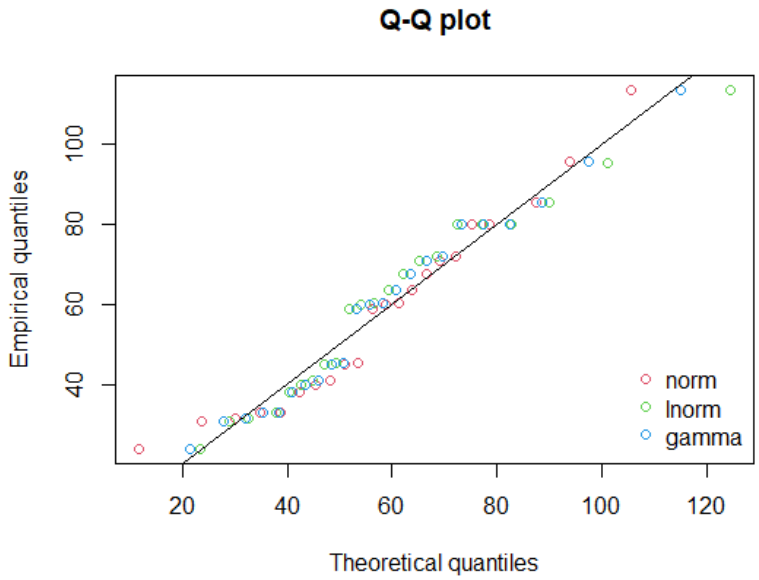


Figure 4.5: Q-Q plot for Straight Line Cutting Operation's Processing Times

From Figure 4.5, it would seem that all the three distributions fall relatively close to the black line, which thus indicates that all the three distributions are a relatively good fit for the dataset being evaluated. It can also be further analysed that the gamma distribution would be the closest to the straight line: therefore, assuming that, between the compared distributions, the gamma distribution is the possible 'best' distribution fit.

The *Chi-squared* (χ^2) *test* was applied to formalise the interpretation of the fitted distribution from a hypothesis test and to quantify the estimation. As a goodness of fit test, the χ^2 test is utilised to determine how well the estimated distribution fits the empirical distribution [58]. The χ^2 test is a hypothesis test, with the null hypothesis being that there is no significant difference between the estimated and the empirical distributions. The alternative hypothesis is that there is a significant difference between the distributions under investigation. The test was carried out in R [50] using the built-in functionality. The probability value is the statistic of interest from this test, where a significance level (α) of 0.5 was used. The null hypothesis is rejected if the probability value is less than the significance level, and if it is higher than the significance level, the null hypothesis is not rejected.

For the *s_line* workstation's processing time, the χ^2 test was applied to the identified three distributions. After evaluation, the gamma distribution is identified as the 'best' distribution fit for the

dataset under investigation. The results obtained, by applying the χ^2 test to the various datasets, can be observed in Table G.1 in Appendix G.

During the goodness of fit test for the gamma distribution, the probability value from the χ^2 test was 0.203, which is greater than α, resulting in the null hypothesis failing to be rejected. Hence, from identifying the parameters, with the functionality and build-in functions of RStudio, it can therefore be assumed that $\sim G(\alpha = 6.13, \beta = 0.24)$ is a decent approximation of the distribution of the processing time for the *s_line* workstation. This fitted distribution is displayed in Figure 4.6 below. It must be noted that the χ^2 test is not the only goodness of fit test, with the *Anderson-Darling*, *Kolmogorov-Smirnov*, and the *Shapiro-Wilk* tests also being used to interpret and quantify results of a hypothesis test [59] [60] [61].

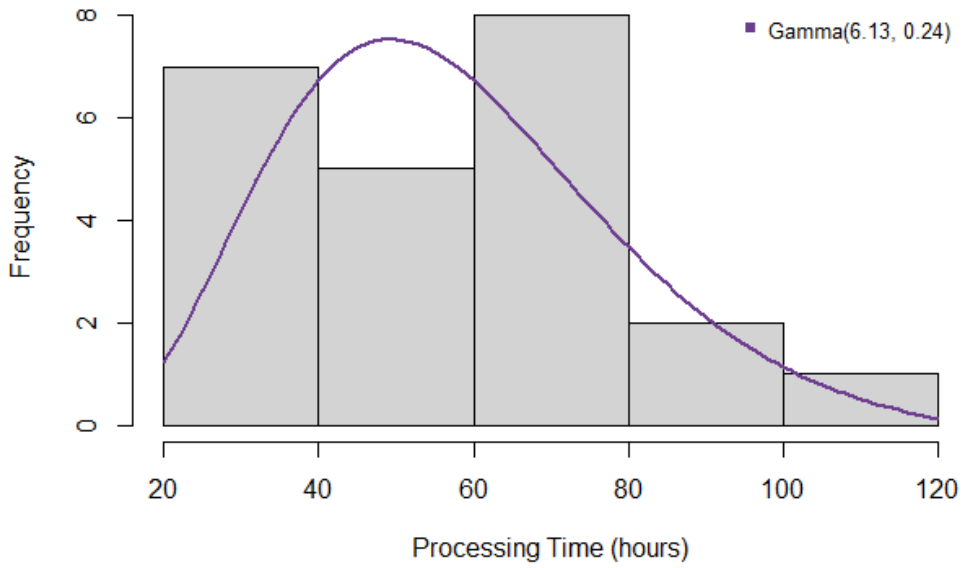


Figure 4.6: Fitted Distribution of Processing Time for the *s_line* Workstation

In essence, EDA is used to estimate probability distributions for the relevant datasets under consideration. The focus was on determining distributions for the processing times of all the workstations related to the pipe manufacturing operation.

However, it was not possible to establish a suitable distribution fit for the processing times at all workstations within the pipe manufacturing operation. This could be due to the complexity and diversity of the data, the presence of outliers or anomalies, small sample sizes, non-parametric distributions, biases in data collection methods, variability in underlying processes, and other related factors [1]. The handling of the instance where no distribution fit is possible is discussed in the following subsection.

4.2.1 No Distribution Fit

As demonstrated above, it is common practice to utilise probability distributions as a means to comprehend the properties of datasets. However, there are instances when a dataset exhibits inadequate conformity to any specific distribution, challenging established assumptions and necessitating alternative approaches for analysis. The identification that the dataset under investigation does not fit any of the identified distributions means that the "standard" list of distributions fails to have an entry that accommodates the data under analysis [1].

To resolve this instance, where a suitable distribution fit cannot be identified for the dataset, the software Stat::Fit is utilised [62]. Stat::Fit is a software tool used for statistical analysis and fitting of data to various probability distributions, allowing users to determine the 'best-fitting' distribution for the dataset under investigation and make predictions based on the fitted model [62].

In the case where an 'ideal' distribution fit cannot be identified, Stat::Fit can produce an empirical distribution, which represents a version of the histogram of the dataset, where variates can be generated from the identified histogram in the Simio Simulation [1]. Note that the relevant dataset is firstly adequately cleaned and analysed within RStudio [50]. After the required EDA is done and no adequate distribution fit can be identified, the relevant dataset is imported into Stat::Fit where all the necessary proceedings are followed to identify an adequate empirical distribution [1].

Within Stat::Fit, a sequence of pairs is identified. These pairs consist of v_i, c_i, where v_i is the ith smallest value of the dataset under investigation. The value c_i is the *cumulative* probability of generating a variate that is *less than or equal to* the corresponding v_i. The identified values must consequently be utilised within the simulation software by implementing either a discrete or continuous distribution. The arguments for both the discrete and continuous distribution are as follows [1]:

- **Discrete:** The adequate Simio expression is *Random.Discrete(v1, c1, v2, c2,...)* where the sequence of pairs can go on as long as needed, depending on the dataset under investigation. Each value v_i will be generated with cumulative probability c_i.
- **Continuous:** The expression to be used is *Random.Continuous(v1, c1, v2, c2,...)*. The CDF from which the values are generated will pass through each (v_i, c_i) and will be connected with straight lines. This will result in a piecewise-linear "connect-the-dots" CDF, rising from 0 at v_1 to 1 at the largest v_i.

Due to the nature of the modelled system and the objectives of the study, it is identified that the *discrete distribution* is to be utilised. The instances where a distribution could not be identified by applying EDA, but where a distribution is identified using the aforementioned procedure, includes the *dress*, *assem (12 m)*, and *disp* workstations' processing times.

4.2.2 Distribution Data

It is required to determine the workstations' processing time distribution functions to consequently take uncertainty into account and reflect the real-world system. The identified distribution functions will serve as input to the developed model, to be discussed in the next chapter.

The aforementioned approach that was followed to determine the 'ideal' distribution fit, was explained by using the *s_line* workstations' processing time as an example. This approach is, however, not only used to determine the one workstations' distribution fit, but the same procedure was done for all the workstations. Hence, the determined distribution fit relevant to each applicable workstation are provided in Table 4.2 below.

Table 4.2: Distribution Fits Related to Workstations' Processing Time

Workstation	Distribution Fit Related to Processing Time
plate_store	-
s_line	Gamma(6.13, 0.24)
dress	Discrete(0.425, 0, 0.881, 0.536, 1.34, 0.75, 1.79, 0.893, 2.25, 1)
roll	Gamma(8.45, 0.23)
re-roll	Gamma(6.72, 0.16)
l_weld	Weibull(2.22, 2.29)
assem (9-metre)	Gamma(8.84, 0.78)
c_weld (9-metre)	Gamma(9.46, 0.88)
assem (12-metre)	Discrete(1.08, 0, 1.94, 0.346, 2.81, 0.769, 3.67, 0.962, 4.54, 1)
c_weld (12-metre)	Lognormal(0.97, 0.32)
inspect	Weibull(1.68, 5.55)
disp	Discrete(0.1, 0, 0.3, 0.567, 0.5, 0.633, 0.7, 0.967, 0.9, 1)
Finished	-

The relevant hypothesis test results obtained to effectively identify the 'ideal' distribution fit for all the workstations is displayed in Table G.1 which is present in Appendix G.

4.3 Concluding Remarks: Data Analysis

The chapter commences with a discussion regarding the data that was used for the analysis. It was stated that the SAP software was used to extract data from the year 2000 to 2019, regarding the 12-metre, 1000 NB pipes. Furthermore, it was emphasised that all data cleansing and analysis took place with the use of RStudio.

The EDA approach was followed to determine the applicable probability distribution for each workstation's processing time. Firstly, to obtain better insight and understanding concerning the dataset under investigation, a histogram of the dataset was displayed. Thereafter, the dataset was depicted on an empirical density and cumulative distribution plot, respectively. This was done to obtain a further understanding of what the dataset entails and to develop a hypothesis of possible distribution fits. To reinforce the hypothesis, the data were plotted in a Cullen and Frey graph [55]. The graph was used to identify three possible distributions that may be a good fit for the dataset under investigation. Further EDA took place by evaluating the three identified distribution fits using a histogram and theoretical density plot, as well as a Q-Q plot. The identified distributions were analysed and evaluated with the use of these EDA plots. The plots assisted in emphasising the developed hypothesis.

To formalise the interpretation of the identified possible distributions, the (χ^2) *test* was applied. The χ^2 was determined for each of the three identified distributions of a specific parameter where the distribution with the highest probability value was chosen as the 'best' distribution fit for the parameter under investigation. The determined χ^2 for each identified distribution for the various workstations' processing times can be seen in Table G.1.

In some instances, it was challenging to identify a suitable probability distribution that accurately described the processing time for certain workstations. These instances raised questions about the applicability of standard distribution models, prompting the utilisation of the Stat::Fit software. This software played an important role in fitting and generating an empirical distribution when an applicable distribution cannot be identified.

Although the data were used in various manners, there are two main uses thereof. The first is the application of the data to the formulation of a discrete simulation model. The second includes assistance in identifying possible improvement opportunities during the intervention to deal with uncertainty, which is further elaborated on in Chapter 6.

To be able to effectively identify improvement initiatives, it is imperative to first develop a model that reflects the current proceedings and operations of the pipe manufacturing system. The development of this model, which accurately represents the current operations, is elaborated on in the next chapter.

Chapter 5

Model Development

The purpose of this chapter is to provide a comprehensive overview regarding the successful development of the simulation model that reflects the real-world proceedings of the pipe manufacturing system. This chapter begins by explaining the key concepts of Simio entities, and attributes and introduces essential objects from the Simio Standard Library. Subsequently, the discussion continues regarding the development of the pipe manufacturing's current state in the Simio Simulation Software. The data examined in Chapter 4, along with pertinent supplementary data, is used to effectively drive the model. Finally, the chapter concludes with a comprehensive process of verification and validation to guarantee the model's accurate development.

5.1 Fundamentals of Model Development in Simio

This section provides fundamentals regarding the development of the required model in Simio. The required concepts regarding entities and attributes are explored, and the concept of the applicable Simio object libraries is introduced.

5.1.1 Entities and Attributes

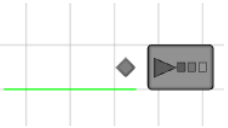
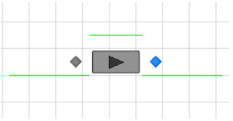
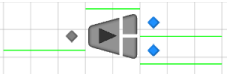
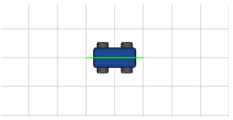
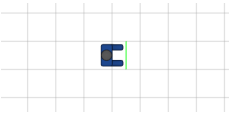
In Simio simulation software, an entity represents discrete objects that move through a simulated system, such as products in a manufacturing process or customers in a service queue. Entities can be created, move through the model, undergo processes and carry attributes, allowing users to model and analyse complex systems [1].

An entity is essentially a concept that requires a precise definition. These entities can take on dynamic characteristics as they move through the system, or they can remain static while playing crucial service roles for other entities. What sets each entity apart is its unique set of attributes. These attributes come with values that distinctly define a particular entity, and these values are exclusive to the specific entity [1].

5.1.2 Object Libraries

Simio object libraries are sets of object definitions that are frequently related to a common modelling theme or domain [1]. The main objects, that are required to develop the pipe manufacturing system within Simio are listed in Table 5.1 below. If deemed necessary, further elaboration regarding the utilised standard library objects is provided in Appendix H.

Table 5.1: Simio Standard Library Objects

Object	Description	Visual Demonstration
Source	An object used in systems simulation to create entities at a specified rate, based on an arrival pattern, triggered by an event.	
Sink	Collect and remove entities from the simulation model.	
Server	Represent resources that provide service to entities.	
Combiner	Merge multiple incoming flows of entities into a single flow.	
Separator	Split an incoming flow of entities into multiple separate flows based on specified criteria.	
Vehicle	Model and simulate the movement and transportation of entities.	
Worker	Model and simulate the behaviour and actions of human or automated workers.	
Basic Node	Represent locations or points.	
Transfer Node	Represent locations where entities can be transferred from one network or path to another within a simulation model.	
Path	Represent the routes or connections between locations.	-

5.2 Model-0.0: Current State

Having established an introduction to Simio's entities, attributes and relevant object libraries, the focus of this chapter shifts toward the development of the current state of the pipe manufacturing operation. To effectively achieve the objectives of this chapter, the discussion commences by briefly identifying the intricate behaviour of Simio by incorporating the system's floor layout. Thereafter, the required properties related to each workstation is identified and the necessary information is elaborated on. Once the essential workstation properties are established, network paths are identified, and material flow is highlighted. To effectively identify the proceedings of the material flow through the system, the presence of symbol transformation is introduced. Furthermore, the utilisation and necessity of worker objects are modelled and discussed. Finally, the applicability and importance of the simulation run-time are identified and elaborated on. To summarise the developed model, an Object Flow Diagram (OFD) is depicted that takes the development of the simulation model together.

Subsequently, upon the completion of the simulation model, it becomes imperative to assess its accuracy and correctness. Accordingly, this chapter concludes by applying comprehensive validation and verification techniques. This is done to validate the model's reliability and establish it as an effective model for identifying improvement opportunities, as done in the next chapter.

5.2.1 Floor Layout Visualisation

Before the relevant objects are identified and developed in the model, it is beneficial to first analyse the floor layout on which the model is built. Importing the system's floor layout is an important aspect of developing a realistic and accurate model. The floor layout gives a representation of the system's flow and operations, enabling Simio to accurately replicate real-world interactions and constraints [1].

Firstly, by accurately representing the placement of the machines, workstations and other critical elements, the model can capture the relationships between these entities and replicate their interactions. Secondly, incorporating a floor layout into the simulation model enhances the visual representation of the model. The floor layout used in the model is depicted in Figure B.2, present in Appendix B.1.

5.2.2 Workstations' Properties

Once the appropriate floor layout has been successfully imported to scale into Simio, the development of workstations and their associated properties took place.

Although each workstation has its unique objectives and operational attributes, there are some common elements that are present. The first includes that each workstation is identified by means of a specific Object Library, where a discussion regarding this characteristic took place in Subsection 5.1.2 above. The second element includes that each workstation has its unique processing time, as addressed and determined in Chapter 4, and summarised in Table 4.2. The object library and processing time, relevant to each workstation, are listed in Table 5.2 below.

Table 5.2: Standard Library and Processing Time Properties of Workstations

Workstation	Object Library	Processing Time
plate_store	Source	-
s_line	Separator	Gamma(6.13, 0.24)
dress	Server	Discrete (0.425, 0, 0.881, 0.536, 1.34, 0.75, 1.79, 0.893, 2.25, 1)
roll	Server	Gamma(8.45, 0.23)
re-roll	Server	Gamma(6.72, 0.16)
l_weld	Server	Weibull(2.22, 2.29)
assem (9-metre)	Combiner	Gamma(8.84, 0.78)
c_weld (9-metre)	Server	Gamma(9.46, 0.88)
assem (12-metre)	Combiner	Discrete (1.08, 0, 1.94, 0.346, 2.81, 0.769, 3.67, 0.962, 4.54, 1)
c_weld (12-metre)	Server	Lognormal(0.97, 0.32)
inspect	Server	Weibull(1.68, 5.55)
disp	Server	Discrete (0.1, 0, 0.3, 0.567, 0.5, 0.633, 0.7, 0.967, 0.9, 1)
Finished	Sink	-

Moreover, a third consistent element across all workstations is the assignment of a single worker object. Additional details regarding this aspect are provided in the Subsection 5.2.6 below.

The three mentioned characteristics are present at all workstations. However, additional aspects, arguments, and properties need to be defined for some of the workstations to effectively develop the respective workstations' objectives. Hence, the workstations that have additional characteristics include the *plate_store*, *s_line*, *roll*, *l_weld*, *assem*, and *c_weld* workstations. Further discussion regarding the development of the additional factors at each relevant workstation is discussed in the underneath subsections.

Plate Store
The additional aspects related to the *plate_store* workstation, include:

- The *Arrival Mode*, i.e. the method used by the Source object to generate a stream of entity arrivals, namely the plates, is defined to have an 'Interarrival Time'. The Interarrival Time states that entities are to arrive at the rate or time specified [1]. It is defined that 5 entities must arrive every 10 hours.
- The maximum arrivals, i.e. the limit on the total number of arrivals to be automatically generated by the Source object during the simulation run, is identified to be 100 arrivals.

The aforementioned arguments are identified to ensure that there is no shortage of input material, namely the plates, to go through the system. This assumption is necessary to adhere to the *Throughput* evaluation criteria, which is further discussed in Chapter 6. Furthermore, it is important to note that the attributes of all the plates developed by the source are the same.

Straight Line Cutting
Additional assumptions and arguments that are built into Simio and are present at the *s_line* workstation include the following:

- One model entity is divided into two model entities, hence the Separator object is identified.
- Divided entity is emphasised with symbol transformation. This concept is elaborated on in the underneath Section 5.2.5.
- Replenishment of inventory takes place. This argument stems from the argument that inventory at the *s_line* workstation needs to be replenished with twenty plates when the inventory stock reaches five entities (plates). The argument is developed within Simio as follows:
  - **Material Element**: Within the Simio model, a "Material Element" is introduced to define materials consumed or produced in the system, with a focus on plates.
  - **Timer Element**: To regulate the periodic checks of plate inventory, a "Timer Element" is incorporated into the model. This timer dictates that plate inventory assessments occur every hour, a frequency determined by experienced personnel.
  - **Inventory Element**: This element is responsible for overseeing plate inventory at the *s_line* workstation. Key operational features of this element include:
    - At the start of the simulation run, the initial inventory at the *s_line* workstation is set to zero.
    - Inventory evaluations occur on an hourly basis, guided by the "Timer Element."
    - When the inventory level reaches five plates, the element ensures that the inventory is replenished to its standard level of twenty plates, ensuring consistent inventory levels.

This well-structured approach ensures that the optimal inventory levels at the *s_line* workstation, effectively preventing operational setbacks due to plate shortages.

To effectively ensure that plates are replenished accordingly, it is required that a process is defined. A process is developed within the *process window*, which is a platform that allows the Simio-user to establish a sequence of actions executed by steps, alerting the state of various model elements [1].

The *PlatesInventory_OnReplenishmentOrder* process is developed to evidently ensure that the plates are replenished. The process consists of two steps, namely a *Delay* and *Produce* step (refer to Figure 5.1 below). The arguments for each of the two steps are as follows:

- **Delay:** The *Delay* step serves to indicate the brief period allocated for restocking the material. This allocated time is exceedingly short.

- **Produce:** This step ensures that the correct material element is re-stocked at the *s_line* workstation. This step also ensures that the correct amount of material is re-stocked, i.e. a total of twenty plates are replenished.

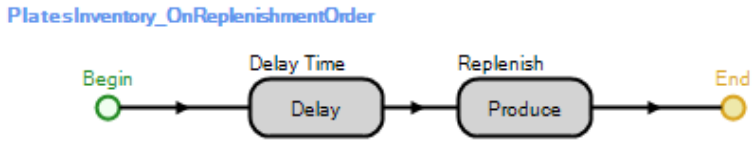


Figure 5.1: Process Related to Inventory Replenishment - *PlatesInventory_OnReplenishmentOrder*

Roll

To effectively adhere to the two, similar operations that take place at the *roll* workstation, the following arguments were incorporated into the Simio model:

- The required Boolean State Variable, known as *Roll*, was developed which is related to the model's entity, namely *ModelEntity* [1]. The identified Boolean variable serves as an identification of whether the entity must adhere to the process of rolling or re-rolling a canister. It is important to take note that the initial state of the Boolean variable is identified as '*True*'.
- Following the establishment of the *Roll* Boolean-variable, the subsequent phase of developing the relevant process, namely *roll_Processing*, was done. A visual representation of the process can be observed in Figure 5.2 below, while the properties of each step are provided in Appendix I.

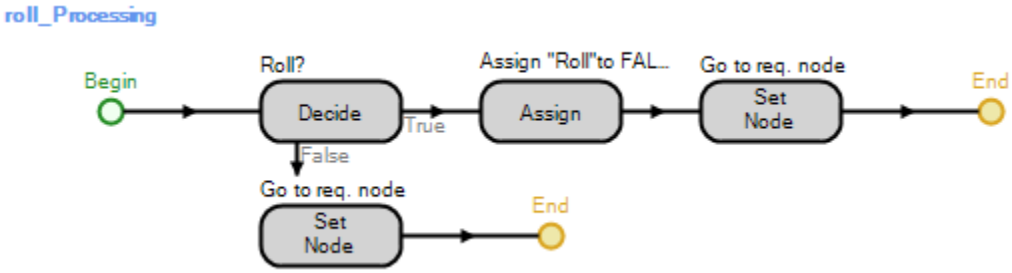


Figure 5.2: Process that Determines Roll or Re-roll Operation at *roll* Workstation - *roll_Processing*

To effectively demonstrate the proceedings of the *roll_Processing* process, a schematic diagram is developed, as presented in Figure 5.3 below.

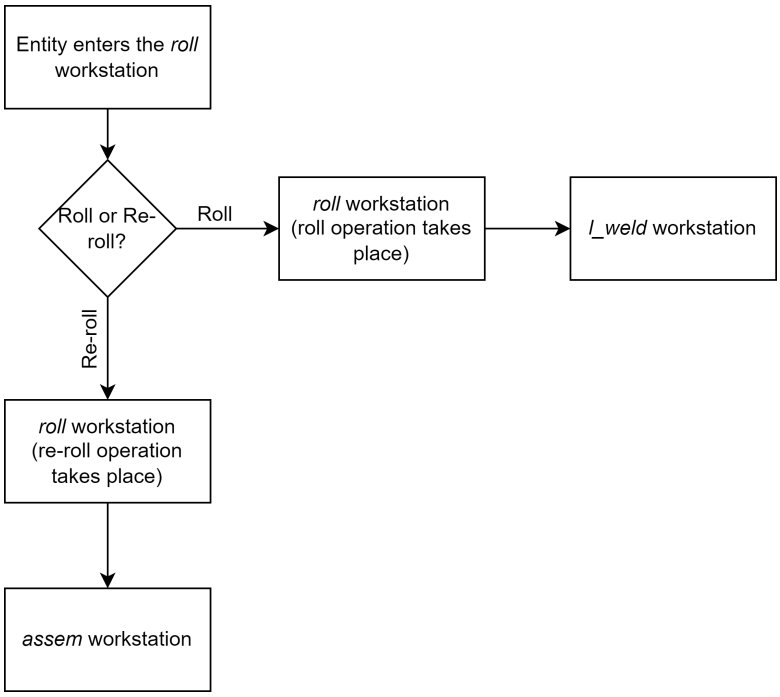


Figure 5.3: Schematic Representation of Process that Determines Roll or Re-roll Operation at *roll* Workstation

Hence, the proceedings of the *roll_Processing* process is as follows:

1. When the entity enters the *roll* workstation, it must be determined whether the entity must conform to the requirement to be rolled or re-rolled.
2. If the *Roll* Boolean-variable is identified as a value of True, then this instance states that the material is still a plate and must be rolled into a canister. Otherwise, the canister adheres to the re-roll operation.
3. If the entity already adhered to the roll operation at the *roll* workstation, then after the relevant operations, the entity must be transferred to the *l_weld* workstation.
4. Else, if the entity is identified as conformed to the re-roll operation, then the entity is transferred to the *assem* workstation.

- The two operations at the *roll* workstation each have its unique processing time. This is due to each operation having different proceedings. The distribution function for each operation's processing time was determined, as elaborated on in Chapter 4, and each operation's applicable distribution fit can be observed in Table 4.2 and/or Table 5.2. To comply with, and simulate, the different processing times within one Server object, the *Math.If* functionality is used. The function is briefly defined as follows:

$$Math.if(test1,\ value1,\ test2,\ value2,\ test3,\ value3, ...,\ otherwise\ value), \tag{5.1}$$

where the function returns the value specified in *valueN* if the *testN* value is True. The function returns the value specified in *otherwise value* if all the *testN* values are False.

For example the processing time at the *roll* workstation is defined as:

$$Math.If(ModelEntity.Roll == True, Random.Gamma(8.45, 0.23), Random.Gamma(6.72, 0.16),$$

where the property states that if the *Roll* Boolean-variable is assigned a value of True, i.e. the roll operation must take place, then the processing time is assigned to be *Gamma(8.45, 0.23)*. Else, the *Roll* Boolean-variable would have a value of False, therefore identifying that the re-rolling

operation must be adhered to, which has a processing time of *Gamma(6.72, 0.16)*. As identified and explained in the literature, the identified *Random* input is to adhere to randomness and consequently reflect real-world operations. Please refer to Section 3.6 for a more clear explanation regarding the incorporation of randomness in the simulating model.

- Symbol transformation occurs at the *roll* workstation, transitioning from a plate to a canister as a result of the operations performed at the *roll* workstation. Further discussion regarding the symbol transformation takes place in the Subsection 5.2.5 below.

Long-weld

An additional characteristic of the *l_weld* workstation pertains to the post-processing phase, where the output location for the entities is divided accordingly. As already mentioned in Subsection 2.2.3, it is assumed that 40% of canisters need to be re-rolled at the *roll* workstation, whereas the remaining 60% are transferred directly to the *assem* workstation. Further elaboration on the implementation of this argument, within Simio, takes place in Subsection 5.2.4 below.

Assemble and Circular-weld

The main additional feature of the *assem* and *c_weld* workstations is that the two workstations are interactively dependent on one another. To effectively demonstrate the proceedings of the interconnectedness of the two workstations, it is beneficial to first provide an overview regarding the relevant objects that are identified in Simio to reflect the two workstations. The library objectives and relevant links - to be further explored later on in the dissertation- are visually represented in Figure 5.4 below.

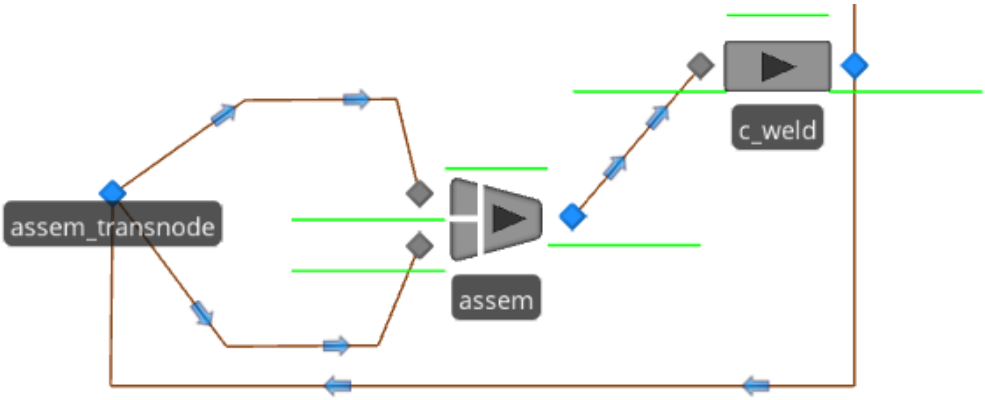


Figure 5.4: assem and c_weld Setup within Simio

Relevant information regarding the setup of the *assem* and *c_weld* workstations within Simio is as follows:

- A transfer node, namely *assem_transnode*, as observed in Figure 5.4 above, is identified. This node is where canisters from the *l_weld* and *roll* workstations enter to consequently be processed by the *assem* workstation. The node is allocated to serve as a location where the required decision-making and/ or steps must take place to evidently adhere to the interconnectedness of the two workstations. A discussion regarding the decision-making takes place underneath, during the evaluation of the *assem_transnode_Entered* process.

- A Combiner object represents the *assem* workstation. As elaborated on in section 2.2.3, the *assem* workstation first delivers a 9-metre assembly, whereafter an additional canister is joined to deliver a 12-metre assembly.

  To obtain the objective of first delivering a 9-metre assembly, one entity must be present at the parent node and two entities must be present at the member node. The mentioned nodes are clarified in Figure 5.5 below.

Figure 5.5: Combiner Object's Node Clarification

To deliver a 12-metre assembly, the relevant 9-metre pipe must be present at the parent node and the additional entity (canister) must be present at the member node. Further elaboration regarding this concept is elaborated on underneath during the explanation of the *assem_transnode_Entered* process.

- A Server object is placed after the *assem* workstation, which is used to adhere to all the requirements of the *c_weld* workstation. After the required proceedings took place at the *c_weld* workstation to evidently deliver a 12-metre pipe, the entity (12-metre pipe) is transferred to the next workstation, namely the *inspect* workstation.

A schematic representation of the proceedings followed by the *assem* and *c_weld* workstation, are presented in Figure 5.6 below. This representation serves as a means to assist during the explanation regarding the development of the interconnectedness of the two workstations. Elaboration of the workstations' operations have already been discussed in Subsection 2.2.3.

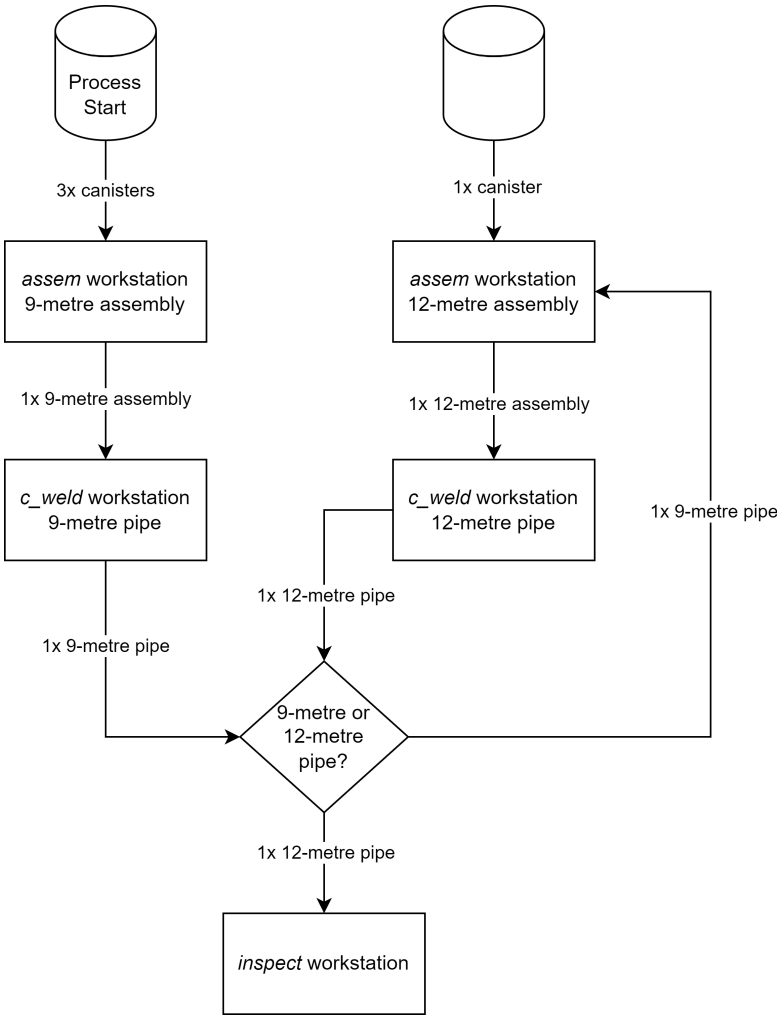


Figure 5.6: Schematic Representation of Operations at *assem*- and *c_weld* Workstations

The complexity of identifying whether the entity is a canister, a 9-metre assembly/ pipe, or a 12-metre assembly/ pipe, and what must be done with the identified entities, are built into the model by specifying a process, namely *assem_transnode_Entered*. This process is triggered at the *assem_transnode* transfer node which is depicted in Figure 5.4 above. The developed process is illustrated in Figure 5.7 below, where each step's properties are provided in Appendix I.

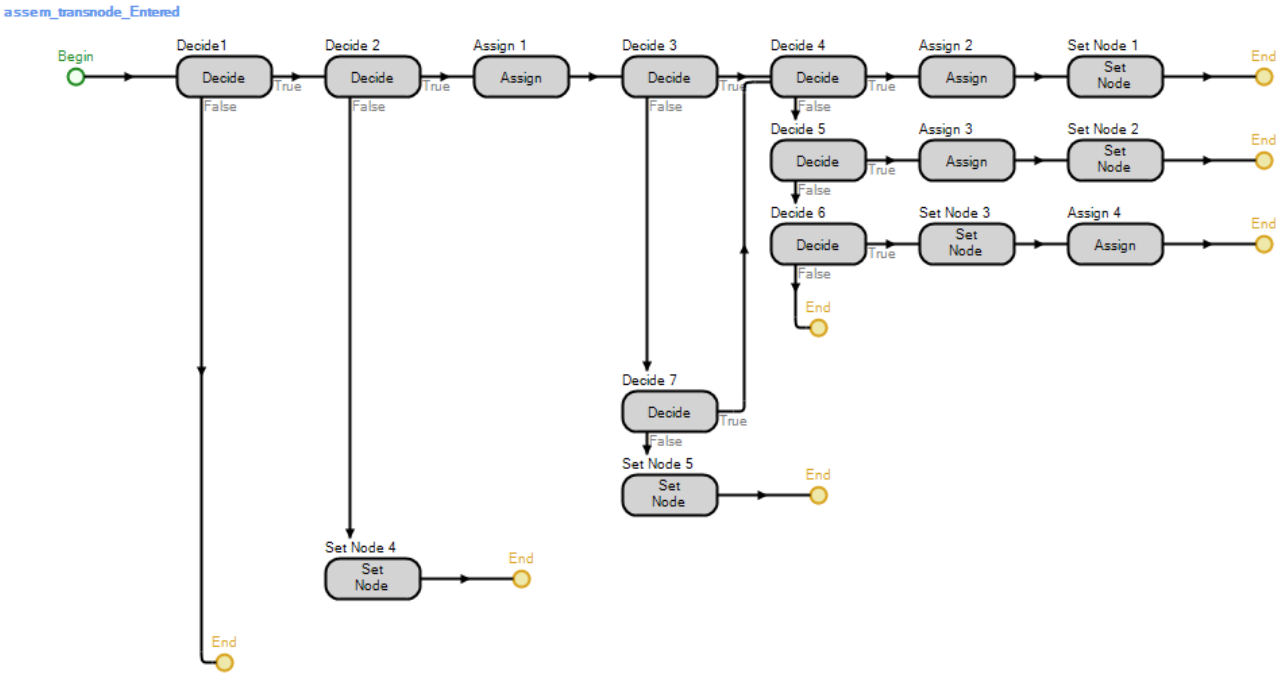


Figure 5.7: Representation of Process at *assem_transnode* - *assem_transnode_Entered* Process

The *assem_transnode_Entered* process is developed as follows:

- First, three boolean state variables are defined namely, *transnode*, *assem*, and *c_weld*. Each of these variables' *Initial State Value* is assigned a value of 'True'.
- Thereafter, an Integer State Variable, namely *v_Count* is identified, which is used to determine the number of canisters to be assembled together.
- The proceedings of the process are as follows:
  - **Decide 1:** This step serves as a decision point. The step ensures that the model uses the logic corresponding to the relevant entity, namely *ModelEntity*. All entities are marked as 'True' and proceed to the next step.
  - **Decide 2:** This decision point determines whether the entity has already gone through the process or not. Specifically, decide if the entity is a canister or a 9-metre pipe based on the Boolean variable called *transnode*. If *transnode* is True, indicating that the entity has not gone through the system and is a canister, the entity goes to the 'Assign 1' step. Otherwise, it proceeds to 'Set Node 4'.
  - **Assign 1:** Here, the *transnode* variable is set to False for the entity under investigation, ensuring that a 9-metre assembly does not go through the following steps again.
  - **Decide 3:** This decision point determines whether the Input Buffer at the parent node of the Combiner is empty. If it is empty, the entity goes to the parent node. Otherwise, the entity goes to 'Decide 7'. The 'Decide 3' step also ensures that the parent node receives an entity before the member node.
  - **Decide 4:** The identified Integer State Variable, *v_Count* is used to decide where the entity should be transferred to acquire a 9-metre assembly. The initial state of *v_Count* is 0,

therefore, the entity proceeds to 'Assign 2'. The 'Assign 2' step increases *v_Count* by 1 to prepare for 'Decide 5'. After 'Assign 2', the entity goes to 'Set Node 1', which sends the entity to the parent node of the Combiner.

- **Decide 5:** At this step, check if *v_Count* is 1, which is True since 'Assign 2' incremented it. The next step is 'Assign 3', which increases *v_Count* by 1 to make it 2. Then, the entity goes to 'Set Node 2', which transfers the entity to the member node of the Combiner.
- **Decide 6:** Similar to 'Decide 5', this step evaluates if *v_Count* is 2 (which is now true due to 'Assign 3'). 'Set Node 3' sends the entity to the member node of the Combiner. Afterward, 'Assign 4' resets *v_Count* to 0, allowing the cycle of steps from 'Decide 4' to 'Decide 6' to repeat for the next 9-metre assembly. This setup ensures that there is one entity at the parent node and two entities at the member node, which are processed at the *assem* workstation to deliver a 9-metre assembly.
- **Set Node 4:** For entities classified as 9-metre pipes (with *transnode* set to False), these entities enter this step. The entity is transferred to the parent node at the Combiner.
- **Decide 7:** This decision point determines whether the entity should contribute to a 9-metre assembly or a 12-metre assembly. If the entity at the parent input buffer has not been through the *assem* workstation (determined by the Boolean variable *assem* set to True), it proceeds to 'Decide 4' for the steps needed to create a 9-metre assembly. Otherwise, if the entity has already undergone the steps for a 9-metre pipe (with *assem* set to False), the entity goes to 'Set Node 5' to create a 12-metre pipe.
- **Set Node 5:** At this step, the entity is transferred to the member node of the Combiner. With one entity at the parent node (as set by 'Set Node 4') and another at the member node, a 12-metre assembly is created by assembling the two entities.

To ensure the effective execution of the 'Decide 3' and 'Decide 7' steps in the *assem_transnode_Entered* process, it is important to distinguish between assembling a 9-metre or a 12-metre pipe at the *assem* workstation. This distinction is achieved through the development of an additional process, namely *assem_Processing*, which is triggered at the *assem* workstation.

The *assem_Processing* process is depicted in Figure 5.8 below, and the properties of each step are detailed in Figure I.6 in Appendix I.

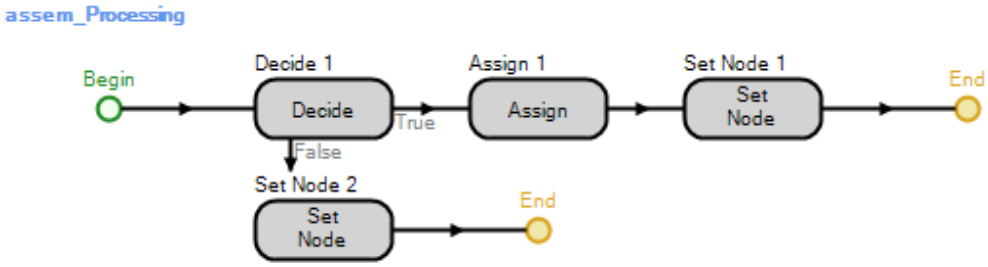


Figure 5.8: Process Defined as *assem_Processing*

The development of the *assem_Processing* process is as follows:

- **Decide 1:** The first step in this process is the 'Decide' step. The step determines whether the *assem* Boolean-variable holds a value of True or False. If the variable is True, indicating that the 9-metre assembly process has not occurred, the subsequent step is 'Assign 1'. If the variable is False, signifying that the 9-metre assembly process has already taken place and the 12-metre assembly process is needed, the step to be followed is 'Set Node 2'.
- **Assign 1:** Here, the *assem* Boolean-variable is assigned a value of False, indicating that the entity under investigation is going or had gone through the necessary process to deliver a 9-metre assembly.

- **Set Node 1:** After assigning the Boolean-variable the value of False, the delivered entity, i.e., the 9-metre assembly, proceeds to the *c_weld* workstation.
- **Set Node 2:** If the necessary steps have already been completed to deliver a 9-metre assembly, indicated by the *assem* Boolean variable having a value of False, the entity is therefore classified as already holding the properties of a 12-metre assembly and proceeds to the *c_weld* workstation.

Furthermore, after the entity is processed at the *c_weld* workstation, it is necessary that the entity is defined as being a 9-metre pipe or a 12-metre pipe. Hence, if the entity is identified as a 9-metre pipe, then the entity must proceed back to the *assem* workstation where another canister is added to form a 12-metre assembly. Otherwise, if the entity is identified as a 12-metre pipe, then the entity proceeds to the *inspect* workstation. The mentioned deliverable is achieved by developing a process, namely *c_weld_Processing*, which is triggered at the *c_weld* workstation. The *c_weld_Processing* process' arguments, related to the steps of the process, are depicted in Figure I.6 which is displayed in Appendix I.

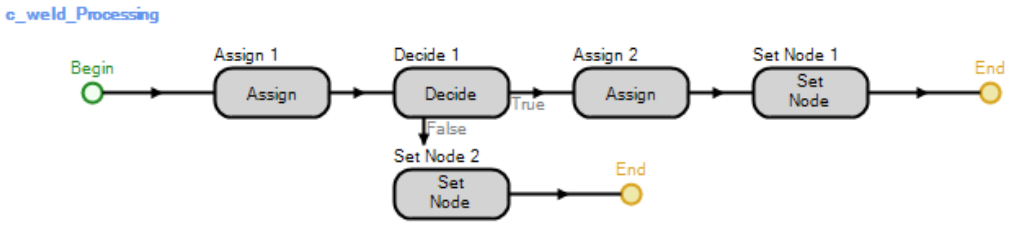


Figure 5.9: Process Defined as *c_weld_Processing*

The development of the *c_weld_Processing* process, is outlined as follows:

- **Assign 1:** In the initial step, an 'Assign' action is added to the process. This step assigns the Boolean-variable *transnode* to a value of False, signifying that after the necessary processes have taken place at the *c_weld* workstation, the entity is classified as a 9-metre pipe. This ensures that after the required operations took place at the *c_weld* workstation to deliver a 9-metre pipe, the entity is transferred to the parent input buffer at the Combiner Object, as identified in the 'Set Node 4' step within the *assem_transnode_Entered* process, as explained above.
- **Decide 1:** Following the 'Assign 1' step, a 'Decide' step is introduced. The purpose of this step is to determine whether a 9-metre pipe or a 12-metre pipe is currently being processed. If the Boolean-variable *c_weld* is assigned a value of True, then the entity has not yet undergone the *c_weld* workstation's process, and the entity must follow the necessary steps to produce a 9-metre pipe. If the entity is assigned a value of False, it is identified as a 12-metre pipe and must proceed to the 'Set Node 2' step.
- **Assign 2:** Another 'Assign' step is added to the process to indicate that the entity has been processed at the *c_weld* workstation. This is achieved by setting the *c_weld* Boolean-variable to False.
- **Set Node 1:** The processed entity, representing a 9-metre pipe, is sent back to the *assem_transnode* transfer node, where the required steps take place to deliver a 12-metre assembly and consequently a 12-metre pipe.
- **Set Node 2:** During this step, the entity is recognised as a 12-metre pipe, and the entity proceeds to the next workstation, namely the *inspect* workstation.

For the *assem* workstation, each of the two operations, involving the assembly of either a 9-metre or a 12-metre pipe, has its own distinct processing time. Similarly, at the *c_weld* workstation, the welding process for a 9-metre pipe and a 12-metre pipe entails different processing times. Consequently, the same method, utilising the *Math.If* function, is employed, as previously described for the *roll* workstation.

5.2.3 Machine Reliability Metrics

Machine reliability metrics are measures used to assess the dependability and performance of equipment, machines, or systems [63]. The workstations within the pipe manufacturing system where machine reliability metrics are applicable, include:

1. *roll*: roll press
2. *l_weld*: ESAB#1
3. *c_weld*: ESAB#2

It is important to note that the *Reliability Logic* is only included for the larger machines and is excluded from workstations that make use of equipment such as angle grinders. One important aspect of *Reliability Logic* is the ability to define failure and repair processes [1]. Failure characteristics of the machines, such as Mean Time Between Failure (MTBF), are modelled in Simio to determine the likelihood of a machine entering the failed state. Likewise, repair characteristics, such as repair times or Mean Time To Repair (MTTR), are defined to represent the duration required to restore a failed machine to its working state.

Similar data analysis took place, as outlined in Chapter 4, to effectively require the MTBF and MTTR values related to each relevant machine. Hence, the relevant data were extracted from SAP, whereafter data cleansing took place. The data were further divided into adequate groups that reflect the specific workstation's reliability logic information. Thereafter, the mean MTBF and MTTR values for each applicable workstation were determined. The acquired and determined MTBF and MTTR values related to the relevant workstations (machines), are listed in Table 5.3 below.

Table 5.3: Reliability Logic of Machines

Machine	MTBF (hours)	MTTR (hours)
Roll Press	602.51	2.99
ESAB#1	1195.44	2.83
ESAB#2	358.82	3.88

Once the necessary machine reliability logic is built into the relevant object libraries, the next step involves establishing connections between the related object libraries. In Simio, this objective is achieved with the use of networks.

5.2.4 Networks within Simio

A network is a collection of linkages that an entity can follow to its required destination [1]. Within the developed model, all the network types are defined as being *Paths*. A path object is a link that defines a pathway between two node locations where the travel time is determined by the path length and traveller's speed [1]. The path object is thus an ideal choice for establishing connections between workstations and facilitating the movement of transporter objects. This is because it allows for a more accurate representation of reality, ensuring that the simulation closely reflects real-world scenarios.

Within the system, after the canisters are completed being processed at the *l_weld* workstation, 60% of the entities must be transferred to the *assem* workstation, while the remaining 40% are rerouted back to the *roll* workstation to adhere to re-rolling, as elaborated on in Subsection 2.2.3. This system configuration is achieved by using a concept called *properties*. In the Simio simulation environment, properties are essentially parameters associated with specific objects [1]. Specifically, a *Selection Weight* property is introduced within the relevant paths, serving as a reference parameter. This reference property is important in determining the appropriate weight allocation for the identified paths.

Regarding the transporter objects within Simio, the Paths on which the vehicles must travel are identified as 'bi-directional' paths. This property identification ensures that the vehicle object can

travel in both directions on the path, which is necessary as vehicles must drop entities at a designated location and then travel back to the specified node to pick entities up again. Details on the vehicles utilised and the necessary properties are defined in Section 5.2.7 below.

Cranes are also utilised to transfer entities to the required destinations. All travel is done through free space without the need to explicitly identify a network. Further details on the Crane object are provided in Section 5.2.7.

The Worker object also utilises Paths to transfer from one location to another. These Paths are configured as 'Bi-directional' to allow a two-way travel on the network. Workers are required to exclusively follow these network paths for their travel activities. For a more detailed understanding of Worker objects, please refer to Section 5.2.6 below.

While it is important to observe and comprehend the flow of materials on the designated networks, the model's build is enhanced by visualising the transformation of raw materials. Consequently, a discussion regarding the symbolic transformation that takes place in the model is discussed in the next subsection.

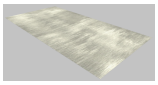
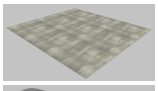
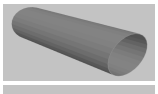
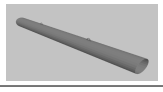
5.2.5 Symbol Transformation

To elaborate and enhance the change of appearance of the model entity during the simulation run, symbol transformation takes place. The various visual representations enhance the comprehension and clarity of the entity when being transferred between the different workstations within the model. The utilisation of symbol transformation can also contribute to the *verification* and *validation* of the constructed model, to be further discussed in Section 5.3.

For symbol transformation, the applicable symbol first needs to be developed, designed or identified. Symbols can be downloaded from Trimble 3D Warehouse, imported from relevant file formats, or developed within Simio [1]. For the purpose of this dissertation, the symbols are either created within Simio, through the use of relevant functionality or developed within AutoCAD. The software AutoCAD is a computer-aided design (CAD) software used for creating 2D and 3D drawings and models [64].

Symbol transformation takes place at workstations where the entity is transformed into another shape or work object. At the beginning of the simulation run, the model entity starts with *Symbol 0*, where a visual illustration of the symbols used is depicted in Table 5.4 below. It is important to note that the symbol list in Simio is zero-based, thus the first symbol is not Symbol 1, but Symbol 0.

Table 5.4: Symbol Transformation

Symbol Number	Illustration	Workstation
0		Start of model
1		s_line
2		roll
3		assem (9-metre)
4		assem (12-metre)

The first symbol transformation, namely *Symbol 1*, takes place at the *s_line* workstation. The *Symbol 1* symbol transformation reflects the instance where one plate is cut according to size to deliver two

smaller plates. This occurrence is achieved by identifying a *State Assignment* which is triggered after the entity is processed, and adheres to the required arguments to achieve the outcome. Hence, the current symbol allocation of the entity, namely *Symbol 0* is re-defined to display *Symbol 1*.

After symbol transformation at the *s_line* workstation, the next symbol transformation takes place at the *roll* workstation, where plates are rolled into canisters. This symbol transformation is from *Symbol 1* to *Symbol 2*. It is important to note that the required arguments are developed to ensure that when re-rolling of canisters takes place at the *roll* workstation, *Symbol 2* are still displayed.

The third and fourth symbol transformations takes place at the *assem* workstation where the canisters are joined to deliver either a 9-metre or a 12-metre assembly. This necessitates two distinct symbol transformations within the *assem* workstation. First, a Boolean-variable, namely *assem_sym* is identified. This variable is used to determine whether the symbol transformation at the *assem* workstation must adhere to comply with *Symbol 3* (9-metre pipe) or *Symbol 4* (12-metre pipe) transformation.

At the *assem* workstation, the *State Assignment* functionality is used to specify what symbol must be declared in the specific instance under consideration. The developed assignment reads as follows:

$$Math.If(ModelEntity.assem_sym == True, 3, 4),$$

where the entity is identified to represent *Symbol 3* when the *assem_sym* Boolean-variable reflects a value of True, else the entity must represent *Symbol 4*.

To determine whether the *assem_sym* Boolean-variable must contains a value of True or False, the process *assem_Exited* is developed, as displayed in Figure 5.10 below, where each step's properties are depicted in Appendix I.

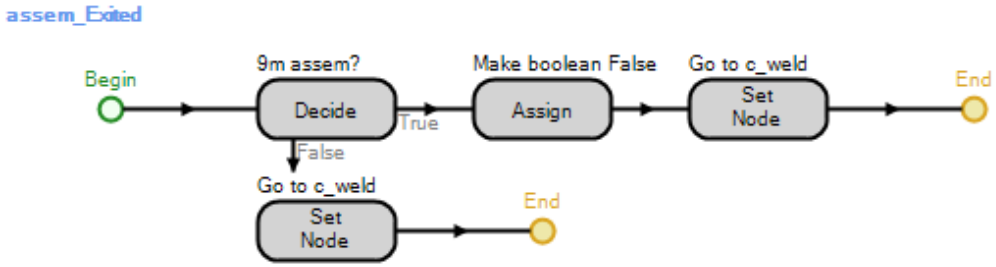


Figure 5.10: Process on Exiting assem workstation

The *assem_Exited* process is built by first identifying a 'Decide' step. I.e. if the *assem_sym* Boolean-variable is True, then the process of first joining entities to deliver a 9-metre assembly must take place. Thereafter the *assem_sym* Boolean-variable, in the 'Assign' step, is allocated a value of False, indicating that the 9-metre process is done. The 9-metre assembly is thereafter sent to the *c_weld* workstation, identified by the 'Set Node' step. If, at the 'Decide' step, the Boolean-variable is identified as False, i.e. indicating that the entity displays properties of being a 9-metre pipe, then the assembly is sent to the *c_weld* workstation where the 12-metre assembly is manufactured in a 12-metre pipe.

Acknowledging the importance of identifying and evaluating symbol transformation within the system is essential. Another important aspect to highlight is the role of the worker object, which significantly impacts the system's functionality and operational dependency. An elaboration regarding the worker object takes place in the next subsection.

5.2.6 Material Handling: Workers

The mobile resources, namely the Workers, serve a dual purpose: these resources can be utilised by the model's process logic to carry out tasks by means of seizing and releasing, as well as facilitate the

movement of entity objects between various node locations.

A brief discussion regarding implementing the Worker object within the developed model is as follows:

- At each workstation, a Basic Node is placed. This serves as a location to where the Secondary Resource, i.e. the Worker object, must travel when working at the specific workstation.
- Each workstation's Basic Node is connected, via Paths, to the identified *Lunchroom*'s Basic Node. It is beneficial to note that for each worker there is a unique *LunchroomN* Basic Node. Figure 5.11 below depicts the location, as highlighted with the purple oval symbol, where the *Lunchroom* is situated. Also, for visualisation purposes, the Paths followed by the Worker objects are identified with the colour green.

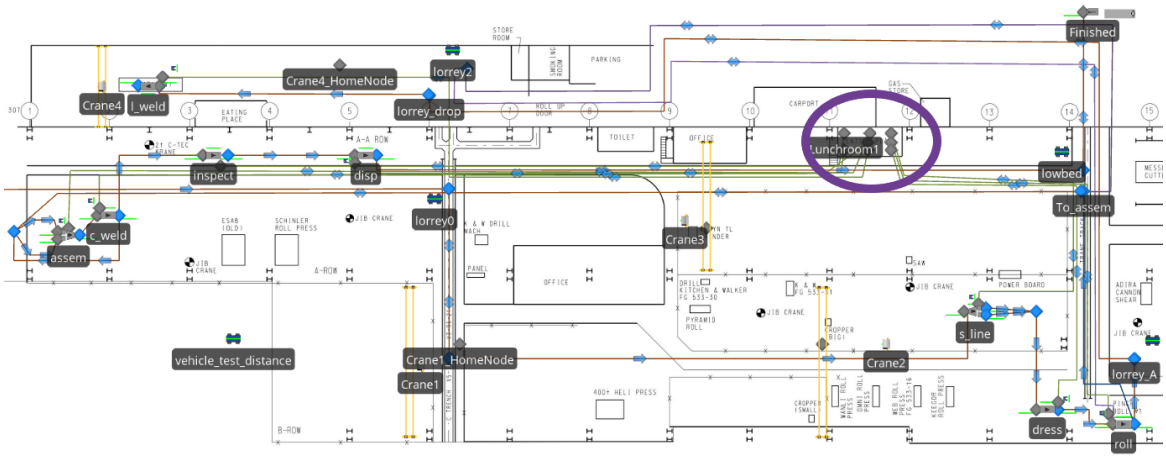


Figure 5.11: Lunchroom Position

- The Path *Type* property of each path is set to 'Bi-directional'. This identification states that the travelling mode is not subjected to follow only one direction, but the paths can be followed in both directions. This is necessary to ensure that the workers can travel to and from the *Lunchroom*.
- To ensure consistency throughout the simulation model, all workers follow a speed of 1 metre per second when travelling from and to various locations within the model.
- Each worker's 'Home' destination is set to the Basic Node at each workstation. The workers are to remain in the current position when *Idling* takes place, and when the worker is *Off-shift*, the worker is to go to the 'Home' destination.
- All worker objects are set to follow the First In First Out *Ranking Rule*. I.e., the first entity entering the workstation is to be processed and/or analysed before proceeding to the next entity.
- The efficiency factor for all the workers is kept the same, where the efficiency is identified as being 1 (100%).

Within the Simio model, a Resource Object, for example, a Server, and the secondary resource, classified as a Worker, need to stop processing an entity when the Worker object goes off shift. When the Worker comes back on shift, the Resource Object and Worker continue processing the interrupted entity of the remaining processing time. The schedule followed by the workers was outlined in Subsection 2.1.3, and the development of the work schedule within Simio is elaborated on in the next subsection.

Workers' Work Schedule

To effectively reflect the operational hours of the Worker resource object, it is important that a work schedule is allocated to the employees. As evident from the work schedule that was outlined in

Subsection 2.1.3, it is observed that there are mainly two Day Patterns that are present during the simulation run, namely *DayPattern1* and *DayPattern2*.

The developed *DayPattern1* is utilised from Monday till Thursday, whereas *DayPattern2* is allocated to Fridays. Furthermore, no operational tasks take place on Saturdays and Sundays. After the Day Patterns were identified within Simio, the necessary Work Schedule was developed, namely *Schedule1*. This Work Schedule is identified as having a 7-day week, from Monday to Sunday. A demonstration of the required schedule is provided in Figure 5.12 below.

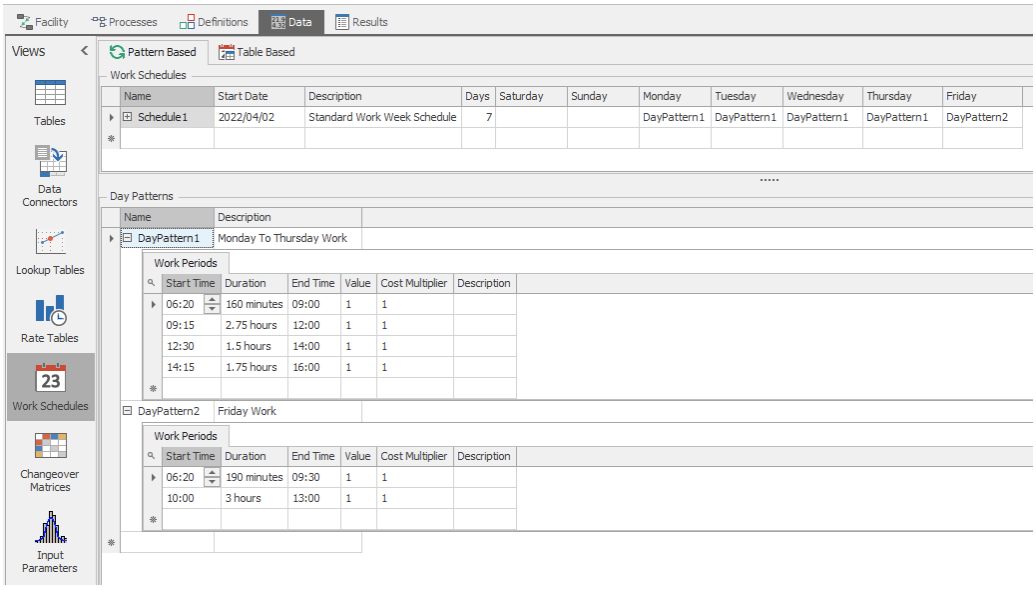


Figure 5.12: Worker's Work Schedule

The Worker object is not the only mobile resource used to effectively reflect to proceedings and operations of the pipe manufacturing system. Other mobile resources include resources that are utilised to transfer entities.

5.2.7 Material Handling: Transferring of Entities

The two main mobile resources used to transfer entities to the desired locations within the simulation model are the Vehicle and the Crane object. Firstly, the usage and implementation of the Vehicle object are elaborated on, where-after the development of the Crane object in Simio is discussed.

Vehicle Object

In the modelled system, a dynamic population of moveable unit resources can be defined by using a Vehicle object. The Vehicle object has the ability to pick up entity objects at one location, transfer the identified objects over a network of linkages or free space, and then drop the entities off at a destination location [1].

There are four vehicle objects that are present within the Simio model: *lorrey0*, *lorrey1*, *lorrey2*, and *lowbed*. The movement and location where the specific Vehicle object is used are depicted in Figure 5.13 below.

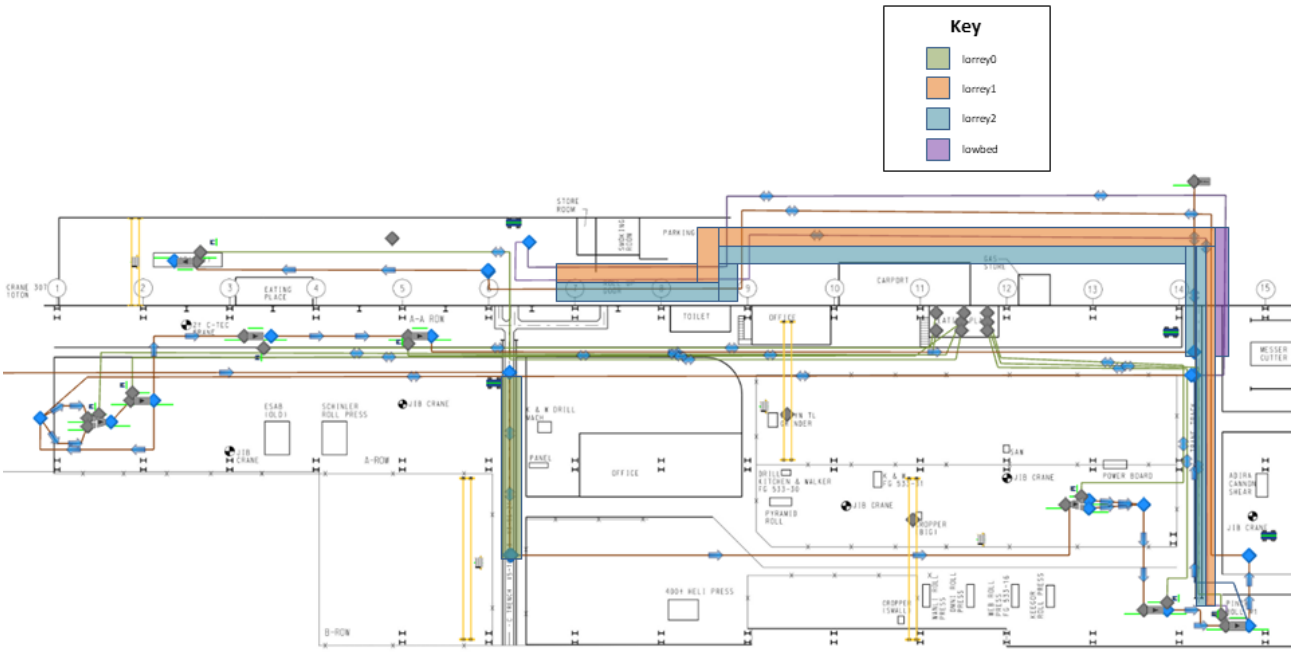


Figure 5.13: Movement of Vehicle Object

Hence, the movement of the Vehicle objects takes place as follows:

- *lorrey0*: Transfers plates from Bay 2 to Bay 1.
- *lorrey1*: Transfers the canisters from Bay 1 to Bay 3.
- *lorrey2*: Transfers entities to one of two locations. As discussed in Section 2.2.3, it is identified that the canisters are either transferred to Bay 1 or Bay 2.
- *lowbed*: Transfers finished 12-metre pipes to customers.

The vehicle objects, as modelled within Simio, consist of the following main properties:

- Each vehicle resource has an identified capacity. I.e., the vehicle must dwell at the relevant destination node until the identified ride capacity of the vehicle is reached. After the vehicle contains the identified number of entities, the vehicle can proceed to the desired location. When the vehicle resource arrives at the drop-off node, all of the entities on the Vehicle object are dropped off before the vehicle resource transfers back to the pick-up node. This capacity is identified as the *Initial Ride Capacity*. Each vehicle's *Initial Ride Capacity* is listed in Table 5.5 below.

Table 5.5: Vehicles' Ride Capacity

	lorrey0	**lorrey1**	**lorrey2**	**lowbed**
Vehicles' Ride Capacity	5	4	2	2

- The First-In-First-Out strategy is used during entity pickups and drop-offs.
- The load and unload time for each vehicle resource is listed in Table 5.6 below.
- Each vehicle resource has a specific speed, as listed in Table 5.6, identified as the *Initial Desired Speed*.
- Each vehicle is associated with a 'Home' node, signifying the initial location of the vehicle at the beginning of the simulation. The *Idle Action* and the location of the vehicle during the *Off Shift*

occurrence, are also identified to take place at the designated 'Home' node. For demonstration purposes, the 'Home' node of each Vehicle object can be observed in Figure 5.14 below.

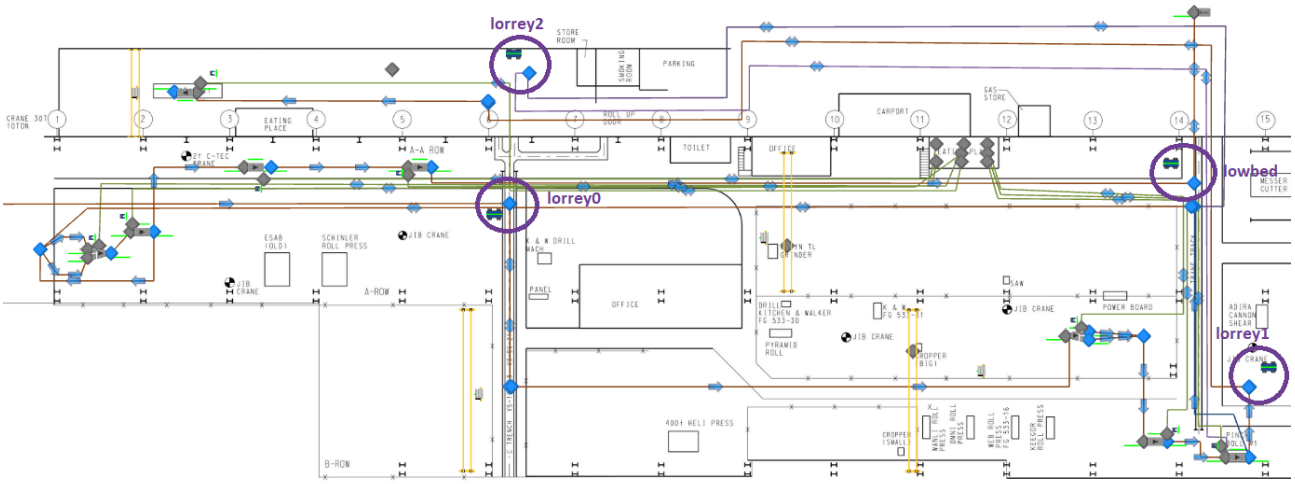


Figure 5.14: Identification of 'Home' nodes for Vehicle Objects

- The Vehicle objects are required to perform travel movement by only using the assigned network. The paths on which the vehicles travel are set to 'Bi-directional' paths. This ensures that the Vehicle object can travel from and to the various destination nodes.

To summarise, the relevant properties regarding the Vehicle objects are listed in Table 5.6 below.

Table 5.6: Vehicle Objects Specifications

	lorrey0	**lorrey1**	**lorrey2**	**lowbed**
Initial Ride Capacity	5	4	2	2
Load Time (m/s)	5	7	7	10
Unload Time (m/s)	5	7	7	10
Initial Desired Speed (m/s)	0.5	2	2	0.5

Crane Object

Within Simio, a Crane object is a dynamic transporter that can pick up, move, and drop entities at the specified nodes [1]. It is important to note that due to overhead cranes being the only type of crane utilised during the pipe manufacturing operation, the term "crane" refers to an overhead crane.

As observed in Section 2.2.3, the presence of three bays for crane usage are present within the MechFab job shop. *Bays* are an important factor to consider when developing cranes within Simio as *Bays* represent the area in which the relevant cranes can travel in [1]. A *Bay*, within Simio, is a fixed object which is identified by a rectangular region over which one or more cranes can travel [1]. Within each Bay, a Basic Node is allocated which depicts the 'Home' node for the Crane object. The cranes are identified to move to the designated 'Home' node during periods of *Idle Action* or at the beginning of the simulation run. The relevant bays were discussed in Section 2.2.4.

The first bay, namely *Bay 1* consists of 2 cranes, and *Bay 2* and *Bay 3* both consist of 1 crane. Therefore, the model consists of a total of four overhead cranes. The functionality of each crane is as follows:

- *Crane1:* Moves plates from the location within *Bay 1* where *vehicle 0* initially placed the plates, to the input node at the *s_line* workstation. A visual demonstration is provided in Figure 5.15 below, where the blue line illustrates the path being followed by *Crane 1*.

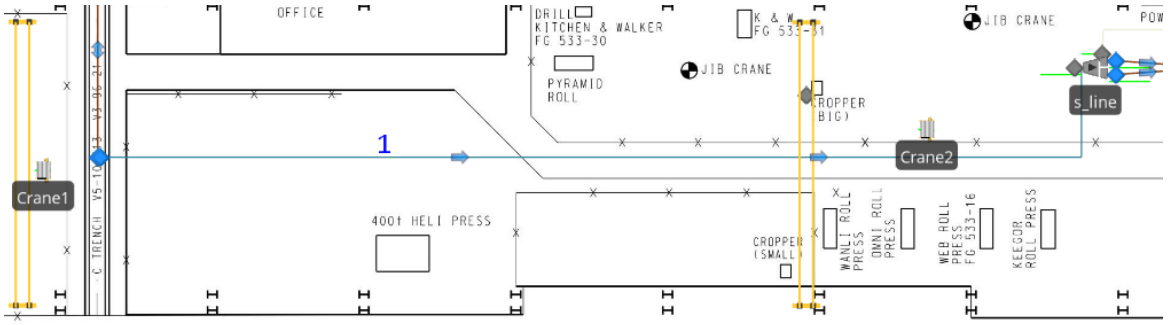


Figure 5.15: Movement of Crane 1

- *Crane2:* This crane is used to transfer entities to and from three network routes. Similar to Figure 5.15 above, the network routes for *Crane 2* are identified by the blue paths in Figure 5.16 below. The applicable networks are as follows:

  1. Crane transfers entities (plates that are cut in half) from the *s_line* workstation to the input buffer at the *dress* workstation.
  2. The entities (plates) are transferred from the output buffer of the *dress* workstation to the input buffer at the *roll* workstation.
  3. Entities (rolled canisters) are transferred from the output buffer of the *roll* workstation to the designated area within *Bay 1* where the canisters are transferred via *lorrey1* to the *l_weld* workstation.

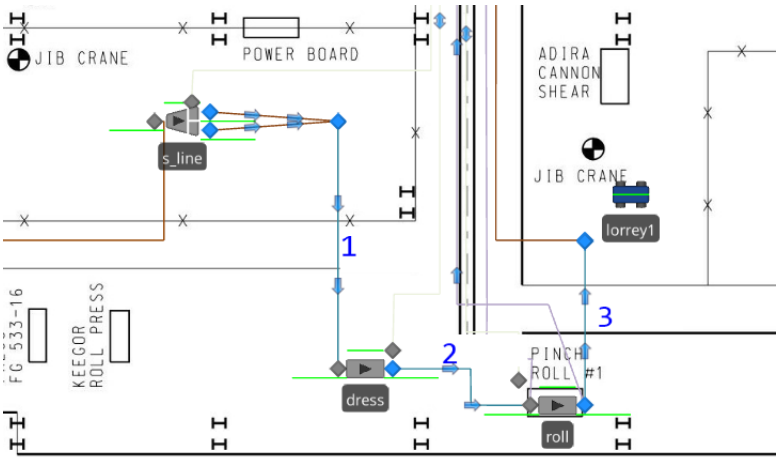


Figure 5.16: Movement of Crane 2

- *Crane3:* A total of 5 network paths are dependent on *Crane 3*, as displayed in Figure 5.17 below. These paths include:

  1. Canisters are transferred from the destination in *Bay 2* to the *assem* workstation. These canisters either come from the *roll* workstation or the *l_weld* workstation.
  2. Either 9-metre or 12-metre pipes are transferred from the *assem* workstation's output buffer to the *c_weld* workstation's input buffer.
  3. At the output buffer of the *c_weld* workstation, if the entity is identified as a 9-metre pipe, then the crane transfers the entity back to the *assem* workstation where another canister is to be joined. If the entity is identified as a 12-metre pipe, then the entity is transferred to the input buffer of the *inspect* workstation.
  4. The inspected 12-metre pipe is transferred from the output buffer at the *inspect* workstation to the input buffer at the *disp* workstation.

5. Finally, the 12-metre pipe is transferred from the output buffer at the *disp* workstation to the area where the pipes are stored and ready to be transferred to the customers. From here, the *lowbed* vehicle object transfers the 12-metre pipes to the customers, i.e. the Sink, namely *Finished*.

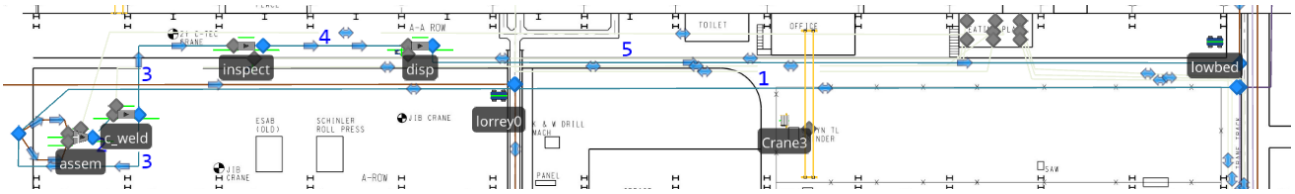


Figure 5.17: Movement of Crane 3

- *Crane 4:* There are two networks dependent on *Crane 4*, as depicted in Figure 5.18 below. These networks include:

  1. Transferring canisters from the designated area in *Bay 3* to the input buffer at the *l_weld* workstation.
  2. Transferring canisters from the output buffer of the *l_weld* workstation to the area from where *lorrey2* will transfer the entities to the relevant destinations.

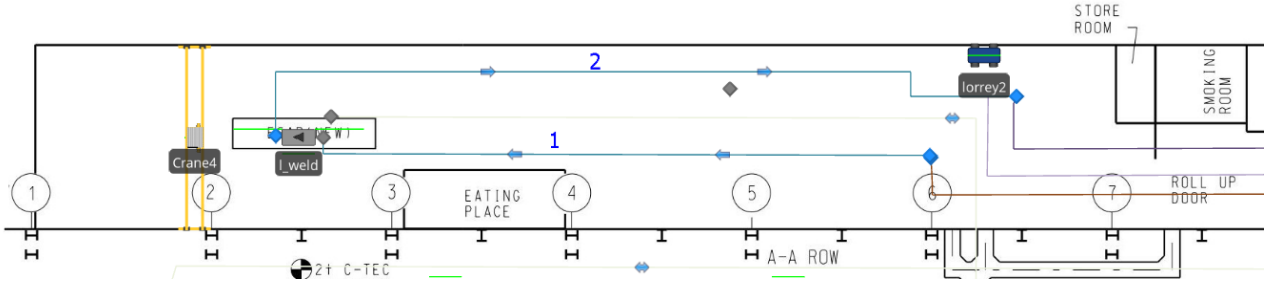


Figure 5.18: Movement of Crane 4

It is important to note that the travel time, handling time of the cranes, and other crane-related time factors are included in the workstations' processing time. Therefore, to ensure that data or actions within the model are not duplicated or unnecessarily added to the system, the *Initial Desired Lateral Speed* and *Lift Speed* are set to a relatively high value. Also, the *Load Time*, *Unload Time*, *Lateral Acceleration* and *Lateral Deceleration* are set to a value of 0.

5.2.8 Simulation Run-Time

The purpose of simulation duration (run-time) is to simulate and observe how a system or process behaves over time, enabling the analysis of dynamic interactions, performance metrics, and potential outcomes in a comprehensible manner [1]. By specifying the starting and ending times of the model, users can develop the simulation to suit required simulation-run needs and in turn gain valuable insights [1]. The following simulation run-time information is built into the model:

- The simulation's starting date is 2022/04/04 06:20:00. Although the date chosen is arbitrary, it is important to note that the simulation run starts on a Monday. The time specified relates to the developed work schedule, as outlined in Subsection 2.1.3.

- The simulation run concludes after 4.5 weeks. The reason for specifying the run time as 4.5 weeks is to be able to compare the developed model's results with the expected outcome. More clarity regarding validating the developed model is provided in Section 5.3.

Now that the fundamental characteristics essential for effectively establishing the current state of the pipe manufacturing system have been discussed, it is advantageous to provide an overview of the system that is modelled in Simio. Hence, the next subsection introduces the development of an OFD.

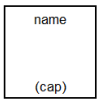
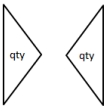
5.2.9 Object Flow Diagram

The OFD can be defined as a schematic sketch that outlines the basic structure of a simulation [65]. The diagram is used to recognise which flow objects and entities need to be considered, the level of detail the model needs to be presented and to emphasise the data necessary for the model development.

The OFD for the current state scenario is displayed in Figure 5.19 below, where the notations used are defined in Table 5.7 below. For clarity purposes, the workstations' queue capacities are provided in brackets at each 'fixed resource', as explained in Table 5.7 below. The breakdown of each relevant workstation is identified as 'downtime' in Table 5.7 below, where the time to downtime and the duration of the breakdown are identified.

This diagram aims to synthesise the information provided in the preceding subsections about the relevant elements required for the development of the simulation model. The diagram serves as a summary, offering an overview that provides a clear understanding of the developed model, ensuring that all relevant aspects have been accounted for. It is important to note that the main goal of the developed OFD is to simplify and understand the simulation logic - and not to represent the complex logic and model design.

Table 5.7: OFD Notations

Symbol	Description
	Source
	Sink
	Flow/ Route
	Communication
	Fixed Resource
	Downtime
	Process
	Mobile Resource
	Group/ Ungroup
	Performance Measure

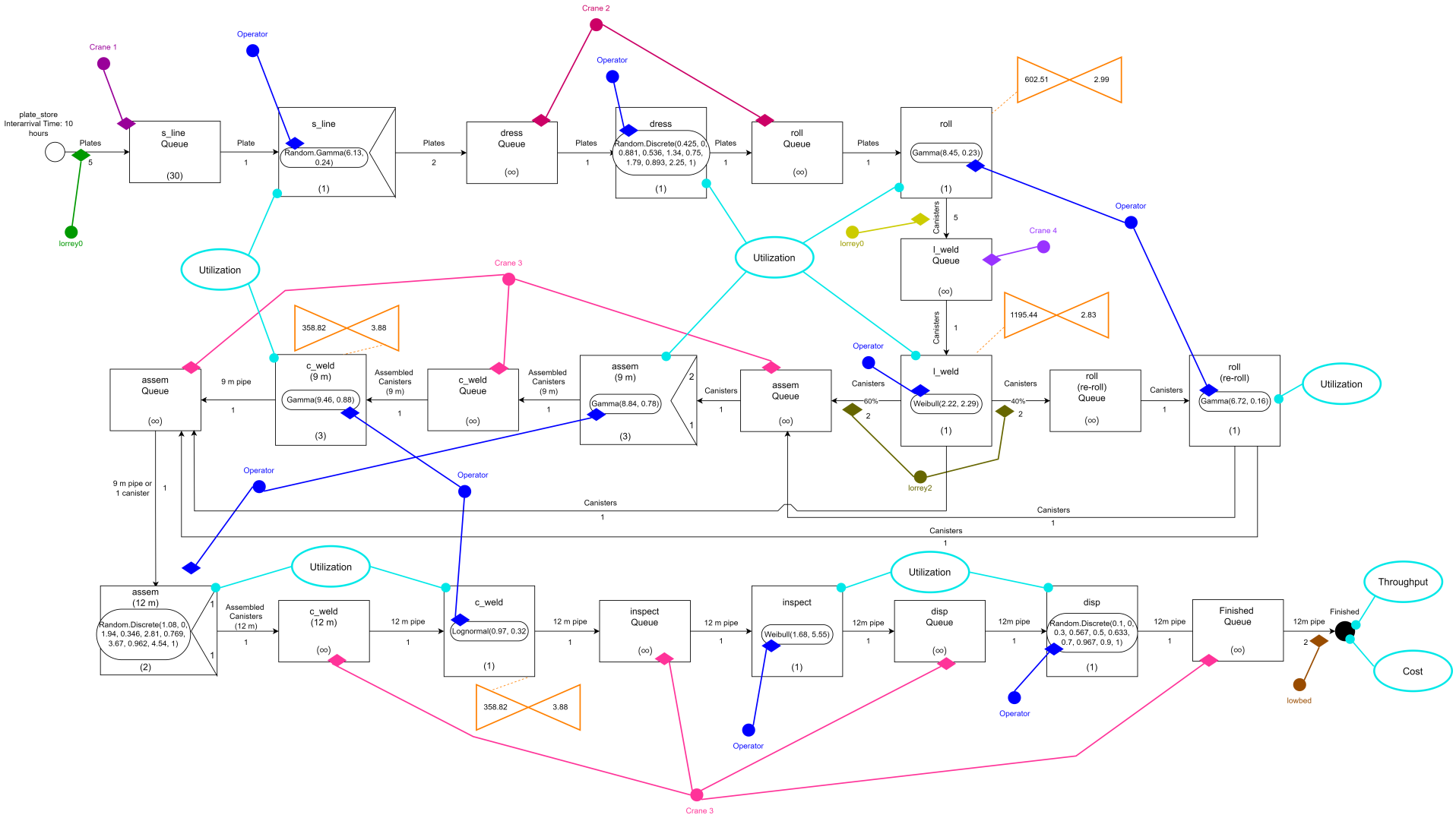


Figure 5.19: OFD of Current State

After building the current state model in Simio, model validation and verification are required. This process is important to confirm the model's correctness in representing the real-world system and to ensure confidence in the accuracy of the model's predictions and results.

5.3 Model Verification and Validation

Before utilising the developed Model-0.0 to analyse the system to identify improvement opportunities, some confidence must be determined to identify the 'correctness' of the model. This analysis is done by *Verifying* and *Validating* the developed model.

5.3.1 Verification

Verification involves the process of confirming that the developed model is accurately implemented according to the intended design and specifications [66]. The verification of the model took place during the development of the simulation model. By verifying the model, it is ensured that the model does not have errors and behaves as intended. The verification step is an 'internal' process that does not involve comparing simulation results with real-world data [66].

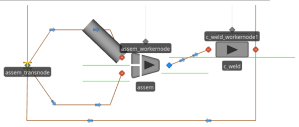
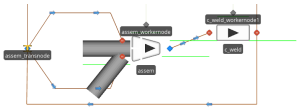
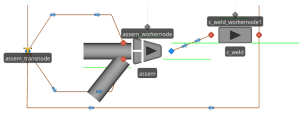
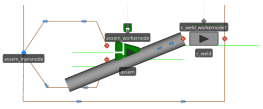
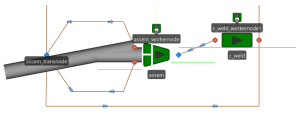
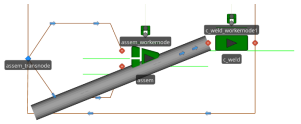
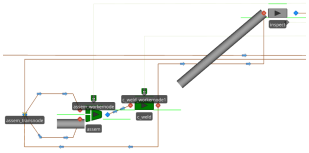
Several important steps and techniques were used during the development of Model-0.0, to evidently ensure a consistent process of *verification* throughout the construction of the model. Some of the steps used to continuously verify the model included [1]:

1. **Logic Review:** A thorough review of the simulation model's logic was conducted throughout the development of the model. This step was done to ensure that the model accurately represents the pipe manufacturing system.

2. **Model Animation:** Simio's animation feature was used to visually trace the operations of the model to ensure that the model align with expectations.

   An example where it was proved beneficial to utilise animation to verify the developed model was during the verification of the *assem* and *c_weld* workstations. The proceedings and operations of the workstations are highlighted in Section 2.2.3, and an explanation regarding the symbol transformation of the various forms of materials are elaborated on in Section 5.2.5. The verification process, which involves the use of animation to assess the *assem* and *c_weld* workstations, is briefly explained in Table 5.8 below.

Table 5.8: Utilising Animation to Verify Model

Step Number	Description	Visualisation in Simio
1	One canister enters Parent node of *assem* workstation.	
2	One canister enters Member node of *assem* workstation.	
3	Second canister enters Member node of *assem* workstation.	
4	9-metre assembled pipe enters Input Buffer of *c_weld* workstation.	
5	9-metre pipe enters Parent node of *assem* workstation (Canister required to deliver 12-metre pipe is already present at Member node of *assem* workstation).	
6	12-metre assembled pipe is transferred to Input Buffer at *c_weld* workstation.	
7	12-metre pipe is transferred to *inspect* workstation.	

3. **Traceability:** Documentation was maintained that traces the model's components to its design specifications, ensuring that all intended functionality has been implemented.
4. **Unit Testing:** Individual components of the simulation model were isolated and tested to ensure the components function as intended before integrating the modules in the larger model.

By diligently applying the aforementioned strategies, the verification process enhances the credibility and reliability of Model-0.0. The correctness of the model is further evaluated by validating the model.

5.3.2 Validation

While verification ensures the continuous evaluation of the model according to its design specifications, validation takes the process a step further. The validation phase assesses the simulation model's results with established real-world observations to determine the model's accuracy. Unlike verification, which is an 'internal' process, validation involves an 'external' evaluation where the model's performance is benchmarked against actual data [66]. The gap is thus bridged between the simulated model and real-world dynamics.

The main methodology utilised to validate the model included *empirical data comparison*. With

this methodology, the actual data collected is used to compare the simulation output. By evaluating how closely the model's predictions align with real-world measurements, insights are gained regarding the model's effectiveness in representing the actual pipe manufacturing system.

Model validation was conducted for the production of a single 12-metre pipe. This was done to ensure a comprehensive and efficient monitoring approach for tracking the activities and progress of entities associated with the system.

The validation phase took place by utilising three distinct factors: workstations' processing times, travel times between workstations, and workers' considerations. The three listed factors are used to effectively validate the model by comparing the factors with what is expected from the model. An elaboration regarding the factors is as follows:

1. **Workstations' processing times:** The validation phase incorporates the mean operation times for each workstation, derived from the data 'booked' by the employees. A discussion regarding the 'booked' times was done in Subsection 4.1.1. The determined mean times can be observed in Table 5.9 below.

Table 5.9: Mean Processing Times of Workstations

Workstation	Mean Hours	Mean Minutes
s_line	1.47	90
dress	1.08	65
roll	1.93	115
l_weld	2.02	120
re-roll	1.06	65
assem (9-metre)	6.92	415
c_weld (9-metre)	8.35	500
assem (12-metre)	2.31	140
c_weld (12-metre)	2.78	165
inspect	4.92	295
disp	0.31	20

   From Table 5.9 above, it can be observed that for each workstation's operation, the mean time in hours and minutes is provided. Firstly, the provided times in hours aim to give an overview of the processes. The rationale behind determining the mean times in minutes is to establish a benchmark for evaluating the performance of the developed model.

   Instead of employing distribution data during the validation phase, the focus is on the use of the calculated mean values to comprehensively assess the model's performance and duration of each workstation's operation. It is important to note that the mean values are merely used for validating the developed model.

2. **Travel times between workstations:** Consideration of material movement between workstations holds importance in the assessment of the developed model. It is important to note that, as mentioned before, the crane transportation times are neglected, due to these specific travelling times being already incorporated into the processing times of the workstations. Consequently, the primary focus rests on the travel times of the vehicle objects.

   To effectively validate the vehicles' movement between workstations it is required that the expected travelling times are compared with the vehicles' travelling times within Simio. The vehicle's expected travelling-time is calculated by dividing the distance travelled, within Simio, by the speed of the vehicle. The distance travelled by vehicles is elaborated on in Section 6.1.3 and the speed that the vehicles travel has been discussed in Section 5.2.7. Furthermore, it is

beneficial to note that the loading and unloading times of the vehicles must also be taken into account, where these relevant times have also been discussed in Section 5.2.7. The expected travel time between the relevant workstations is calculated and provided in Table 5.10 below.

Table 5.10: Expected Travelling Times of Vehicles Between Workstations

Vehicle	From	To	Travelling Time (minutes)	Notes
lorrrey0	plate_store	s_line	30	Vehicle must first contain 5 entities before proceeding.
lorrey1	roll	l_weld	20	The vehicle must first contain 4 entities (canisters) before travelling to the l_weld workstation.
lorrey2	l_weld	assem	15	Vehicle must contain 2 entities (canisters) before travelling to assem workstation.
lowbed	disp	Finished	20	For the purpose of validation, only one pipe is required before transportation commences to Finished.

In Table 5.10 above, an overview of vehicle movements between the identified 'From' and 'To' workstations, along with their corresponding travel times, is presented. Additionally, notable observations are made regarding specific instances where vehicles must be loaded with materials before proceeding to the next workstation. These waiting periods are important to note as these periods have an effect on the overall manufacturing system.

3. **Workers' considerations:** Mainly two factors were taken into consideration when comparing the simulation results with what is expected. First, the work schedule of the workers, as listed and elaborated on in Section 2.1.3. Secondly, the workers' travelling speed from the applicable workstation to the lunchroom and back to the workstation. As previously stated, the workers' speed is identified as one metre per second.

The three listed factors are used to effectively validate the model by comparing the factors with what is expected from the model. This objective is achieved by developing a chart, similar to a Gantt Chart [67], where the chart reflects the expected outcome results of the three aforementioned factors. The total developed chart can be observed in Figure J.1 in Appendix J. It is important to note that this analysis only focused on the manufacturing of one 12-metre pipe.

To effectively determine whether the simulation results adequately reflect what is expected from the model, Simio's *breakpoints* and *tracing* functionalities were used. A *breakpoint* can be added to a node, link or object within the Facility window, or within a step within the Process window [1]. The break-point causes the execution of the model to pause when a transporter or an entity arrives at the specified node, link or object [1]. Furthermore, *tracing* the model entails capturing and recording the sequences of events and activities that occur during the simulation, allowing for in-depth analysis and performance evaluation [1]. Therefore, by using *breakpoints* and *tracing*, the various steps and sequences can be evaluated and compared with the developed Gantt Chart. By comparing the simulation's proceedings with the expected outcomes as stipulated in the Gantt Chart, it can be declared that the model effectively reflects what is expected and has a high accuracy.

For demonstration purposes on how the validation took place, an explanation is provided on validating the operational time for the first model entity at the *dress* workstation. Within Figure 5.20 below, the red dots on the nodes of the *dress* workstation's Server object represent the breakpoints. The results for *tracing* the model are provided in Figure 5.20 underneath. The execution of the model is therefore paused when the entity enters the Input node at the *dress* workstation. From the *trace*

result it can be seen that the entity entered the workstation at time 08:20:00 on Monday, 4 April. This date and time can be observed in the bottom left corner of Figure 5.20.

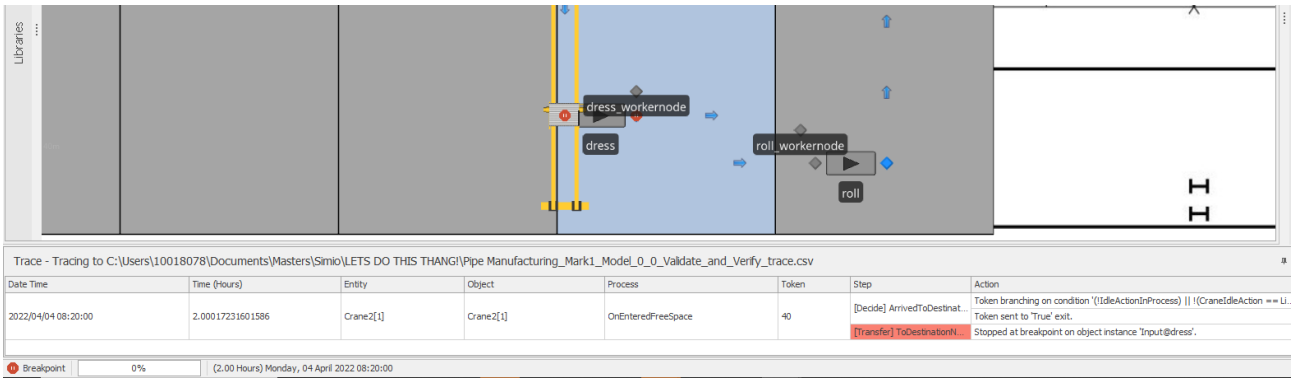


Figure 5.20: Validating *dress* Workstation's Processing Time at Input Node

By unpausing the execution of the model, it is observed that the entity is finished being processed and leaves the *dress* workstation at time 09:41:54, as shown in Figure 5.21 below.

Figure 5.21: Validating *dress* Workstation's Processing Time at Output Node

The expected processing time of the *dress* workstation, as listed in Table 5.9 above, is 65 minutes, which is equal to one hour and five minutes. However, the result obtained within Simio results in that the *dress* workstation took a total of 1 hour, 21 minutes and 54 seconds to process. This observation was expected due to the lunch break that takes place during the specified operation's proceedings. The lunch break is identified to be 15 minutes, as previously stated in Section 2.1.3. The additional 1 minute and 54 seconds can be allocated to the time it took the worker to move to the lunchroom and back to the workstation. The evaluation done within the Gantt Chart, for the aforementioned occurrence, can be observed in Figure 5.22 below. It is important to note that certain time intervals have been intentionally concealed in the constructed Gantt chart to enhance the chart's overall readability.

	Time	6:25 am	6:50 am	6:55 am	8:15 am	8:20 am	8:25 am	9:00 am	9:05 am	9:15 am	9:20 am	9:40 am	9:45 am
	Minutes	5	30	35	115	120	125	160	165	175	180	200	205
	Part of Hour	0.0833	0.5000	0.5833	1.9167	2.0000	2.0833	2.6667	2.7500	2.9167	3.0000	3.3333	3.4167
Workstation/ Activity	Travel												
	s_line			1			2				2		
	dress						1A				1A		
	roll												
	l_weld												
	assem (9-metre)												
	c_weld (9-metre)												
	assem (12-metre)												
	c_weld (12-metre)												
	inspect												
	disp												

Figure 5.22: Section of Gantt Chart

As a result, it can be concluded that the *dress* operation has been successfully modelled in accordance with the stipulated factors, as observed within the developed Gantt Chart. The provided analysis serves as an example, where a comprehensive validation of all the operations within the developed model was done.

The complex nature of simulating the interactions between the *assem* and *c_weld* workstations, as discussed in Section 2.2.3, necessitated the use of an additional approach to thoroughly validate the operations that take place at the two workstations. Further validation took place by splitting the workstations' operations into additional workstations and consequently evaluating the proceedings, as illustrated in Figure 5.23 below.

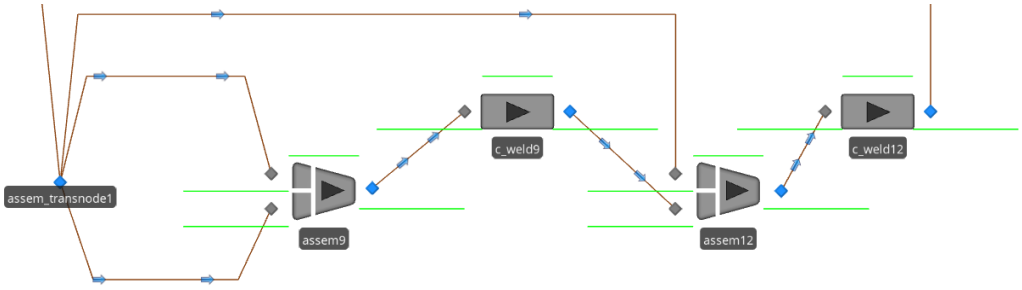


Figure 5.23: Representation of Alternative *assem* and *c_weld* Workstations

It can be observed from Figure 5.23 above that the two operations at the original *assem* workstation is divided into the *assem9* and *assem12* workstations. The operations at the *assem9* workstation adhere to the required specifications to assemble a 9-metre pipe and the *assem12* workstation adhere to the specifications to deliver a 12-metre assembled pipe. Also, the operations that were originally present at the *c_weld* workstation are divided between the *c_weld9* and *c_weld12* workstations. The *c_weld9* workstation adheres to the properties to deliver a 9-metre pipe and the *c_weld12* workstation reflects the operations to deliver a 12-metre pipe. The reason for evaluating the workstations in this manner is due to the interconnectedness of the original layout, as displayed in Figure 5.24 below.

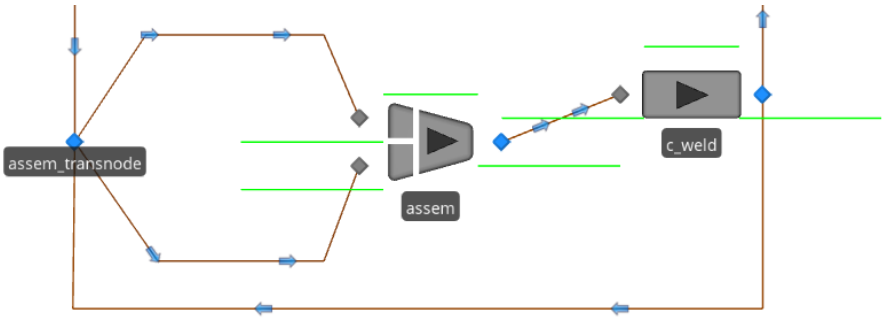


Figure 5.24: Original Setup of *assem* and *c_wled* Workstations

The approach of splitting the workstations' operations was done due to the processing times of other entities at the original *assem* and *c_weld* workstation, influencing the validation of the single 12-metre assembly or pipe. This argument stems from the fact that when the first 9-metre pipe is finished being manufactured at the *c_weld* workstation and transferred back to the *assem* workstation, then the 9-metre pipe must first wait for the *assem* workstation's current operations to complete before processing can start on the 9-metre pipe under investigation. This additional time will influence evaluating and effectively validating the model with the empirical data (Gantt Chart). Hence, the workstations' operations are divided and validated accordingly.

A thorough analysis was done to evaluate the developed Gantt chart with the entire simulation model. As the results from both the developed Gantt chart and the simulation model exhibit close

similarity, a conclusive validation of the model's effectiveness is attained, affirming its accurate development and alignment with the intended specifications. Hence, the techniques applied in the validation stage ensure a credible model and ensure model credibility.

5.4 Concluding Remarks: Model Development

In this chapter, the central focus was the development of Model-0.0, a simulation model developed to reflect the current operations of the pipe manufacturing system. The chapter started with an exploration of relevant model development information, applicable to Simio, establishing a foundation of the subsequent discussions. Thereafter, a discussion regarding the development of the workstations to effectively reflect the required objectives of the stations took place. To ensure an accurate representation of the system, the additional factors and components, related to the workstations were also elaborated on.

Another important aspect that was considered is the reliability of the machinery. The focus was on the MTBF and MTTR metrics, recognising the important impact of these metrics on production time. Further characteristics such as the use of networks to connect workstations and symbol transformation were emphasised. Thereafter, a discussion regarding material handling within the system took place. First, the worker object, and its necessity in the system were highlighted, whereafter the transferring of entities between workstations facilitated by cranes and vehicles was discussed. Lastly, the important concept of simulation run-time was elaborated on. To effectively integrate the identified concepts and elements, an OFD was developed. This diagram offered a comprehensive summary of the simulation model's development and required characteristics.

The chapter then transitioned to elaborate on the important process of verifying and validating the simulation model, a necessary step aimed at ensuring the accuracy and reliability of the simulation model. The verification phase, an internal process, involved confirming that the model's logic aligns with its intended design. Validation was conducted by comparing the simulation outcomes with empirical data from the actual system. Through comprehensive verification and validation, the model's correctness was affirmed, demonstrating its alignment with the required specifications. The current state model establishes a foundation for further analyses to identify improvement opportunities within the pipe manufacturing system.

Chapter 6

Improvement Opportunities Analysis

Within the previous chapter, a discussion took place regarding the development of the simulation model, namely Model-0.0, that effectively reflects the current pipe manufacturing operation.

By assessing the current process flow, as detailed in Chapter 2 in Section 2.2.4, it becomes evident that the process flow is disorganised, with ineffective routes being followed. It was further stated that the identification of an improved process flow is advised. The evaluation and implementation of the improved process flow took place simultaneously and in conjunction with the study's objectives. Due to the improved process flow being a parallel project that took place during this study's investigation, only a brief elaboration is provided regarding the analysis and implementation of the improved process flow.

Given the availability of resources and data, it was deemed most appropriate to conduct model validation, as discussed in Section 5.3, on the existing process flow of the pipe manufacturing operation, rather than on the improved process flow. Nevertheless, since the improved process flow is a parallel project and has recently been implemented within the job shop, it is advisable and preferable to focus the analysis on identifying improvement initiatives related to the state of the improved process flow.

Based on the system developed in Simio, which reflects the improved process flow, bottlenecks are identified iteratively. These bottlenecks are identified with the use of TOC principles and by evaluating the utilisation result of the models, which also serves as one of the utilised performance measures. It is important to note that it is ideal to attain at least a 90% utilisation rate for the workstations to be classified as effective. A workstation utilisation threshold of 90% was identified based on recommendations and advice from experienced personnel. This level ensures efficient resource use while allowing flexibility for fluctuations and minimising the risk of overburdening workstations. The throughput and production cost for the models serve as further performance measures.

The identification of improvement opportunities is subject to certain constraints. Principal among these constraints is the prerequisite that, when considering the addition of an additional workstation to the system, it is required to determine whether the machinery, for the additional workstation, is present within the job shop. If the machinery is not present and needs to be purchased, then the additional workstation cannot be identified as a possible improvement initiative.

6.1 Improved Process Flow

There are many benefits to improving a system's process flow [68]. Some of these benefits include better communication between various departments within the system, standardisation of the system, the ability to deliver comprehensiveness, the possibility of maximising resources, and various other relevant factors [68].

It is important to highlight that the improved process flow was a side project that took place in parallel with this study. Therefore, a mere discussion and background are provided regarding the

improved process flow.

The assessment of the process flow initiates with the development of a From-to-Chart, to be discussed in the next subsection. Thereafter a comparison of travel distances between successive workstations contributes to the evaluation of the current and proposed state. Ultimately, the improved process flow of the pipe manufacturing is identified.

6.1.1 From-to-Chart

To evaluate the current relationships between the workstations, a From-to-Chart is developed. The From-to-Chart is briefly defined as a chart that displays the relationship between a set of points [69]. The chart is developed by first allocating a number to each workstation, as displayed in Table 6.1 below.

Table 6.1: Number Allocation to Workstations for From-to-Chart

Number	Workstation
1	plate_store
2	s_line
3	dress
4	roll
5	l_weld
6	assem
7	c_weld
8	inspect
9	disp

It is also necessary to determine evaluation criteria, which are used to identify the necessity of workstation adjacency. The required evaluation criteria are listed in Table 6.2 below.

Table 6.2: Colour Codes Evaluation Criteria for From-to-Chart

Value	Closeness
A	Necessary
B	Important
C	Ordinary
D	Unimportant
E	Unnecessary

When focusing on the developed From-to-Chart, the allocation of an adjacency value is based on the material flow between the workstations as well as the influence of activities performed in the specified workstation on another workstation. The developed From-to-Chart, which depicts the current state of the workstations' adjacency, is displayed in Table 6.3 below.

Table 6.3: From-to-Chart depicting Current State

From/ To	1	2	3	4	5	6	7	8	9
1		E	E	E	E	D	D	D	D
2			B	C	E	E	E	E	E
3				B	E	E	E	E	E
4					E	E	E	E	E
5						E	E	E	E
6							B	C	D
7								D	D
8									C
9									

An improved adjacency relationships is depicted in Table 6.4 below. This chart is iteratively developed by taking various factors such as crane utilisation, material flow, and adjacent workstation requirements into account. The main focus was to improve the process flow and consequently centralise the process.

Table 6.4: From-to-Chart depicting Proposed Improved Process Flow

From/ To	1	2	3	4	5	6	7	8	9
1		C	D	D	D	D	D	D	D
2			B	D	D	D	D	D	D
3				B	D	D	D	D	D
4					B	D	D	D	D
5						B	D	D	D
6							B	D	D
7								B	D
8									C
9									

By comparing the two charts, as displayed in Table 6.3 and Table 6.4 above, it is evident that there is a difference in the current state and the state where an 'ideal' workstation adjacency is displayed. It can be seen from Table 6.3 above that the various workstations are placed far from one another, not ensuring an effective process flow. The only workstation adjacency that could be classified as adequate includes the *s_line* and the *dress* workstations, the *dress* and *roll* workstations, and the *assem* and *c_weld* workstations. The remainder of the workstations are positioned in such an undesirable manner. Thus, the current positioning of workstations leads to long transportation times of resources and materials, unnecessary travel distances, and a redundant loss of production time.

The closeness design aspect is better visualised using an activity relationship chart, as can be seen in Figure 6.1.

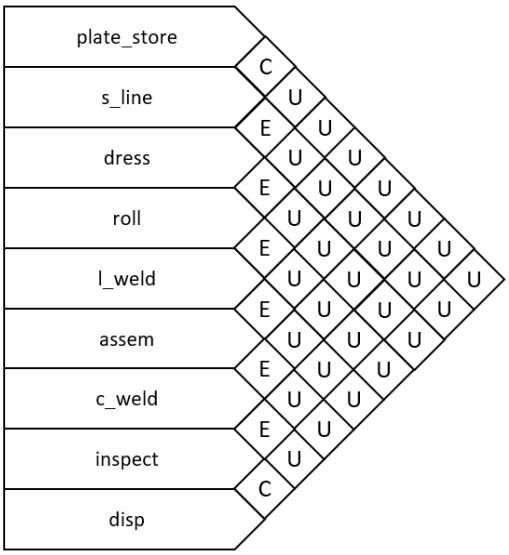


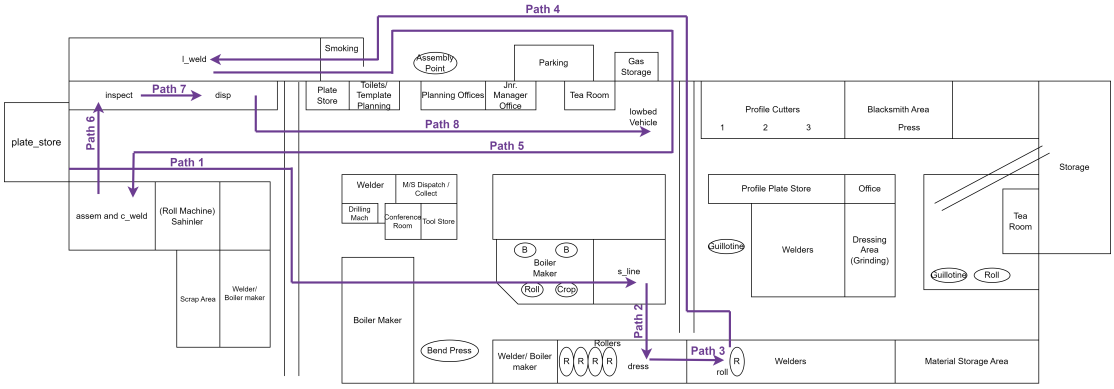
Figure 6.1: Activity Relationship Chart

It can be observed from Table 6.4 and Figure 6.1, above that the workstations are positioned in such a manner to ensure that adjacent workstations adhere to adequate closeness. The identified improved From-to-Chart, and activity relationship chart, provide the possibility of eliminating redundant loss of production time and the possibility of reducing transportation times of resources and materials. With the recognition of the benefits of placing adjacent workstations closer together, the subsequent task is to evaluate possibilities to achieve this requirement.

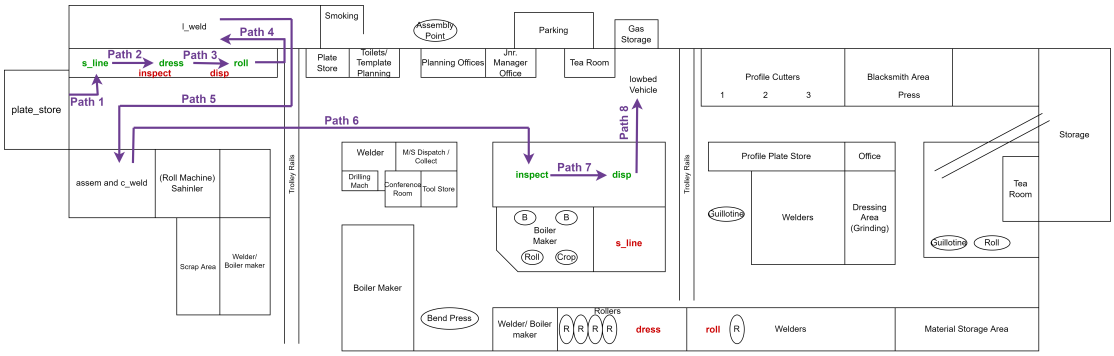
6.1.2 Evaluating the Current State and Identifying a Proposed State

The network paths between consecutive workstations, for the current state, are visualised in Figure 6.2a below. Based on the illustration in Figure 6.2a it is evident that the material flow exhibits disarray and lacks organisation.

A proposed improved process flow is introduced, as can be seen in Figure 6.2b below. The workstations identified in 'red' are the workstations' previous locations, whereas the workstations in green are the proposed positions. It is important to note that the improved workstation allocations are identified with the use of the developed From-to-Chart in Table 6.4 and with the guidance and assistance of knowledgeable employees of The Company. The proposed workstation allocations are strategically identified, taking various factors such as other job requirements, space requirements, and resource availability, into account.



(a) Current State



(b) Proposed Improved State

Figure 6.2: Path Numbering Associated with Distance Travelled

To obtain a better visual comparison between the current and improved process flow, a Spaghetti Diagram is developed. A Spaghetti diagram is a visual representation used to portray the physical flow of intricate pathways of materials, information, or activities through a given space or system [70]. The developed Spaghetti diagram is depicted in Figure 6.3, where the red line reflects the current state's flow, and the green line represents the improved workstation allocation's process flow. From the diagram, it can be seen that the current state's flow is busy and unorganised. Whereas, the proposed improved process flow, depicted by the green lines, visualises a more ordered process.

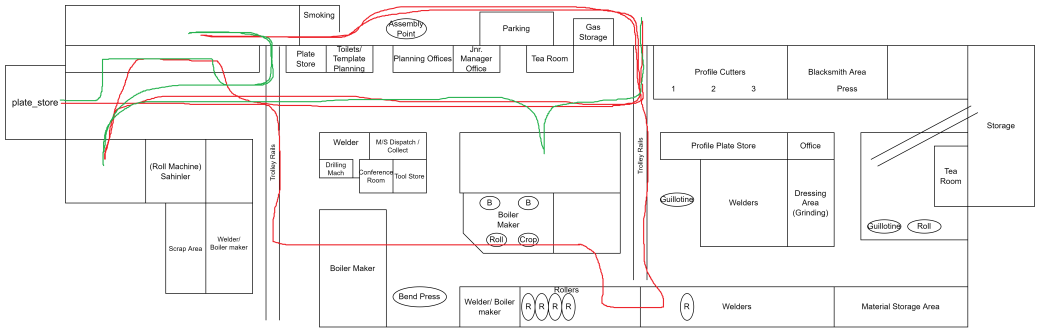


Figure 6.3: Current and Proposed Workstation Allocations' Spaghetti Diagrams

Besides the improved process flow that is present with the proposed state, the following benefits can also be obtained:

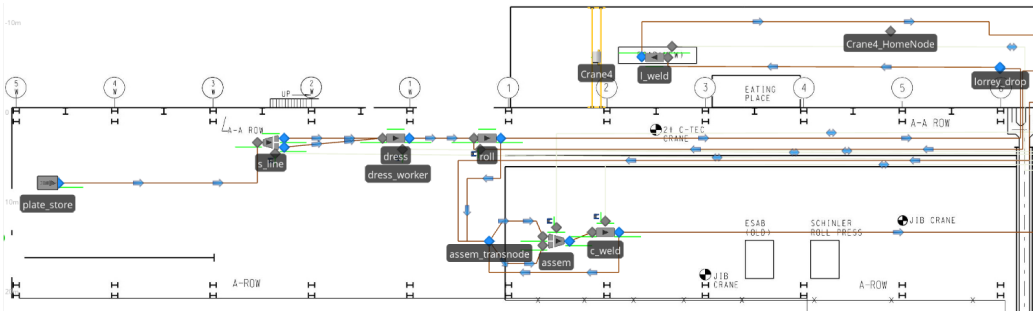
- The system is more centralised.
- Only Bay 2 and Bay 3 are used, whereas other activities within the job shop can utilise the space in Bay 1.
- Paths between consecutive workstations are shorter, which in turn will lead to shorter transportation time between workstations. Further elaboration on *Path Distances* takes place in the next

section.

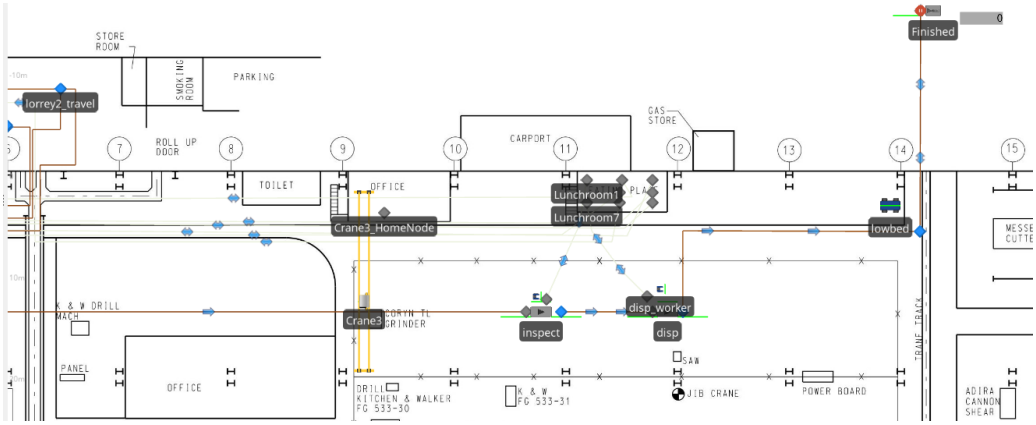
To effectively evaluate the effect of the suggested improved process flow, it is advised to model the improved flow in Simio and consequently compare the current and proposed improved state. To accurately model the improved state, a copy of Model-0.0 is created, referred to as Model-1.0. The primary alterations employed to the duplicate of Model-0.0, namely Model-1.0, includes the following:

- The *s_line*, *dress*, and *roll* workstations' improved destinations are identified to be located in Bay 2, opposite the *assem* and *c_weld* workstations. This re-location of workstations can be observed in Figure 6.2b above.
- Due to no workstations - applicable to the pipe manufacturing operation - being present within Bay 1, it is not necessary to utilise the vehicle object, namely *vehicle1*, that transfers entities from Bay 1 to Bay 3. Instead, the entities are transferred from Bay 2 to Bay 3 by utilising the identified networks. Also, the two overhead cranes present within Bay 1 are not utilised within the improved model.
- The *inspect* and *disp* workstations' proposed location is identified as the area opposite the *Jnr. Manager Office* and *Tea Room* within Bay 2. This location is ideal as the area is currently mainly used as a storage area. Additional aspects that make this location ideal for the mentioned workstations are the space requirements as well as the position of the area. The identified area not only provides sufficient space to store the 12-metre pipes but also enables a more effective and organised manner to transfer the pipes from the stored area onto the desired vehicle, namely *lowbed*, to be transferred to customers.
- All worker objects, with the required and applicable nodes and networks, are re-developed to suit the relocation of the workstations.
- All the necessary properties, processing times, operational factors, and processes, are developed to be the same as for the current state.

A visual representation of the improved state, modelled within Simio, is displayed in Figure 6.4 below.



(a) Snippet 1 - Left Side



(b) Snippet 2 - Right Side

Figure 6.4: Improved State Modelled in Simio

6.1.3 Transportation Distance between Consecutive Workstations

Building on the observations and insights obtained, it will be beneficial to compare the current state and the proposed state's transportation distances to adequately evaluate the effect of re-locating workstations. Due to Simio's capabilities to adequately import a floor plan drawn to scale, and effectively reflect the floor layout with the correct dimensions, the distances are calculated and determined using Simio. The use and specifications regarding the floor layout were discussed in Chapter 5, in Section 5.2.1.

To adequately determine the distances of the paths followed, as displayed in Figure 6.2 above, a vehicle object is used to measure the relevant travel distance between consecutive workstations. The desired results are achieved by adhering to the following steps:

1. A vehicle object was identified, namely *vehicle_test_distance*.
2. A floor label, which is a marker or label placed on the simulated floor to provide specific information within the simulation model [1], are identified. The label is developed to reflect the specified distance travelled. The required argumentation that is developed within the floor label, to obtain the relevant output, is as follows:

$$Math.Round(vehicle_test_distance[1].TotalDistanceTraveled, 2), \tag{6.1}$$

   where the *Math.Round* argument, in conjunction with the identified *2*, is specified to identify that the provided value must be rounded to two decimal places. Furthermore, the argument states that the *vehicle_test_distance* object is the object under consideration, and the *TotalDistanceTraveled* argument states that the distance travelled by the object must be provided.
3. To obtain the various distances between consecutive workstations, it is required that the vehicle object's properties reflect the requirements pertaining to the specific path being followed. The vehicle object's specifications that need to be adjusted, depending on the path that is being measured, include identifying the relevant *Home Node*, as well as setting the relevant node destinations for the vehicle object to travel on.

The calculated and measured distances for both states are provided in Table 6.5 below.

Table 6.5: Distance Travelled on Paths for Current and Proposed Workstation Allocations

Path Number	**Distance Travelled Current State (metre)**	**Distance Travelled Improved State (metre)**	**Percentage Difference (%)**
Path 1	210	49	329%
Path 2	14	5	180%
Path 3	9	9	0%
Path 4	200	82	144%
Path 5	316	123	157%
Path 6	20	99	-80%
Path 7	18	17	6%
Path 8	99	25	296%
Total	886	409	117%

By examining Table 6.5, it is apparent that the proposed state generally results in shorter path distances for all considered *Path N*'s, except for *Path 6*, when compared to the current state's configuration. Furthermore, it is observed that the improved workstation allocation's path distances are approximately 117% shorter compared to the current state's path distances. It can, therefore, be stated that the improved state reflects a significant improvement in travel distance between consecutive workstations.

After further analysis of this parallel project took place, it was successfully motivated that the improved state must be implemented within the MechFab job shop.

6.1.4 Implementation of Improved Process Flow

The improved state is employed by ensuring access directly from Bay 2 to Bay 3. The access is achieved by constructing an opening in the wall, that restricted access. The schematic of the MechFab job shop in Figure 6.5 below illustrates where the opening in the wall took place. A visual demonstration is provided in Figure 6.6 below which displays the wall under consideration.

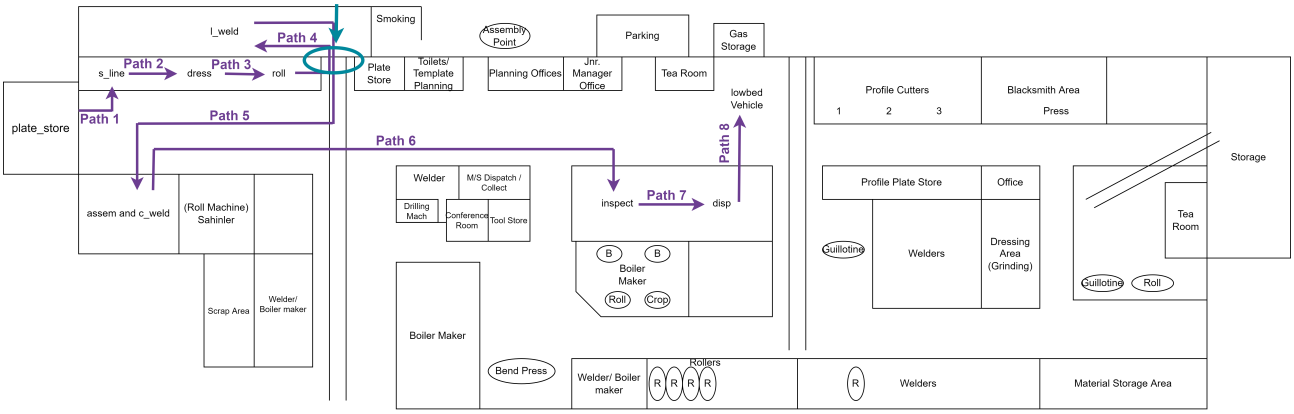


Figure 6.5: Allocation of Wall-Restriction

Figure 6.6: Wall Restriction

The construction of the opening in the existing wall was successfully finalised by November 2022. The opening in the wall is approximately 3,500 mm in width and 3,000 mm which ensures the effective transportation of canisters. Figure 6.7 below illustrates the constructed opening in the wall. More details regarding the opening in the wall can be observed in Appendix K.

Figure 6.7: Opening in Wall Visualised from *ESAB-room*

6.2 Improvement Opportunities Analysis

This section focuses on identifying and evaluating improvement opportunities within the pipe manufacturing system. To achieve this objective, two factors were concurrently applied and considered: TOC principles and relevant performance measures.

6.2.1 Theory of Constraints

As observed from the literature in Section 3.7, the TOC, developed by Dr. Eliyahu M. Goldratt, provides a comprehensive framework for identifying and managing the limiting factors that impede a system's ability to achieve its goals [47]. The framework offers a systematic approach to identify, exploit, subordinate, and elevate constraints within a system, leading to improvement opportunities, enhanced throughput, and ultimately increased profitability.

6.2.2 Performance Measures

The performance measures used to identify system constraints and, in turn, identify opportunities for improvement include:

1. Utilisation
2. Throughput
3. Production Cost

It is important to note that all performance measures are captured and evaluated at the end of the simulation run. Therefore, the results obtained reflect the proceedings of 4.5 weeks of operation.

Utilisation

The *utilisation* performance measure focuses on the efficiency and operational effectiveness of the workstations within the pipe manufacturing system. By default, Simio tracks the following states for the applicable resource objects [1]:

- Starved: The resource object is idle and waiting for an entity.
- Processing: The resource object is processing one or more entities.
- Blocked: The resource object has completed processing, but the entity cannot release the resource object.
- Failed: The resource object has failed.
- Off-shift: The server uses a schedule and is off-shift.

The *processing* state is identified as the state to measure the utilisation of the respective workstation [1]. I.e. the utilisation performance measure is a measure of the percentage time that the workstation was busy processing.

Throughput

The *throughput* performance measure is defined as the total number of items (12-metre pipes) that enter the Sink resource after the simulation run [1].

To effectively comply with the requirements to adequately utilise the throughput performance measure, it is important that the Source object, namely *plate_store*, is developed accordingly. The model is developed to ensure that there is always an entity waiting for the first station, which is the *s_line* workstation. By implementing this argument, it is ensured that the first station will never be in the *starved* state. This argumentation is equivalent to assuming that there is an infinite supply of raw material for the line. Therefore, the throughput performance metric can be accurately assessed, with the system itself being the sole determinant of the number of entities passing through it, unaffected by the supply of entities to the system.

Production Cost

The *production cost* performance measure is utilised to evaluate the effect on cost factors when improvement initiatives are implemented. The measured expenditures that are evaluated within the pipe manufacturing system, include the following:

- **Internal Labour:** Refers to the expenses incurred by The Company for the wages, salaries, and benefits paid to the employees.
- **Material:** The cost refers to the expenses incurred for procuring or acquiring raw materials. The only input material that forms part of this cost factor is the cost of plates.
- **Miscellaneous:** The cost refers to a category of diverse and infrequent expenses that do not fall into specific predefined categories. Within the context of this study, the miscellaneous cost serves as a cost for unforeseen expenditures.
- **Tools:** The cost related to tools is mainly used for welding consumables.

The production cost, in the context of this study, serves as a means to compare the cost of production with the throughput of the system. The study is not focused on profit or any sort of financial gain. The evaluation of production cost, rather serves as a metric to exploit the effects of the system when alterations and improvements are done.

As previously noted, the pertinent data have been adjusted to safeguard proprietary design information from being disclosed. However, meticulous attention was given to ensure that the statistics faithfully represent reality. Therefore, techniques such as factor-reduction techniques were employed to protect The Company's data and information.

6.2.3 Model-1.0

Due to the improved process flow being implemented within the MechFab job shop, Model-1.0 is the first model to be evaluated. It is important to note that Model-1.0 serves as a means to determined what the results of the 'base' performance measures are. The identification of bottlenecks will commence in the next model iteration.

Utilisation

To adequately measure the utilisation of the various workstations within Model-1.0, the *Status Pie Chart* functionality within Simio is used to track the states of the workstations. The Status Pie Chart illustrates the percentage of time that the given resource object, i.e., workstations, was in the respective state [1]. For demonstration purposes, Figure 6.8 illustrates an example of the states in which the *c_weld* workstation was for Model-1.0.

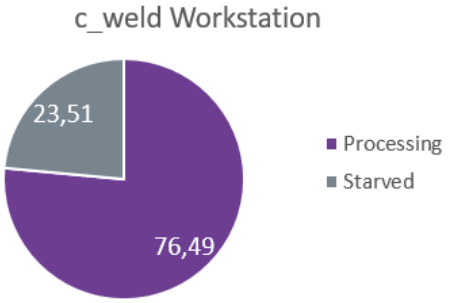


Figure 6.8: Representation of a Status Pie with States

From Figure 6.8 above, it can be observed that the *c_weld* workstation was utilised (processing) 76.49% of the time, while for the remaining 23.51%, the workstation did not have enough material to process (starved).

The utilisation results of the workstations are determined and evaluated through the development of Status Pie Charts, presented in Figure 6.9 below.

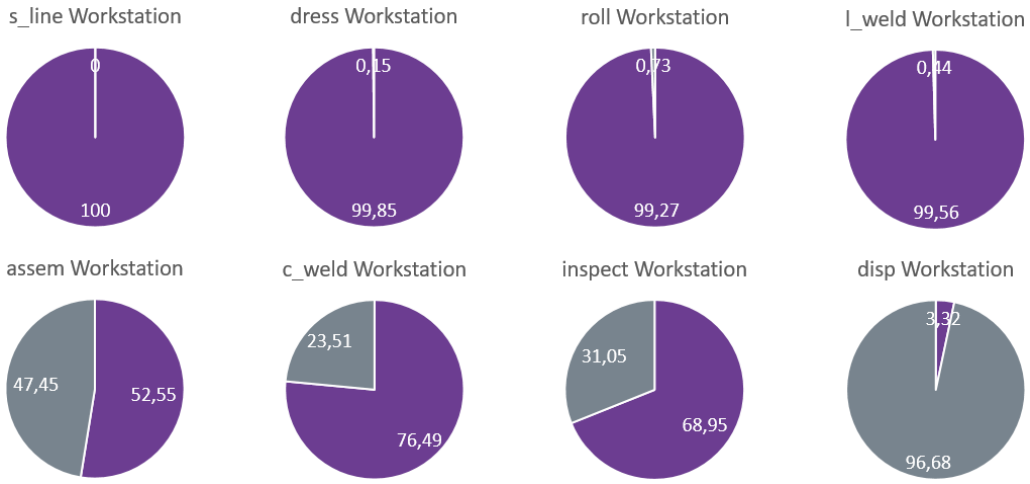


Figure 6.9: Status Pie Charts for Model-1.0

From the charts in Figure 6.9 above, it can be observed that the *s_line*, *dress*, *roll*, and *l_weld* workstations are close to 100% busy processing during the simulation run. The observation was expected for the *s_line* workstation as it is modelled to ensure that the first workstation, namely the *s_line* workstation, always has buffer material to produce.

Furthermore, it can be observed that the *assem* workstation is approximately only 52.55% of the time processing and the remaining 47.45% of the time the workstation is starved. Due to the *roll* and *l_weld* workstations being the workstations that supply the *assem* workstation with material, it can be stated that the *roll* and *l_weld* workstations are the first bottlenecks to be addressed.

Ideally, only one bottleneck must be addressed at a time, however, due to the nature of the system and the fact that the *assem* workstation's supply comes from both workstations, the *roll* and *l_weld* workstations will be further addressed simultaneously. Changes and alterations, regarding the aforementioned observations, are done with the next developed model, namely Model-2.0.

Throughput

To effectively determine the throughput of the system, *Experiments* are utilised. Experiments in Simio refer to a feature that allows users to systematically investigate and analyse different scenarios within a simulation model [1]. It enables users to define variations in input parameters, components, or logic to represent various conditions or situations. [1].

An Experiment is built by identifying and developing a *response* [33]. The response is further constructed by identifying and building the required expression. The relevant expression required to obtain the throughput of the system is defined as follows:

$$Finished.InputBuffer.NumberExited, \tag{6.2}$$

where the first input property states that the resource object being evaluated is the Sink object, namely *Finished*. Secondly, it is stated that the input buffer of the Sink object must be evaluated. This is identified by including the *InputBuffer* argument. Finally, the measuring criteria are identified by providing the argument of *NumberExited*, which returns the number of entities that exited the input buffer of the Sink object.

The throughput results obtained, by running 10 replications of Model-1.0, resulted in 19.2 entities (12-metre pipes). There are various reasons for the 0.2 fractional part present within the throughput

result of 19.2 entities. The non-integer result is likely attributable to statistical variation or random fluctuations within the system [1]. The randomness in the model can cause slightly different results to be obtained during different simulation runs. Thus, it can be stated that a total of 19 entities is delivered as throughput at the end of the simulation run for Model-1.0.

Production Cost
As already mentioned, four cost factors make up the production cost. These cost factors include internal labour cost, material cost, miscellaneous cost, and the cost of tools.

The first cost factor evaluated is the **Internal Labour** cost. This cost factor entails allocating a cost related to the relevant discipline at each workstation. For example, the personnel that is present at the *dress* workstation adheres to the "grinding" discipline, and the personnel that is present at the *l_weld* workstation adheres to the "welding" discipline. To quantify the cost of labour, a Rand value is allocated to each discipline. Each workstation's allocated Rand value, depending on the relevant discipline present at the workstation, is summed together to acquire the total cost of labour at the specific workstation. The relevant disciplines with the associated Rand-value are provided in Table 6.6 below. The provided data (Rand values) have been adjusted to safeguard against the disclosure of proprietary design details. Nonetheless, measures were implemented to ensure that the results represent real-world conditions.

Table 6.6: Rand value per hour Associated with Discipline

Discipline	Rand value per hour
Grinding	R 156
Boilermaker	R 429
Welding	R 429
Planning, Clerk, Dispatching, Inspection	R 0.01

With reference to Table 6.6 above, the *Planning, Clerk, Dispatching, Inspection* discipline (activities) is assigned a R-value of R0.01 per hour. This Rand value is assigned to state that the activities do take place, however, there is no labour cost related to the activities.

The workstations' disciplines, i.e., the skill set that the worker must have to adequately complete tasks at the workstation, are listed in Table 6.7 below, with the corresponding Rand value per hour.

Table 6.7: Discipline Associated with Specific Workstation with Corresponding Rand value per Hour

Workstation	Discipline	Cost per Hour
plate_store	Plate transporter	R 0.01
s_line	Boilermaker	R 429
dress	Grinding	R 156
roll	Boilermaker	R 429
l_weld	Welding	R 429
assem	Boilermaker	R 429
c_weld	Welding	R 429
inspect	Inspection	R 0.01
disp	Dispatching	R 0.01

Thus, for each hour that a worker is busy processing at the specified workstation, the allocated Rand value is summed together to deliver a total amount that correlates to the workstation's labour cost.

To effectively achieve and determine each workstation's labour cost within Simio, *Cost Centres* are developed. Cost centres are used to define areas of responsibility where costs are allocated and incurred [1].

A *LabourCost* cost centre is developed to consequently sum all the workstations' cost centres together to retrieve the total cost relating to Internal Labour. To achieve the required objective, all the workstations' parent cost centres are assigned to relate with the *LabourCost* cost centre.

After the adequate cost centres are developed, it is required that the results are effectively displayed to consequently deliver the total *Internal Labour* cost. The result of obtaining the total *Internal Labour* cost is determined with the use the Experiment functionality within Simio. The utilised Experiment is defined as *ProductionCost*.

Within the *ProductionCost* experiment, a response, namely *LabourCost*, is assigned which contains the required developed expression. The expression must refer to the *LabourCost* cost centre to display the summed cost centres and in turn, determine the calculated internal labour cost. The required expression reads as *LabourCost.Cost*.

In Table 6.8 below, it can be observed that the required result, after running 10 replications of the model, is R 1 490 410.

The second calculated cost is the **Material Cost** of the plates. It is stated that the supplier of plates charges approximately $R\ 11$ *per kilogram* for the utilised plates. As already mentioned, one plate weighs approximately 1 575 kg. Therefore, one plate's approximate material cost is calculated as follows:

$$1\ 575\ kg \times R\ 11 = R\ 17\ 325. \tag{6.3}$$

As described in Section 2.2.3, two plates are required to manufacture a total of 4 canisters to deliver one 12-metre pipe. Thus, the total material cost present to deliver one 12-metre pipe is calculated as follows:

$$R17\ 325 \times 2 = R\ 34\ 650. \tag{6.4}$$

To effectively display the *Material Cost* factor, the required argumentation is developed into the *ProductionCost* Experiment. A response, namely *MaterialCost* is identified, within the *ProductionCost* Experiment, that will contain the necessary argument to reflect the required value.

Before the material cost can be determined, the number of pipes manufactured, during the simulation run, must first be identified. The desired objective relates to the throughput of the system. As mentioned in Section 6.2.3, the throughput of the system is determined with the use of the expression '*Finished.InputBuffer.NumberExited*'. Consequently, the calculated cost per 12-metre pipe is multiplied by the number of manufactured pipes to determine the total material cost. Hence, the required expression reads as follows: *(Finished.InputBuffer.NumberExited)*34650*. The result obtained by developing the required expression, equals R 665 280, can be observed in Table 6.8 below.

The next cost factor is the **Miscellaneous Cost**. The primary purpose of allocating funds to the miscellaneous cost is to provide financial flexibility to accommodate unforeseen expenses that may arise during the pipe manufacturing process. This cost item is derived from the Labour Cost and, more specifically, is calculated as 20% of the total labour cost.

To effectively determine the miscellaneous cost within Simio, a response, namely *MiscellaneousCost*, is developed within the *ProductionCost* experiment. The calculated *LabourCost* cost element is referenced within the *MiscellaneousCost* response and multiplied by *0.20* to achieve the desired outcome. The developed expression reads as follows: *(LabourCost.Cost)*0.20*. After the simulation run, and the required expression is developed, the value of the miscellaneous cost is determined to be R 298 081.

The final cost factor relevant to effectively calculate the production cost of the system is the cost related

to **Tools**. The allocation of tool costs primarily serves to encompass the welding expenditure within the overall pipe manufacturing cost framework. Thus the tools' cost is a fixed cost of R 14 680. The cost factor is built into the *ProductionCost* Experiment by developing a response, namely *ToolsCost*, and inserting the fixed amount in the response's expression.

The necessary calculations are done and properties are developed to effectively determine the system's **Production Cost**. Therefore, all four cost factors, namely internal labour, material, miscellaneous, and tools costs are added together to consequently deliver the production cost of the system. To effectively deliver the desired outcome, a response, namely *ProductionCost* is first identified within the *ProductionCost* experiment. The four cost factors' expressions are summed together within the *ProductionCost*'s expression to deliver the desired outcome. The result obtained regarding the *ProductionCost* of Model-1.0 is R 2 468 450, as presented in Table 6.8 below.

Table 6.8: Cost Factors Related to Model-1.0

Cost Factor	**Labour**	**Material**	**Miscellaneous**	**Tools**	**Production**
Rand-value	R 1 490 410	R 665 280	R 298 081	R 14 680	R 2 468 450

6.2.4 Model-2.0

The results achieved through the evaluation of Model-1.0's performance measures serve as a foundation for identifying potential areas for improvement.

Utilisation

From analysing Model-1.0's utilisation results, it becomes evident that the first workstation requiring an increase in utilisation is the *assem* workstation. To increase utilisation at the *assem* workstation, it is required that the workstation gains more material (entities) to process.

Due to the system's operations, it can be stated that there are two constraints hindering the *assem* workstation from getting more material. The two workstations that supply the *assem* workstation are the *l_weld* and *roll* workstation. Both workstations are operating at nearly 100% utilisation, as depicted in Figure 6.9 above. This observation strongly indicates that these workstations are acting as bottlenecks for the *assem* workstation.

By adhering to a revised implementation of the TOC principles, the first objective of the development of Model-2.0 is to ensure that the *assem* workstation is supplied with sufficient material. Also, the second objective is to ensure that all the workstations before the *assem* workstation is supplied with sufficient material. In this manner, the TOC principles are applied by addressing the aspects that influence the bottlenecks.

Considering that all entities need to pass through the *l_weld* workstation first, and given that a significant majority (60%) of the material (entities) for the *assem* workstation are sourced from the *l_weld* workstation, it becomes evident that addressing the *l_weld* workstation should take precedence before attending to the *roll* workstation. Hence, the first iteration of Model-2.0, namely Model-2.1, focuses on addressing the *l_weld* workstation.

Model-2.1

To adequately adhere to the TOC steps of *subordinating* and *elevating* the identified constraint, it is deemed beneficial to evaluate the possibility of decreasing the processing time of the *l_weld* workstation. To effectively analyse the effect of a decrease in processing time at the *l_weld* workstation, an analysis is done regarding the effect of decreasing the *l_weld* workstation's processing time by increments of 10%, starting at 10% and ending at 100%. The required objective is developed within Simio by specifying

the relevant incremental decrease, by multiplying the processing time by:

$$1 - N/100, \tag{6.5}$$

where N represents the required percentage decrease. For example, if for the specified modulation run, it is required that the *l_weld* workstation's processing time is decreased by 10%, then *'1-10/100'* will be multiplied by the processing time.

The results obtained, by incrementally decreasing the relevant workstation's processing time by 10%, are evaluated with the use of the Status Pie Chart functionality in Simio. The results obtained are summarised in Table 6.9 below.

Table 6.9: Utilisation of *assem* Workstation for Specific Decrease in Processing Time of *l_weld* Workstations

Processing Time Increment Decrease (%) *l_weld* Workstation	**Utilisation (%) *assem* Workstation**	**Percentage Difference of Utilisation (%)**
0	52.55	0.00
10	71.60	36.25
20	76.61	45.78
30	83.53	58.95
40	88.05	67.55
50	78.42	49.23
60	84.65	61.08
70	96.06	82.80
80	87.53	66.57
90	95.17	81.10
100	85.60	62.89

Table 6.9 illustrates that the *assem* workstation's utilisation results are displayed with the corresponding decrease in the *l_weld* workstation's processing time. It is important to note that each increment decrease is compared with the original processing time of the *assem* workstation. For example, a percentage decrease of 40% at the *l_weld* workstation's processing time is compared with the original utilisation value of the *assem* workstation which is 52.55%. By comparing the two incremental decreases in the *l_weld* workstation's processing time, a percentage difference of approximately 67.55% is achieved.

To effectively identify the percentage to which the *l_weld* workstation's processing time can be decreased, the identification of an exact or universal applicable percentage decrease is not feasible. The appropriate percentage decrease depends on various factors, including the current processing time, the task executed, and the overarching objectives and constraints associated with tasks at the workstation.

A possible approach to determine a feasible percentage decrease is to analyse the expected (pre-allocated) processing times. As previously discussed in Section 4.1.1, a pre-determined time is allocated to a task to be completed, where the employee then, after the completion of the task, 'book' the actual time that he/she took to complete the task. This 'booked' time may or may not be more or less than the time allocated to complete the allocated task. The pre-allocated times are compared with the times 'booked' by the employees to determine whether there is an opportunity to decrease the processing time of the *l_weld* workstation.

It is determined, as observed in Chapter 4 in Table 4.1, that the mean value of the pre-allocated processing time of the *l_weld* workstation is approximately 1.17 hours. The duration that employees typically 'book' for task completion, is approximately 2.02 hours. The percentage difference between the hours that are pre-allocated and the actual time that the worker took to complete the task, is

calculated to be approximately 72.82%. It can, therefore, be assumed with relative accuracy that the *l_weld* workstation's processing time can be decreased by approximately 70%. The utilisation results of the system, after the *l_weld* workstation's processing time is decreased by 70%, are displayed in Figure 6.10 below. To thoroughly assess the ongoing changes made to the system, the outcomes from the previous iteration are visually depicted in the bottom right corner of each workstation's utilisation result, as can be observed from Figure 6.10.

It is important to note that the evaluation and identification of factors to achieve the desired decrease in processing time does not form part of this study and can be analysed in further studies.

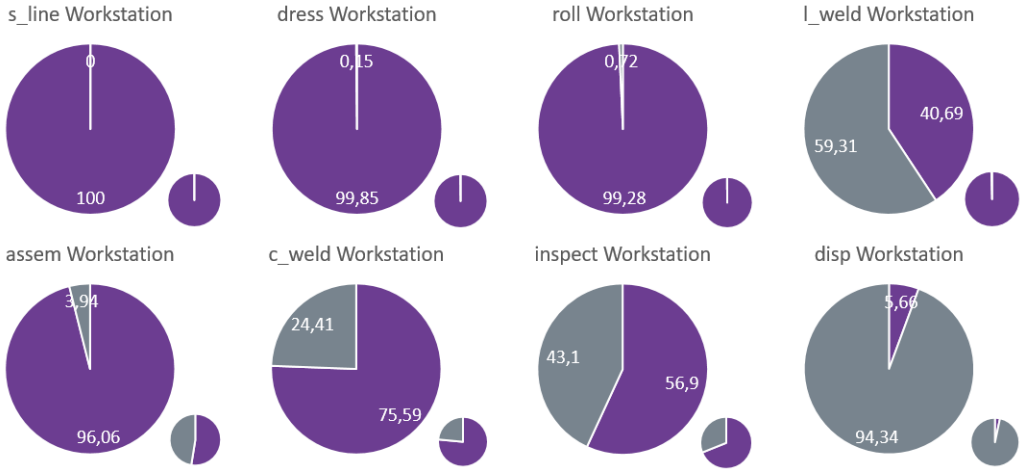


Figure 6.10: Representation of Workstations' Utilisation After *l_weld*'s Processing Time is Decreased by 70%

It can, therefore, be observed from Table 6.10 above that a 70% decrease in the processing time of the *l_weld* workstation, when compared to Table 6.9 and Figure 6.10, results in the *assem* workstation to reflect approximately 96.06% utilisation, rather than 52.55% utilisation.

Model-2.2

Upon reducing the processing time of the *l_weld* workstation by 70%, the result is that the *l_weld* workstation is mostly starved, as presented in Figure 6.10 above. Therefore, the *l_weld* workstation needs more material to process and this is achieved by evaluating the *roll* workstation's proceedings.

To effectively evaluate the possibility of decreasing the *roll* workstation's processing time, the pre-allocated and 'booked' processing times are first evaluated, as can be observed in Table 6.10 below.

Table 6.10: Mean values (hours) of roll Workstation's Two Tasks

	Pre-allocated (hours)	**'Booked' (hours)**	**Percentage Difference (%)**
roll	1.93	1.93	-0.08
re-roll	0.99	1.062	7.33

Table 6.10 above illustrates that the percentage difference for the roll task is approximately -0.08%. The value thus states that the mean 'booked' times are 0.08% faster than the pre-allocated times. Hence, the percentage difference at which the processing time of the roll operation can be decreased, cannot be suggested with certainty, due to the 'booked' time being less than the pre-allocated time. For the re-roll task, the percentage difference is approximately 7.33%. Conversely, when compared to the roll task, the re-roll task demonstrates a clearer distinction.

Due to the aforementioned observations that reflect that the processing times of the *roll* workstation's roll operation cannot necessarily be decreased with certainty, it is advised to propose an alternative.

This alternative includes the investigation regarding the possibility of introducing an additional *roll* workstation within the system.

When adding a new workstation to the system, it is important to conduct a comprehensive evaluation, considering factors such as space efficiency and machinery availability in the job shop. In alignment with the defined project scope and the capabilities of both the pipe manufacturing operation and the MechFab job shop, it is essential to adhere to a notable constraint: the acquisition of additional machinery is not a viable option. However, the possibility of repositioning existing machinery within the job shop can take place.

The possibility of utilising an additional and available roll machine is deemed beneficial to evaluate. The position of an additional roll machine, that is present within the MechFab job shop, is identified with the purple oval symbol in Figure 6.11 below. It is important to note that the additional roll machine adheres to all the requirements needed to deliver the *roll* workstation's objectives. Furthermore, it would be ideal to utilise the identified roll machine as the position is ideal to keep the system centralised.

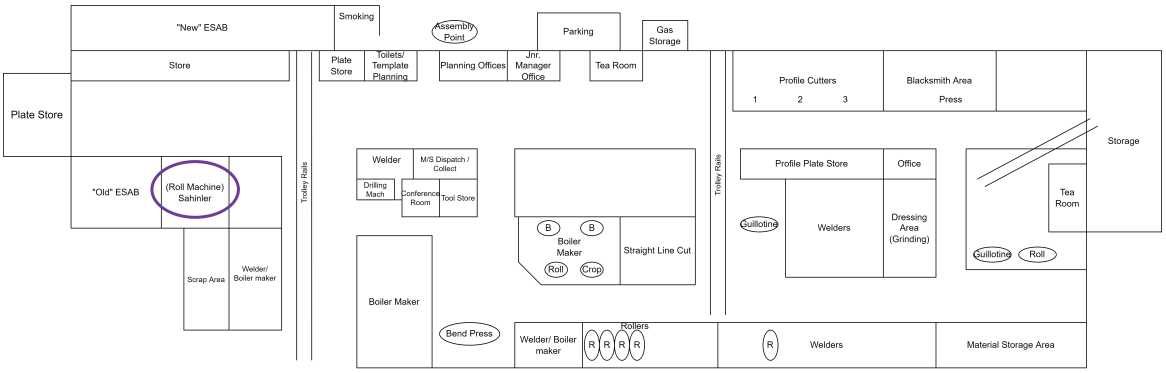
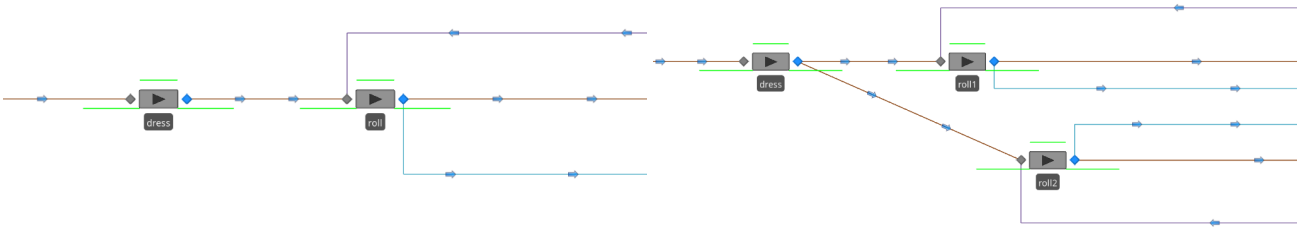


Figure 6.11: Placement of Additional *roll* Machine

Therefore, Model-2.2 is developed within Simio by cloning the *roll* workstation's resource object and inserting the additional workstation in the identified location. For identification purposes, the original workstation is re-named to *roll1* and the additional workstation is referred to as *roll2*. All the necessary properties, objectives, and requirements for the *roll2* workstation are modelled in such a manner to deliver the same deliverables as the original *roll*, re-named as *roll1*, workstation. The material (entities) supplied to the two *roll* workstations by the *dress* workstation is evenly distributed. For demonstrative purposes, a comparison between the roll workstation-area, within Model-2.1 and Model-2.2, is provided in Figure 6.12 below.



(a) *roll* Workstation's Representation of Model-2.1

(b) *roll* Workstations' Representation of Model-2.2

Figure 6.12: Comparison Between Model-2.1's and Model-2.2's *roll* workstation(s)

The utilisation results, for Model-2.2, are displayed in Figure 6.13 below. As already mentioned, to comprehensively evaluate the iterative system changes, the previous iteration's outcomes are visually represented in the lower right corner of each workstation's utilisation result.

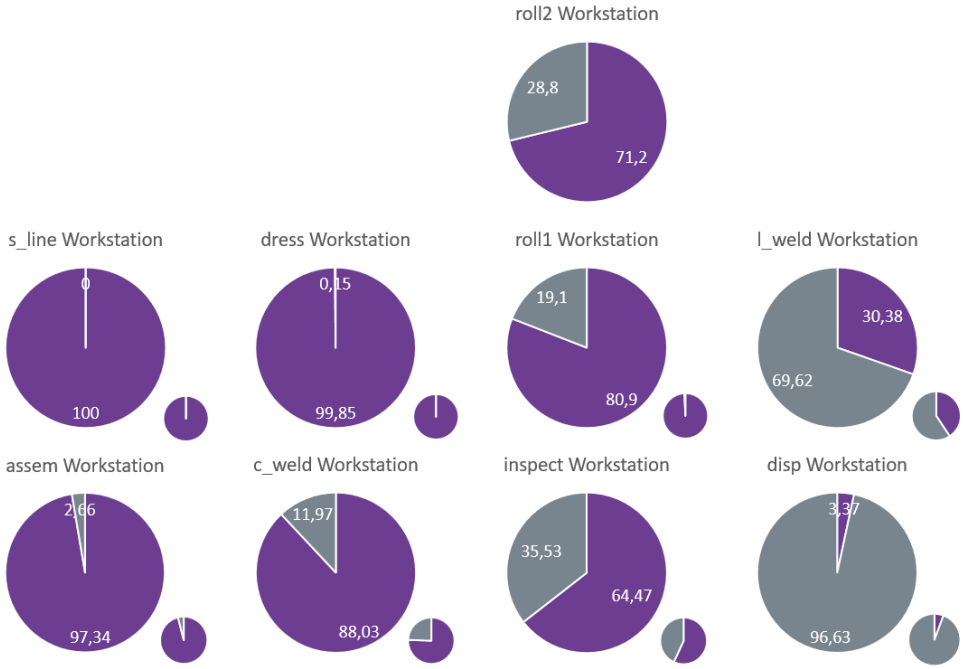


Figure 6.13: Results Obtained for Adding Additional *roll* Workstation

Figure 6.13 illustrates that by adding an additional *roll* workstation, there is not enough supply to ensure that the *roll* workstations are utilised sufficiently. Consequently, the *l_weld* workstation's utilisation also decreased.

Model-2.3

To increase material (entity) supply to both the *roll* and *l_weld* workstations, it is necessary to assess the *dress* workstation, given that the *dress* workstation provides material to the *roll* workstation. The pre-allocated and 'booked' times are observed, as listed in Table 4.1 in Section 4.1.1, and it can be seen that the *dress* workstation's processing time can be decreased by at least 30%. The results obtained by decreasing the *dress* workstation's processing time accordingly, are displayed in Figure 6.14 below.

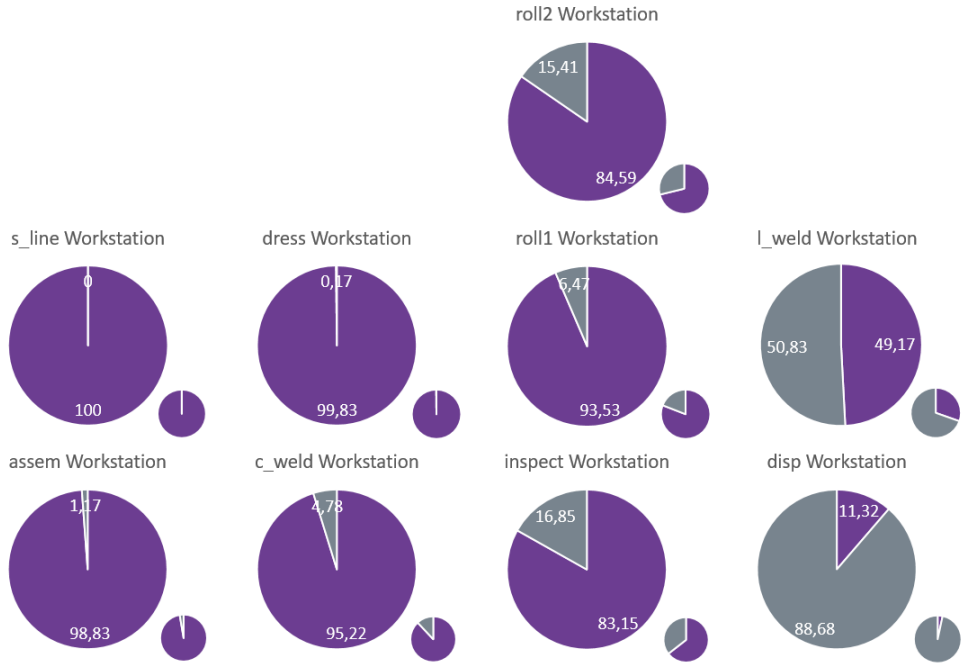


Figure 6.14: Results Obtained for Decreasing *dress* Workstation's Processing Time by 30%

The results presented in Figure 6.14 above illustrate an increase in utilisation for both the *roll* workstations and the *l_weld* workstation. Although an increase is present, it can be observed that the workstations' utilisation can still be increased.

Model-2.4

Model-2.4 further focuses on enabling the *dress* workstation to provide more material to the *roll* workstations and consequently the *l_weld* workstation. To achieve this objective, an additional *dress* workstation is introduced into the system. As already mentioned and listed in Table 2.4, the optimal operation of the *dress* workstation requires approximately 125 m^2 of space to align with its intended objectives. Hence, to adhere to the workstation's space requirements and to ensure that the system stays centralised, the proposed location of the additional *dress* workstation is located near the original *dress* workstation. The placement of the additional workstation is identified in Figure 6.15 below with the purple oval symbol.

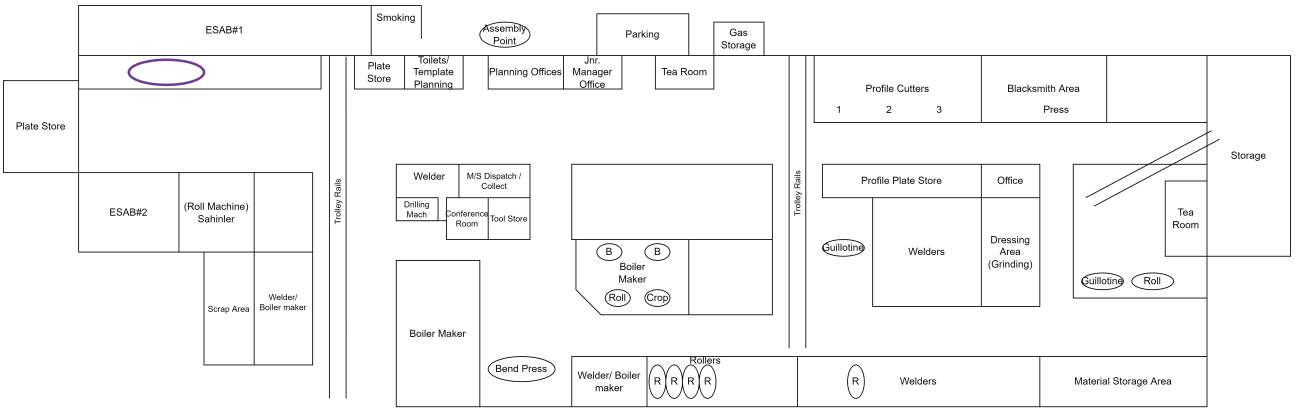


Figure 6.15: Identification of Additional *dress* Workstation

The development of the additional *dress* workstation within Simio was done by ensuring that all the required processes and properties are adhered to. The additional workstation is denoted as *dress2* whereas the original *dress* workstation is re-named to *dress1*. The output of each *dress* workstation is allocated to a *roll* workstation. It is beneficial to note that the same outcome would have been achieved if the output node of each *dress* workstation were identified to equally split the material to the two *roll* workstations. A visual demonstration of the additional *dress* workstation within the system is represented in Figure 6.16 below.

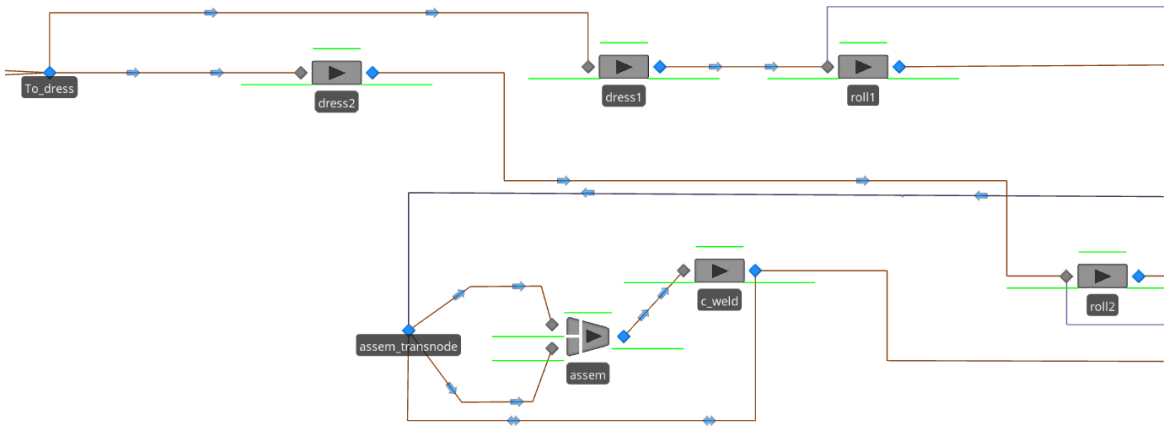


Figure 6.16: Demonstration of Additional *dress* Workstation

The results obtained, with the use of the Status Pie Chart functionality within Simio, are displayed in Figure 6.17 below.

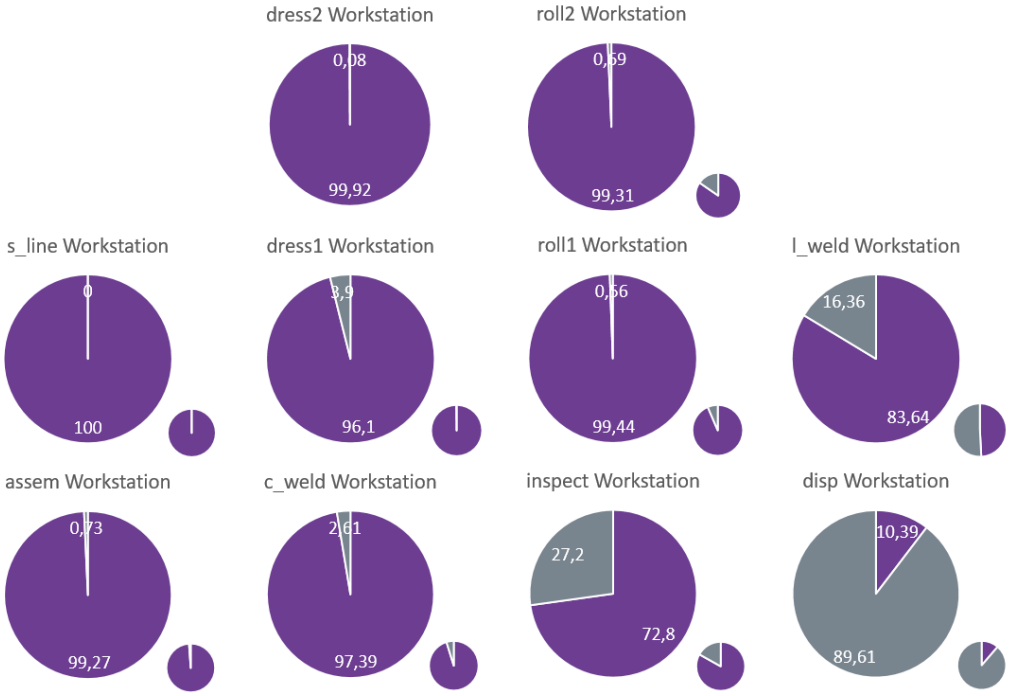


Figure 6.17: Results Obtained for Adding an Additional *dress* Workstation

From the results in Figure 6.17 above, it can be observed that the additional *dress* workstation assists in ensuring that the *roll* workstations utilisation is close to 100%. However, the *l_weld* workstation's utilisation still needs to be increased, where the workstation has material shortages for approximately 16.36% of the simulation run time.

Model-2.5

To evidently increase the material supply to the *l_weld* workstation, it is necessary that the *roll* workstations are addressed. As previously observed in Table 4.1, the roll operation's 'booked' times are approximately 0.08% faster than the pre-allocated times. It can, therefore, not be declared with certainty that the *roll* workstations' processing time can be decreased. However, for the operation of re-rolling the canisters, it is observed that this operation can be decreased by approximately 7.33%. Hence, by implementing adequate improvement strategies within the real-world system, it can be assumed that the workstation's re-roll processing time can be decreased by 10%. The results obtained are displayed in Figure 6.18 below.

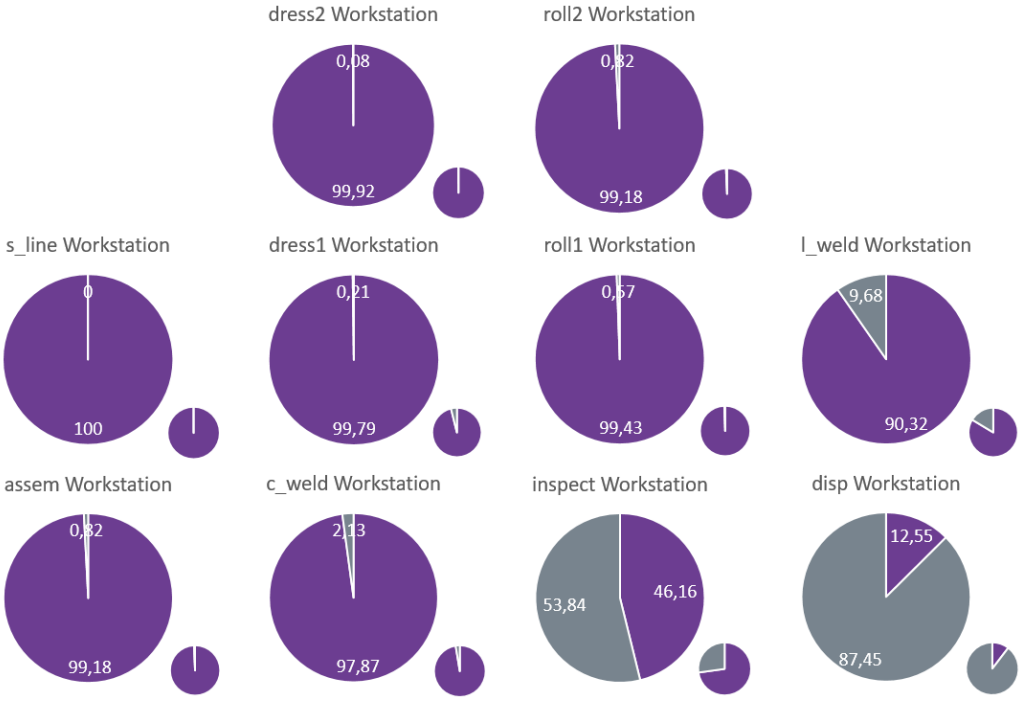


Figure 6.18: Results Obtained after re-roll Operation is Reduced by 10%

Figure 6.18 above illustrates that the *l_weld* workstation's utilisation result increased from approximately 83.64%, as presented in Figure 6.17 above, to approximately 90.32%. The utilisation results, up until the *assem* workstation, is above 90% and is deemed sufficient.

The remaining performance measures, namely throughput and production cost, are discussed in the following subsections. During the analysis of the remaining performance measures, it is important to note that the results are evaluated in an iterative manner for all the versions of Model-2.0.

Throughput

By evaluating the throughput of the five versions of Model-2.0, further insight can be obtained regarding the identification of improvement opportunities within the system. The throughput of each developed model is obtained by utilising a "status label" within Simio. A *status label* is a visual indicator that displays real-time information about the state of an object or entity in a simulation model, enabling users to track and understand system behaviour [1]. It is important to note that this approach differs from the technique used to determine Model-1.0's throughput results, as described in Section 6.2.3. There is no ideal or preferred technique to determine the required objective. The use of the Status Label functionality is, therefore, an alternative method used to determine the throughput results.

The Sink object, namely *Finished*, is used to determine the throughput result of the system after the simulation run. The same argument that was developed for Model-1.0's throughput results is used to determine the throughput results of the different versions of Model-2.0. Hence the argument *Finished.InputBuffer.NumberExited* is specified.

The results obtained, by applying the aforementioned developed status label to the five versions of Model-2.0, are listed in Table 6.11 below. The *Percentage Difference* outcomes listed in the table, represent a comparison between the model under consideration and Model-2.0. I.e., each iteration of Model-2.0 is evaluated against the throughput results of Model-2.0 itself.

Table 6.11: Throughput Results for Different Versions of Model-2.0

Model Number	Throughput	Percentage Difference (%)
2.0	19	0.00
2.1	26	36.84
2.2	28	47.37
2.3	26	36.84
2.4	26	36.84
2.5	24	26.32

It was anticipated that Model-2.5 would demonstrate the highest throughput, however, the results differ. Model-2.2 exhibits the highest throughput rate, reaching 28 entities (12-metre pipes).

It is important to note that the evaluation of Model-2.0 was exclusively conducted on workstations from the *assem* workstation and those preceding it. Thus, it is reasonable to infer that workstations following the *assem* workstation could significantly impact the overall system throughput. As a result, it is necessary to perform further analysis on the remaining workstations. The required analysis is done in the successive model iterations of Model-3.0's.

Production Cost

The production cost of the five versions of Model-2.0 is evaluated to obtain further insight into the various alterations of the system. To obtain the required objective, similar steps were followed as outlined in Section 6.2.3, where the proceedings of determining the production cost for Model-1.0 were elaborated on.

Based on the incorporation of additional workstations into the system, it can be reasonably inferred that the production cost for Model-2.0 will surpass that of Model-1.0. This assumption stems from the understanding that the introduction of additional resource objects, will impact the labour cost, the miscellaneous cost, and the material cost factors. The labour cost factor directly correlates with the number of workers within the system, where additional resource objects will result in additional worker entities. The miscellaneous cost factor is calculated based on the computed labour cost, and the material cost correlates with the amount of entities that are processed by the system. The calculated costs, for the different versions of Model-2.0, are provided in Table 6.12 below.

Table 6.12: Model-2.0's Versions Cost Results

	Model's Rand-value					
Cost Factor	**Model-2.0**	**Model-2.1**	**Model-2.2**	**Model-2.3**	**Model-2.4**	**Model-2.5**
Labour	R 1 490 410	R 1 441 720	R 1 583 210	R 1 838 860	R 2 109 220	R 2 151 150
Material	R 665 280	R 866 250	R 942 480	R 921 690	R 900 900	R 859 320
Miscellaneous	R 298 081	R 288 343	R 316 642	R 367 771	R 421 844	R 430 230
Tools	R 14 680	R 14 680	R 14 680	R 14 680	R 14 680	R 14 680
Production	R 2 468 450	R 2 610 990	R 2 857 010	R 3 143 000	R 3 446 650	R 3 455 380

Table 6.12 illustrates that the production cost increased by approximately 39% when comparing Model-2.0 with Model-2.5's results. As expected, the observation is a result of integrating additional resources into the system to enhance the workstations' utilisation. It will, therefore, be ideal to require a balanced throughput that justifies the increased production cost.

6.2.5 Model-3.0

This section centres on addressing potential constraints within the system that are present after the *assem* workstation.

Utilisation

It can be observed from Model-2.5's utilisation results, as depicted in Figure 6.18 above, that the main constraints within the system are currently the *inspect* and *disp* workstations. Because the *inspect* workstation supplies the *disp* workstation, the initial assessment will focus on enhancing the supply of the *inspect* workstation.

Model-3.1

To effectively increase the supply of the *inspect* workstation, it is reasonable to first evaluate the possibility of increasing the preceding workstation's throughput. From Table 4.1 in Section 4.1.1 it can be observed that the *c_weld* workstation, which is the workstation that supplies the *inspect* workstation, reflects a percentage difference of approximately 52% between the pre-allocated and 'booked' times. Hence, for the purpose of this study, it is suggested that the *c_weld* workstation's processing time is decreased by approximately 52%. The utilisation results obtained, are depicted in Figure 6.19 below.

Figure 6.19: Representation of Workstations' Utilisation After the *c_weld* Workstation's Processing Time is Decreased by 52%

The result of decreasing the *c_weld* workstation's processing time by 52%, is that the *c_weld* workstation reflects utilisation of approximately 39.83%, as can be seen in Figure 6.19. This result was expected due to the significant decrease in processing time of the *c_weld* workstation. Hence, the *c_weld* workstation processes entities faster than what the *assem* workstation can supply.

Model-3.2

The Model-3.2 focuses on increasing the supply of entities at the *assem* workstation. It is observed from Table 4.1 that the *assem* workstation's processing time can be decreased by approximately 22%. The results obtained are provided in Figure 6.20 below.

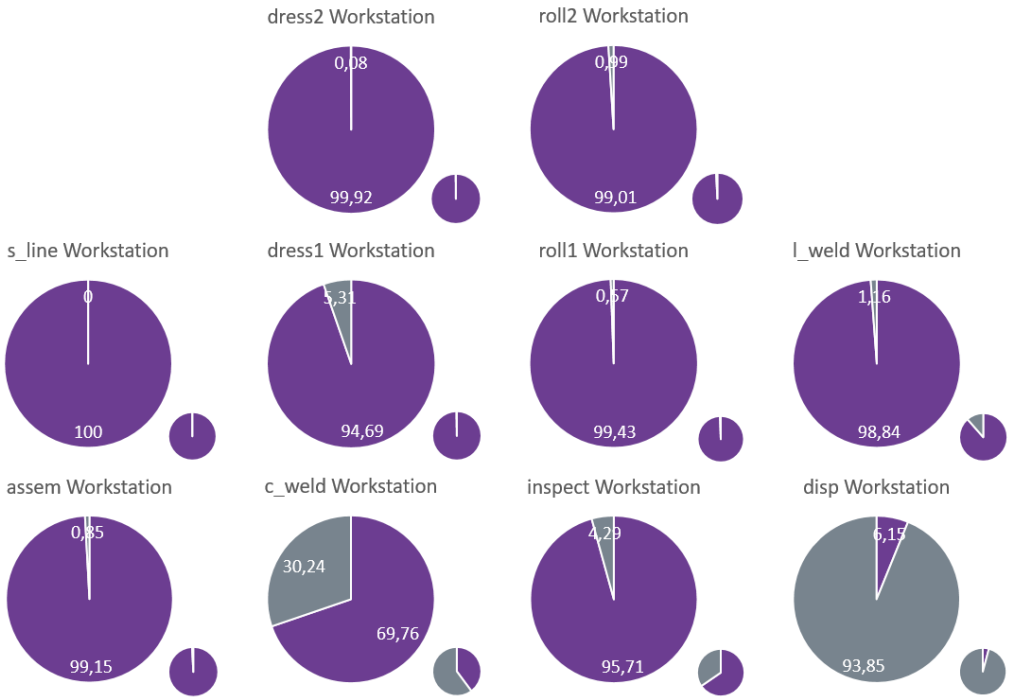


Figure 6.20: Representation of Workstations' Utilisation After the *assem* Workstation's Processing Time is Decreased by 22%

The result in Figure 6.20 above illustrates that the *c_weld* workstation still does not attain enough material to be classified as effectively utilised. Further reduction in the *assem* workstation's processing time is not feasible due to the identified maximum decrease of 22% already implemented. Hence, the possibility of identifying another *assem* workstation is evaluated during the development of Model-3.3.

Model-3.3
Due to the *assem* workstation still not obtaining sufficient material to adequately supply the *c_weld* workstation, the possibility of specifying an additional *assem* workstation are explored.

As evident from the space requirements provided in Table 2.4, it can be observed that the *assem* workstation requires approximately 94 m^2 to operate effectively. A position is identified for the additional *assem* workstation, taking factors, such as ensuring that the process is centralised and space requirements, into account. The location where the additional *assem* workstation is positioned within the MechFab job shop is illustrated in Figure 6.21 below, where the position is highlighted with purple oval circles. The original *assem* workstation is re-defined as *assem1* and the additional *assem* workstation is identified as *assem2*.

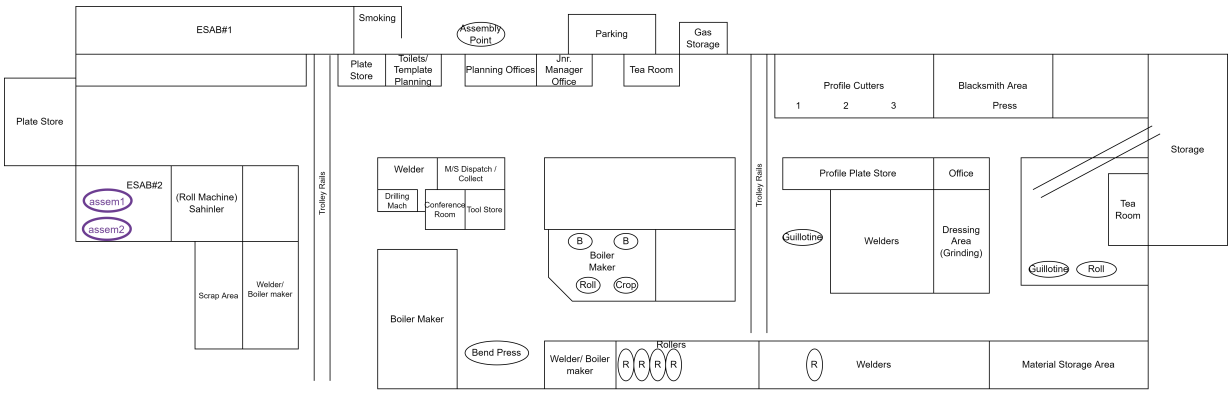


Figure 6.21: Location of Additional *assem* Workstation

The implementation of the additional *assem* workstation is modelled in Simio through cloning the *assem* workstation and ensuring that all operations, processes, and procedures are correct to adequately adhere to the outcome of the additional workstation within the model. Some of the alterations that are developed to ensure that the adjustments comply with what is expected from the model include:

- Ensuring that the *assem1* and *assem2* workstations gain the same amount of entities to process. This argument is obtained by ensuring the supply to each *assem* workstation is divided equally.
- The process that ensures that the required entities are transferred to the correct nodes (Parent or Member node) as explained and elaborated on in Section 2.2.3, are adjusted accordingly to comply with the requirements of the *assem2* workstation.

A visual demonstration of the additional *assem* workstation within the developed model, is depicted in Figure 6.22.

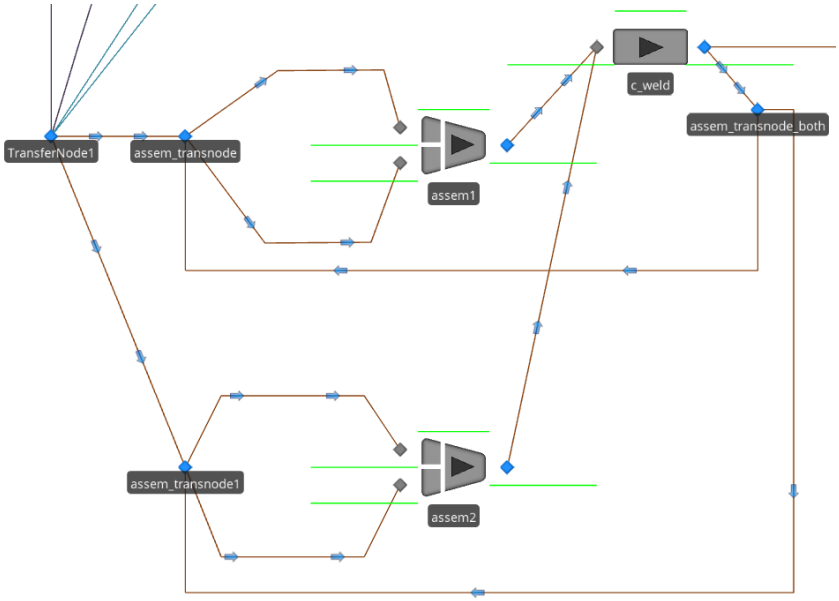


Figure 6.22: Representation of Additional *assem* Workstation within Simio

The utilisation results for Model-3.3 are depicted in Figure 6.23 below.

Figure 6.23: Representation of Workstations' Utilisation After an Additional *assem* Workstation is Added to System

The analysis of Figure 6.23 above reveals a noticeable trend: with the integration of the additional *assem* workstation into the system, several workstations experience periods of being 'starved'. The following model iteration focuses on ensuring that the adequate workstations' utilisation results are sufficient.

Model-3.4

It is important to note that during the build of the previous Model-3.0's versions, the processing times were adjusted according to the capabilities of the workstations, as outlined in Table 4.1 in Section 4.1.1. To ensure that the workstations' depict a utilisation result close to 90%, the relevant workstations' processing times are adjusted accordingly. The *l_weld*, *assem1*, *assem2*, and *c_weld* workstations are evaluated to address the required objective.

Firstly, any alteration done to the original *assem1*, *assem2*, and *c_weld* workstations' properties are removed. This is done to effectively evaluate the alteration's result after the required adjustments are done to the *l_weld* workstation. Thereafter, the *l_weld* workstation's processing time is adjusted to 68%, rather than 70%, as previously modelled during the development of Model-2.1. The result of adjusting the processing time at the *l_weld* workstation to 68% is identified through iteratively evaluation the workstation's processing time. This iterative approach continued until a sufficient utilisation level of 99% was achieved. The results obtained, are displayed in Figure 6.24 below.

Figure 6.24: Representation of Workstations' Utilisation After *l_weld* Workstation's Processing Time is Decreased by 68%

Figure 6.24 illustrates that the objective of the model-iteration was successful due to the *l_weld* workstation's utilisation result reflecting approximately 99%. Also, it can be seen that the *assem* workstations' utilisation results are 91.92% and 69.63% for the *assem1* and *assem2* workstations, respectively. The difference in utilisation results between the *assem1* and *assem2* workstations can be assigned to how the entities arrive at the workstation in batches due to the ride capacities of the vehicles.

Secondly, the *assem1* and *assem2* workstations' utilisation results need to be addressed. Due to the sufficient utilisation results displayed at the workstations that precede the *assem1* and *assme2* workstation, it is suggested to not further alter or decrease the preceding workstations' processing time. An alternative approach to enhance the utilisation of *assem1* and *assem2* workstations involves the identification of an additional *l_weld* workstation. However, given the restriction on investing in additional machinery within the MechFab job shop, and that no other ESAB machine is available, an additional *l_weld* workstation cannot be suggested. Therefore, it is assumed that the utilisation results at both the *assem* workstations are sufficient.

Thereafter, the *c_weld* workstation needs to be evaluated where the objective is to decrease the workstation's processing time to such an extent that the workstation's result reflects a value within the low 90s percentage. This objective needs to be achieved to ensure sufficient material for the next workstation, namely the *inspect* workstation, but still ensuring that the *c_weld* workstation reflects a sufficient utilisation result.

An iterative approach was followed to determine the adequate percentage decrease in the processing time of the *c_weld* workstation. This approach involved reducing the processing time in increments of 10%. The results obtained after the *c_weld* workstation's processing time is iteratively decreased, are

provided in Table 6.13 below.

Table 6.13: Utilisation Results After Incremental Decrease in c_weld Workstation's Processing Time

Percentage Decrease (%)	c_weld Workstation's Utilisation (%)
0	95.63
10	95.65
20	95.63
30	95.53
40	90.15
50	84.92

From Table 6.13 above, it is observed that with a percentage decrease of 50%, the utilisation of the *c_weld* workstation is approximately 84.92%. The evaluation can, therefore, be stopped at 50% due to the utilisation results falling below 90%. By decreasing the *c_weld* workstation's processing time by approximately 40%, a utilisation result of approximately 90.15% is achieved. It can, therefore, be confirmed that the *c_weld* workstation's processing time can be reduced by 40%. The Status Pie Chart results obtained, after the *c_weld* workstation's processing time is reduced by 40%, are displayed in Figure 6.25 below.

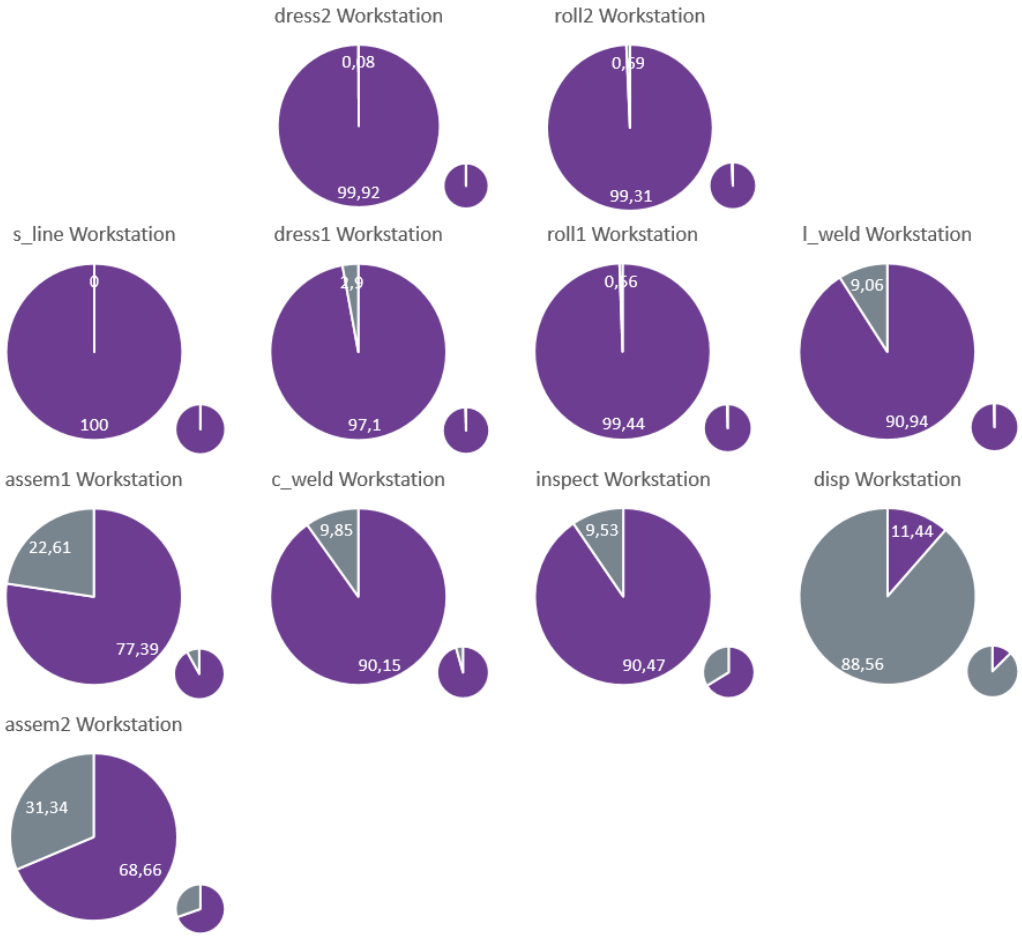


Figure 6.25: Representation of Workstations' Utilisation After *c_weld* Workstation's Processing Time is Decreased by 40%

As depicted in Figure 6.25, the workstations, up until the *disp* workstation, exhibit appropriate utilisation results. An examination regarding the *disp* workstation's utilisation is addressed during the development of Model-4.0, to take place during the next model developments.

Throughput

The throughput for the four different versions of Model-3.0 is determined using the same development technique used to determine the utilisation results for the different versions of Model-2.0, as previously discussed in Section 6.2.4. The results obtained, with the use of Simio's *status label* functionality, are displayed in Table 6.14 below.

Table 6.14: Throughput Results for Different Versions of Model-3.0

Model Number	Throughput	Percentage Difference (%)
3.0	24	0
3.1	30	25
3.2	34	42
3.3	36	50
3.4	42	75

The acquired outcomes depict the throughput achieved following each model run. From the throughput results obtained, it is evident that each iterative model development leads to an increase in throughput for the system. Between the developed Model-3.0 and Model-3.4, there is a tremendous increase of approximately 18 entities which results in a percentage difference of approximately 75%.

Production Cost

The assessment of the production cost of the four different versions of Model-3.0 aims to provide more insights into the systems' modifications. To achieve this objective, the same procedural steps were followed as outlined in Section 6.2.3 and Section 6.2.4. The relevant cost factors related to the different versions of Model-3.0, are provided in Table 6.15 below.

Table 6.15: Model-3.0's Versions' Cost Results

	Models' Rand-value				
Cost Factor	**Model-3.0**	**Model-3.1**	**Model-3.2**	**Model-3.3**	**Model-3.4**
Labour Cost	R 2 151 150	R 1 977 250	R 2 038 190	R 2 225 280	R 2 262 170
Material Cost	R 859 320	R 928 620	R 1 219 680	R 1 226 610	R 1 455 300
Miscellaneous Cost	R 430 230	R 395 450	R 407 638	R 445 056	R 452 435
Tools Cost	R 14 680	R 14 680	R 14 680	R 14 680	R 14 680
Production Cost	R 3 455 380	R 3 316 000	R 3 680 190	R 3 911 630	R 4 184 590

The information presented in Table 6.15 underscores a noticeable trend: as the model progresses through different versions, there is an increase in labour cost. This can be attributed to the increase in utilisation of various workstations, directly increasing labour expenses. Interestingly, a deviation from this pattern is observed during the transition from Model-3.0 to Model-3.1, where the labour cost decreased. One plausible explanation for this could be due to the decrease of approximately 52% in processing time for the *c_weld* workstation. Consequently, entities processed by this workstation are now 52 times faster, creating intervals during the simulation run where the workstation remains underutilised. As a result, this leads to a decrease in the labour rate, as the employees at the *c_weld* workstation are utilised less during the simulation run.

When evaluating the cost related to material, it can be seen that the cost increases as the different versions progress. This was expected as the increase in throughput result is directly related to the material cost being increased. Furthermore, it can be observed that when manually calculating the material cost, not the exact values, as displayed in Table 6.15, will be achieved. This is due to the experimental results being identified by running a certain amount of replications. Therefore, the throughput results obtained, as listed in Table 6.14, will not necessarily reflect the same results achieved by the experiment. The reason for this is that the experimental results represent the average over a certain amount of replications done for the model.

6.2.6 Model-4.0

Utilisation

The analysis focuses on addressing the utilisation result of the *disp* workstation.

Model-4.1

By evaluating the tabulated data, presented in Table 4.1 in Section 4.1.1, a notable observation can be made: the *inspect* workstation has the capability of decreasing its processing time by approximately 85%.

Therefore, it will be ideal to decrease the processing time of the *inspect* workstation to consequently increase the utilisation result of the *disp* workstation. The *inspect* workstation's processing time is decreased accordingly, and the results are illustrated in Figure 6.26 below.

Figure 6.26: Representation of Workstations' Utilisation After *inspect* Workstation's Processing Time is Decreased by 85%

As expected, when the processing time of the *inspect* workstation is significantly reduced, the possibility of the workstation being in the 'starved' state increases. This occurs due to the entities being processed faster which causes the workstation to wait for new entities to arrive, leading to periods where the workstation is not in use.

The requirement for high utilisation values at the *inspect* and *disp* workstations is not a critical factor due to the nature of the pipe manufacturing system. This flexibility is supported by the MechFab job shop environment, where underutilised employees can be effectively used to attend to other tasks. Furthermore, as presented in Table 6.6 above, due to the two workstations having a Rand value of R0.01, these workstations confirm that there are no cost implications when employees are not fully utilised at these stations.

The results obtained, as depicted in Figure 6.26 below, can be declared sufficient and the iterative model development is concluded.

Throughput
To calculate the system's throughput for Model-4.1, the same approach was applied as for the previous models. Consequently, the throughput of Model-4.1 is determined to be 50 entities (12-metre pipes).

This assessment allows for a meaningful comparison with the final iteration of Model-3.0, namely Model-3.4, where the recorded throughput is approximately 42 entities. This comparison reveals an increase of around 19% when examining the percentage difference. This outcome underscores the dynamic improvement made during the iterative modelling process, resulting in a notable enhancement in system throughput.

Production Cost
Similar procedures, as outlined for the different versions of Model-1.0, Model-2.0, and Model-3.0, were followed to determine Model-4.1's production cost. Consequently, the relevant cost factors are listed in Table 6.16 below.

Table 6.16: Model-4.0's Version's Cost Results

	Models' Rand-value	
Cost Factor	**Model-4.0**	**Model-4.1**
Labour Cost	R 2 262 170	R 2 366 350
Material Cost	R 1 455 300	R 1 732 500
Miscellaneous Cost	R 452 435	R 473 271
Tools Cost	R 14 680	R 14 680
Production Cost	R 4 184 590	R 4 586 810

Table 6.16 above includes Model-4.0, a duplicate of Model-3.4, providing a basis for meaningful comparison against the cost-related outcomes yielded by Model-4.1.

The increased labour cost is assigned to the higher utilisation of workstations with greater Rand values. The relevant Rand values can be observed in Table 6.6 above. In addition, an anticipation of increased material costs correlates with the observed increase in throughput. As previously mentioned, the miscellaneous cost is 20% of the labour cost. Therefore it was also expected that the miscellaneous cost would increase due to an increase in labour cost. The cost associated with tools remains fixed, serving as a constant in the cost analysis.

Ultimately, the summation of all the mentioned cost factors is summed together to calculate the production cost. Given that certain cost components exhibit higher values during the simulation run of Model-4.1 in contrast to Model-4.0, it becomes evident that the production cost for Model-4.1 is higher than the cost related to Model-4.0.

6.3 Concluding Remarks: Improvement Opportunities Analysis

This chapter focused on identifying and addressing improvement opportunities within the pipe manufacturing operation through the use of Simio. The chapter commenced with an evaluation of the existing process flow, followed by the strategic repositioning of workstations. To maintain alignment with the study's objectives, a deliberate decision was made to limit the detailed explanation of the improved process flow. Subsequently, the improved process flow was evaluated using the From-to-Chart technique, highlighting improvements achievable through workstation rearrangements. The proposed improved process flow was further evaluated by comparing the distances between consecutive workstations for the current as well as the proposed improved state. After the improved process flow is identified, validated and implemented, the chapter focused on identifying improvement opportunities. This involved a systematic and iterative approach, resulting in the development of various model versions.

Key performance measures, including workstation utilisation, throughput and production cost, were employed to assess the identified bottlenecks within each model iteration. From the iterative model development, it was evident that each model version contributed to enhancing the system's performance by addressing specific constraints.

The chapter underlined the interconnectedness of the identified performance measures and the importance of considering all three concurrently to achieve a comprehensive evaluation of the system model under consideration. The outcome of this iterative process to identify improvement opportunities led to a more efficient pipe manufacturing operation. The different models that were developed are outlined in Table 6.17 below, along with a summary of the alterations made.

Table 6.17: Summary of Developed Models

Model	Description
Model-0.0	Pipe manufacturing process flow before the improved process flow is implemented.
Model-1.0	System that reflects improved process flow.
Model-2.1	Decrease *l_weld* workstations' processing time with 70%.
Model-2.2	Add additional roll machine.
Model-2.3	Decrease *dress* workstation's processing time with 30%.
Model-2.4	Add additional *dress* workstation.
Model-2.5	Decrease *roll* workstations' re-roll operation with 10%.
Model-3.1	Decrease *c_weld* workstation's processing time with 52%.
Model-3.2	Decrease *assem* workstation's processing time with 22%.
Model-3.3	Add additional *assem* workstation.
Model-3.4	Adjust workstations' processing time accordingly to acquire desired utilisation results of workstations.
Model-4.1	Decrease *inspect* workstation's processing time with 85%.

An adaptive strategic approach to TOC principles was applied to identify bottlenecks within the system. The identified bottlenecks were addressed and evaluated with the use of three performance measures. The three performance measures include utilisation, throughput and production cost. The utilisation performance measure results are summarised in Table 6.18 and Table 6.19, whereas the throughput results are summarised in Table 6.20, and the cost factors' results, relating to the production cost, are summarised in Table 6.21 and Table 6.22.

Table 6.18: Utilisation Results - Model-0.0 till Model-2.4

	Utilisation Results (%)					
Workstation	**Model-0.0**	**Model-1.0**	**Model-2.1**	**Model-2.2**	**Model-2.3**	**Model-2.4**
s_line	100	100	100	100	100	100
dress1	99,79	99,85	99,85	99,85	99,83	96,1
dress2	-	-	-	-	-	99,92
roll1	70,53	99,27	99,28	80,9	93,53	99,44
roll2	-	-	-	71,2	84,59	99,31
l_weld	98,9	99,56	40,69	30,38	49,17	83,64
assem1	80,56	52,55	96,06	97,34	98,83	99,27
assem2	-	-	-	-	-	-
c_weld	60,78	76,49	75,59	88,03	95,22	97,39
inspect	65,09	68,95	56,9	64,47	83,15	72,8
disp	9,96	3,32	5,66	3,37	11,32	10,39

Table 6.19: Utilisation Results - Model-2.5 till Model-4.1

	Utilisation Results (%)					
Workstation	**Model-2.5**	**Model-3.1**	**Model-3.2**	**Model-3.3**	**Model-3.4**	**Model-4.1**
s_line	100	100	100	100	100	100
dress1	99,79	99,79	94,69	99,76	97,1	99,64
dress2	99,92	99,92	99,92	99,92	99,92	99,92
roll1	99,43	99,43	99,43	99,44	99,44	99,44
roll2	99,18	99,18	99,01	99,31	99,31	99,31
l_weld	90,32	88,5	98,84	83,8	90,94	99,28
assem1	99,18	99,18	99,15	75,8	77,39	83,47
assem2	-	-	-	62,04	68,66	82,34
c_weld	97,87	39,83	69,76	79,36	90,15	93,23
inspect	46,16	65,44	95,71	93,41	90,47	24,68
disp	12,55	4,13	6,15	4,83	11,44	7,03

Table 6.20: Throughput Results

Model Number	**Model-0.0**	**Model-1.0**	**Model-2.1**	**Model-2.2**	**Model-2.3**	**Model-2.4**
Throughput Result	20	19	26	28	26	26
Model Number	**Model-2.5**	**Model-3.1**	**Model-3.2**	**Model-3.3**	**Model-3.4**	**Model-4.1**
Throughput Result	24	30	34	36	42	50

Table 6.21: Cost Factors - Model-0.0 till Model-2.4

Cost Factor	**Model-0.0**	**Model-1.0**	**Model-2.1**	**Model-2.2**	**Model-2.3**	**Model-2.4**
Labour	R 1 544 030	R 1 490 410	R 1 441 720	R 1 583 210	R 1 838 860	R 2 109 220
Material	R 672 210	R 665 280	R 866 250	R 942 480	R 921 690	R 900 900
Miscellaneous	R 308 807	R 298 081	R 288 343	R 316 642	R 367 771	R 421 844
Tools	R 14 680	R 14 680	R 14 680	R 14 680	R 14 680	R 14 680
Production	R 2 539 730	R 2 468 450	R 2 610 990	R 2 857 010	R 3 143 000	R 3 446 650

Table 6.22: Cost Factors - Model-2.5 till Model-4.1

Cost Factor	**Model-2.5**	**Model-3.1**	**Model-3.2**	**Model-3.3**	**Model-3.4**	**Model-4.1**
Labour	R 2 151 150	R 1 977 250	R 2 038 190	R 2 225 280	R 2 262 170	R 2 366 350
Material	R 859 320	R 928 620	R 1 219 680	R 1 226 610	R 1 455 300	R 1 732 500
Miscellaneous	R 430 230	R 395 450	R 407 638	R 445 056	R 452 435	R 473 271
Tools	R 14 680	R 14 680	R 14 680	R 14 680	R 14 680	R 14 680
Production	R 3 455 380	R 3 316 000	R 3 680 190	R 3 911 630	R 4 184 590	R 4 586 810

A discussion regarding the objectives achieved within this dissertation, as well as future research, is done in the next chapter.

Chapter 7

Conclusion and Recommendations

7.1 Study Overview

The challenges present within the Central Workshops, which are located within an integrated steel manufacturing environment, necessitated a strategic and systematic approach to address operational deficiencies. Recognising the need to identify improvement opportunities, the development of a methodology was proposed. Due to the complexity and unpredictability within a job shop environment, a strategic approach was followed to first focus on one job.

The developed methodology, built on a sequence of steps, emphasised that the system first needed to be adequately understood. Thereafter, the relevant data analysis and cleansing took place with the use of RStudio. Due to the relevant data having a stochastic nature, probability distribution functions were identified for each applicable workstation's processing time. As stated in the literature, it is required to employ distribution functions to effectively reflect the real-world operations within the developed model.

The probability functions were determined through a sequence of steps, mainly employing EDA techniques. The data was first plotted on a histogram where the required hypothesis was developed. Thereafter, the Cullen and Frey graph was used to narrow down three possible distribution fits. Further EDA took place by developing a Q-Q plot, where the plot was used to further evaluate the three distribution fits. To establish a formal interpretation of the potential distributions that have been identified, the *Chi-squared* (χ^2) test was applied. The distribution with the highest probability value was identified as the 'best' distribution fit for the workstation's processing time data under consideration. In the context of datasets where it was not necessary to determine a distribution fit, the mean value of the dataset was used.

Simulation modelling was identified, by employing a comprehensive literature study, as an ideal technique to identify improvement opportunities. Simio Simulation Software -the identified discrete simulation modelling tool- was used to develop the current-state operations of the system under consideration, where the identified distribution fits and mean values served as input data for the model. To ensure the accuracy and reliability of the simulation model, verification and validation took place. The model was verified by ensuring that the model's logic aligned with the system's operations, where validation took place by comparing the simulation results with empirical data from the actual system. Following the thorough verification and validation processes, the model was identified as accurate and correct.

The developed model was used as the basis to identify improvement opportunities. An iterative approach was followed to identify bottlenecks, guided by TOC principles and a thorough assessment of workstation utilisation. The utilisation results also served as the first performance measure to evaluate the model iterations. Thereafter, the remaining performance measures used included the throughput and the production cost of the system. The iterative process continued until satisfactory results were achieved, affirming the correctness and adequacy of the proposed methodology.

Although the model proposed in this dissertation was comprehensive by only focusing on one job within the job shop environment, there still exists future research that needs to be investigated to improve the ability of the model to consider various jobs. A summary of the aspects that are recommended for inclusion in future work is provided in the next section.

7.2 Future Research

The simulation model developed in this dissertation only took one job into consideration. Although the foregoing methodology served as an important tool to identify and evaluate improvement opportunities within the system itself, not all existing jobs were incorporated into the methodology. An opportunity exists to integrate the remaining jobs within the job shop environment into the simulation model.

The task of modelling various jobs is a complex approach due to various factors, such as the heterogeneity of jobs within a job shop environment and the limited predictability of jobs that need to be attended to. Thus, to further address the complexity of integrating additional jobs into the developed simulation model, several approaches can be considered. Firstly, employing advanced scheduling algorithms such as Genetic Algorithms or Ant Colony Optimisation can improve job sequencing and resource allocation [71]. Predictive analytic and machine learning models can assist in forecasting job demands and machine breakdowns, enabling proactive planning and also assisting in determining the areas or jobs where it is needed to identify improvement initiatives [72].

Furthermore, dynamic job prioritisation systems can adapt to changing conditions, and real-time monitoring and control systems offer real-time decision support [73]. This approach could be highly advantageous, given the unpredictable nature of the jobs within the job shop environment. Assessing and exploring the previously mentioned approaches can prove advantageous in further developing a methodology to integrate multiple jobs within a single simulation model. This, in turn, facilitates the identification of opportunities for improvement.

Another important factor to consider for future research is the practical implementation of the identified improvements. Two main adjustments were considered in this study to consequently implement improvement opportunities within the developed model. Firstly, the processing times allocated to employees to complete a task are compared to the actual time that the employee takes to complete the specified task. The proposed and actual processing times are compared against one another, with instances where the actual time exceeds the proposed duration, identifying an opportunity for enhancing the task's efficiency to align with the originally suggested processing time. Further research is required to determine practical actions or approaches to ensure that the task under consideration adheres to the proposed processing time.

The second type of adjustment involves the identification of the need for additional workstations. Further research is essential to ascertain the feasibility of integrating additional workstations into the job shop environment. This investigation should encompass considerations such as the recruitment of additional employees, the procurement of necessary tools and resources, and other factors integral to this decision-making process.

It is worth highlighting that the proposed methodology, in combination with the incorporation of supplementary approaches in further studies, is not confined solely to the MechFab job shop. Instead, the methodology can be developed to have the potential for broader application across other job shops that form part of the Central Workshops. Through the diligent application of further research and integration of the advanced simulation model, the proposed methodology will serve as a tool for identifying and assessing opportunities for improvement. Consequently, the methodology will yield significant benefits, not only for the Central Workshops but for The Company as a whole.

Bibliography

[1] W. D. Kelton, J. S. Smith, and D. T. Sturrock, "Simio and simulation: Modeling, analysis, applications third edition," 2013.

[2] G. T. A. Welding, "American welding society," *Welding Handbook, 8th*, 1991.

[3] E. W. Weisstein, "Normal distribution," *https://mathworld. wolfram. com/*, 2002.

[4] L. Kuipers and H. Niederreiter, *Uniform distribution of sequences*. Courier Corporation, 2012.

[5] F. Proschan and P. Sullo, "Estimating the parameters of a multivariate exponential distribution," *Journal of the American Statistical Association*, vol. 71, no. 354, pp. 465–472, 1976.

[6] N. Balakrishnan, *Handbook of the logistic distribution*. CRC Press, 1991.

[7] A. K. Gupta and S. Nadarajah, *Handbook of beta distribution and its applications*. CRC press, 2004.

[8] E. L. Crow and K. Shimizu, *Lognormal distributions*. Marcel Dekker New York, 1987.

[9] E. W. Stacy, "A generalization of the gamma distribution," *The Annals of mathematical statistics*, pp. 1187–1192, 1962.

[10] H. Rinne, *The Weibull distribution: a handbook*. CRC press, 2008.

[11] L. M. Wein and P. B. Chevalier, "A broader view of the job-shop scheduling problem," *Management science*, vol. 38, no. 7, pp. 1018–1033, 1992.

[12] R. B. Dunham and J. L. Pierce, "Attitudes toward work schedules: construct definition, instrument development, and validation," *Academy of Management Journal*, vol. 29, no. 1, pp. 170–182, 1986.

[13] A. Beckman and M. Sundström, "Development of stationary shoulder for friction stir welding," 2014.

[14] W. E. DEMING, "Using knowledge about variation."

[15] J. M. Juran and A. B. Godfrey, *The quality control process*. McGraw-Hill, 1999.

[16] R. K. Roy, *Design of experiments using the Taguchi approach: 16 steps to product and process improvement*. John Wiley & Sons, 2001.

[17] J. McDonnell, M. Sivapalan, K. Vaché, S. Dunn, G. Grant, R. Haggerty, C. Hinz, R. Hooper, J. Kirchner, M. Roderick *et al.*, "Moving beyond heterogeneity and process complexity: A new vision for watershed hydrology," *Water Resources Research*, vol. 43, no. 7, 2007.

[18] J. Lee, I. Cameron, and M. Hassall, "Improving process safety: What roles for digitalization and industry 4.0?" *Process safety and environmental protection*, vol. 132, pp. 325–339, 2019.

[19] W. A. Shewhart and W. E. Deming, *Statistical method from the viewpoint of quality control*. Courier Corporation, 1986.

[20] J. M. Juran, "Departmental quality planning," *National Productivity Review*, vol. 11, no. 3, pp. 287–300, 1992.

[21] W. Shewhart, "Statistical method from an engineering viewpoint," *Journal of the American Statistical Association*, vol. 26, no. 175, pp. 262–269, 1931.

[22] D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, 2019.

[23] E. L. Grant and R. S. Leavenworth, *Statistical quality control*. McGraw-Hill New York, 1980, vol. 7.

[24] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[25] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[26] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming*. Princeton university press, 2015, vol. 2050.

[27] B. Lantz, *Machine learning with R: expert techniques for predictive modeling*. Packt publishing ltd, 2019.

[28] Z. Chen and B. Liu, *Lifelong machine learning*. Springer, 2018, vol. 1.

[29] E. Rafalski, "Using data mining/data repository methods to identify marketing opportunities in health care," *Journal of Consumer Marketing*, vol. 19, no. 7, pp. 607–613, 2002.

[30] P. H. Patil, S. Thube, B. Ratnaparkhi, and K. Rajeswari, "Analysis of different data mining tools using classification, clustering and association rule mining," *International Journal of Computer Applications*, vol. 93, no. 8, 2014.

[31] L. G. Birta and G. Arbez, *Modelling and simulation*. Springer, 2013.

[32] J. Banks, "Principles of simulation," *Handbook of simulation*, vol. 12, pp. 3–30, 1998.

[33] C. D. Pegden, "An introduction to simio® for beginners," 2009.

[34] N. Boccara and N. Boccara, *Modeling complex systems*. Springer, 2010, vol. 1.

[35] A. Azapagic, "Life cycle assessment and its application to process selection, design and optimisation," *Chemical engineering journal*, vol. 73, no. 1, pp. 1–21, 1999.

[36] N. Hounsell, B. Shrestha, J. Piao, and M. McDonald, "Review of urban traffic management and the impacts of new vehicle technologies," *IET intelligent transport systems*, vol. 3, no. 4, pp. 419–428, 2009.

[37] F. Brauer, C. Castillo-Chavez, Z. Feng *et al.*, *Mathematical models in epidemiology*. Springer, 2019, vol. 32.

[38] J. A. Wiens, D. Stralberg, D. Jongsomjit, C. A. Howell, and M. A. Snyder, "Niches, models, and climate change: assessing the assumptions and uncertainties," *Proceedings of the National Academy of Sciences*, vol. 106, no. supplement_2, pp. 19 729–19 736, 2009.

[39] J. Banks, "Introduction to simulation," in *2000 Winter simulation conference proceedings (Cat. No. 00CH37165)*, vol. 1. IEEE, 2000, pp. 9–16.

[40] P. Mehta, R. Haj-Ahmad, M. Rasekh, M. S. Arshad, A. Smith, S. M. van der Merwe, X. Li, M.-W. Chang, and Z. Ahmad, "Pharmaceutical and biomaterial engineering via electrohydrodynamic atomization technologies," *Drug discovery today*, vol. 22, no. 1, pp. 157–165, 2017.

[41] M. Pearson, T. Monks, A. Gibson, M. Allen, A. Komashie, A. Fordyce, F. Harris-Golesworthy, M. Pitt, S. Brailsford, and K. Stein, "Involving patients and the public in healthcare operational research—the challenges and opportunities," *Operations Research for Health Care*, vol. 2, no. 4, pp. 86–89, 2013.

[42] M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani, "Network state estimation and prediction for real-time traffic management," *Networks and spatial economics*, vol. 1, pp. 293–318, 2001.

[43] A. H. Feinstein and H. M. Cannon, "Constructs of simulation evaluation," *Simulation & Gaming*, vol. 33, no. 4, pp. 425–440, 2002.

[44] D. P. Heyman and M. J. Sobel, *Stochastic models in operations research: stochastic optimization*. Courier Corporation, 2004, vol. 2.

[45] C. Vidalakis, J. E. Tookey, and J. Sommerville, "Demand uncertainty in construction supply chains: a discrete event simulation study," *Journal of the Operational Research Society*, vol. 64, no. 8, pp. 1194–1204, 2013.

[46] C. D. Pegden, "Introduction to simio," in *2008 Winter Simulation Conference*. IEEE, 2008, pp. 229–235.

[47] V. Mabin, "Goldratt's" theory of constraints" thinking processes: A systems methodology linking soft with hard," *History (London)*, 2015.

[48] E. M. Goldratt, *Theory of constraints*. North River Croton-on-Hudson, 1990.

[49] Y. Blount, B. Abedin, S. Vatanasakdakul, and S. Erfani, "Integrating enterprise resource planning (sap) in the accounting curriculum: a systematic literature review and case study," *Accounting Education*, vol. 25, no. 2, pp. 185–202, 2016.

[50] F. Kronthaler and S. Zöllner, *Data Analysis with RStudio*. Springer, 2021.

[51] R. Peng, *Exploratory data analysis with R*. Lulu. com, 2012.

[52] A. Mobarhani, S. Nazari, A. H. Tamjidi, and H. D. Taghirad, "Histogram based frontier exploration," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1128–1133.

[53] M. Pastore, "Overlapping: a r package for estimating overlapping in empirical distributions," *Journal of Open Source Software*, vol. 3, no. 32, p. 1023, 2018.

[54] E. Phadia, "Minimax estimation of a cumulative distribution function," *The Annals of Statistics*, vol. 1, no. 6, pp. 1149–1157, 1973.

[55] M. Santhanagopalan, M. Chetty, C. Foale, S. Aryal, and B. Klein, "Modeling neurocognitive reaction time with gamma distribution," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2018, pp. 1–10.

[56] P. Kontkanen and P. Myllymäki, "Mdl histogram density estimation," in *Artificial intelligence and statistics*. PMLR, 2007, pp. 219–226.

[57] A. Almeida, A. Loy, and H. Hofmann, "ggplot2 compatible quantile-quantile plots in r." *R J.*, vol. 10, no. 2, p. 248, 2018.

[58] P. E. Greenwood and M. S. Nikulin, *A guide to chi-squared testing*. John Wiley & Sons, 1996, vol. 280.

[59] F. W. Scholz and M. A. Stephens, "K-sample anderson–darling tests," *Journal of the American Statistical Association*, vol. 82, no. 399, pp. 918–924, 1987.

[60] V. W. Berger and Y. Zhou, "Kolmogorov–smirnov test: Overview," *Wiley statsref: Statistics reference online*, 2014.

[61] N. M. Razali, Y. B. Wah *et al.*, "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests," *Journal of statistical modeling and analytics*, vol. 2, no. 1, pp. 21–33, 2011.

[62] J. C. Benneyan, "Software review: Stat:: Fit," *OR/MS Today*, vol. 25, no. 1, pp. 38–41, 1998.

[63] H. P. Bloch, *Improving machinery reliability*. Gulf professional publishing, 1998, vol. 1.

[64] A. Yarwood, *Introduction to AutoCAD 2010*. Routledge, 2010.

[65] A. G. Greenwood, P. Pawlewski, and G. Bocewicz, "A conceptual design tool to facilitate simulation model development: object flow diagram," in *2013 Winter Simulations Conference (WSC)*. IEEE, 2013, pp. 1292–1303.

[66] R. G. Sargent, "Verifying and validating simulation models," in *Proceedings of the 28th conference on Winter simulation*, 1996, pp. 55–64.

[67] H. Maylor, "Beyond the gantt chart:: Project management moving on," *European management journal*, vol. 19, no. 1, pp. 92–100, 2001.

[68] T. Melton, "The benefits of lean manufacturing: what lean thinking has to offer the process industries," *Chemical engineering research and design*, vol. 83, no. 6, pp. 662–673, 2005.

[69] E. Esgin and P. Senkul, "A hybrid approach to process mining: Finding immediate successors of a process by using from-to chart," in *2009 International Conference on Machine Learning and Applications*. IEEE, 2009, pp. 664–668.

[70] K. Senderska, A. Mareš, and Š. Václav, "Spaghetti diagram application for workers' movement analysis," *UPB Scientific Bulletin, Series D: Mechanical Engineering*, vol. 79, no. 1, pp. 139–150, 2017.

[71] Z.-J. Lee, S.-F. Su, C.-C. Chuang, and K.-H. Liu, "Genetic algorithm with ant colony optimization (ga-aco) for multiple sequence alignment," *Applied Soft Computing*, vol. 8, no. 1, pp. 55–78, 2008.

[72] J. D. Kelleher, B. Mac Namee, and A. D'arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.

[73] M.-I. Chen and K.-J. Lin, "Dynamic priority ceilings: A concurrency control protocol for real-time systems," *Real-Time Systems*, vol. 2, no. 4, pp. 325–346, 1990.

[74] S. Gholizadeh, "A review of non-destructive testing methods of composite materials," *Procedia structural integrity*, vol. 1, pp. 50–57, 2016.

[75] A. Zolfaghari, A. Zolfaghari, and F. Kolahan, "Reliability and sensitivity of magnetic particle nondestructive testing in detecting the surface cracks of welded components," *Nondestructive Testing and Evaluation*, vol. 33, no. 3, pp. 290–300, 2018.

Appendix A

Validate Pipe Manufacturing Operation

A.1 Throughput

There is a well-known shortage of low-pressure gas pipes, needed for safe and cost-effective operation, in the integrated steel manufacturer. This is evident from a comparison of the number of metres of pipes manufactured by the pipe manufacturing operation, and the demand for pipes required by the company. The required comparison was done from the year 2020 to 2022. The comparison can be seen in Table A.1, where it is observed that in each year instance, the demand for pipes was not met.

Table A.1: Additional Supply of Pipes Needed

	2020	**2021**	**2022**
Total Demand (m)	600	850	900
Total Throughput (m)	196	800	840
Supply Short (m)	404	50	60

The supply shortage of 404 metres experienced in 2020 can be attributed to the halt in manufacturing caused by the COVID-19 pandemic. The increased demand persisted through 2021 and 2022 as a result of the shortfall in the year 2020.

A.2 Cost Comparison

A comparison took place between the pricing per metre of pipe offered by external vendors and the internal pricing per metre of pipe. It is important to note that the pricing evaluated only consists of the labour cost and does not include material cost. The material cost is the same for both parties and can therefore be excluded. Furthermore, any additional cost factors, such as miscellaneous costs, are taken into consideration within the labour cost. The pricing comparison is displayed in Table A.2.

Table A.2: Cost Comparison

Year	**Through-put (m)**	**External Cost/m**	**Cost External**	**Internal Cost/m**	**Cost Internal**	**Saving /m**	**Total Saving**
2020	196	R 2,000	R 392,000	R 1,663	R 325,948	R 337	R 66,052
2021	800	R 2,812	R 2,249,600	R 1,220	R 976,000	R 1,592	R 1,273,600
2022	840	R 3,084	R 2,590,560	R 1,973	R 1,657,320	R 1,111	R 933,240
Average	612	R 2,632	R 1,744,053	R 1,619	R 986,423	R 1,013	R 757,631

The provided table outlines relevant data spanning from 2020 to 2022, detailing the number of manufactured pipes in metres, along with both external and internal costs per metre. Additionally, the external and internal costs are computed by multiplying the throughput with the corresponding cost per metre. The cost saving per metre is ascertained by calculating the difference between the external cost per metre and the internal cost per metre. A consistent trend emerges where, in each annual instance, external vendors prove to be more expensive than internal production. To quantify the benefits, the

overall saving from internal production is determined by multiplying the calculated cost saving per metre by the annual pipe production quantity.

As a result, the calculated average saving represents the clear financial benefit of manufacturing pipes internally. The data states that by opting for internal pipe manufacturing, an average annual saving of approximately R 757 631 was achieved.

Appendix B

Mechanical Fabrication Job Shop

B.1 Floor Layouts

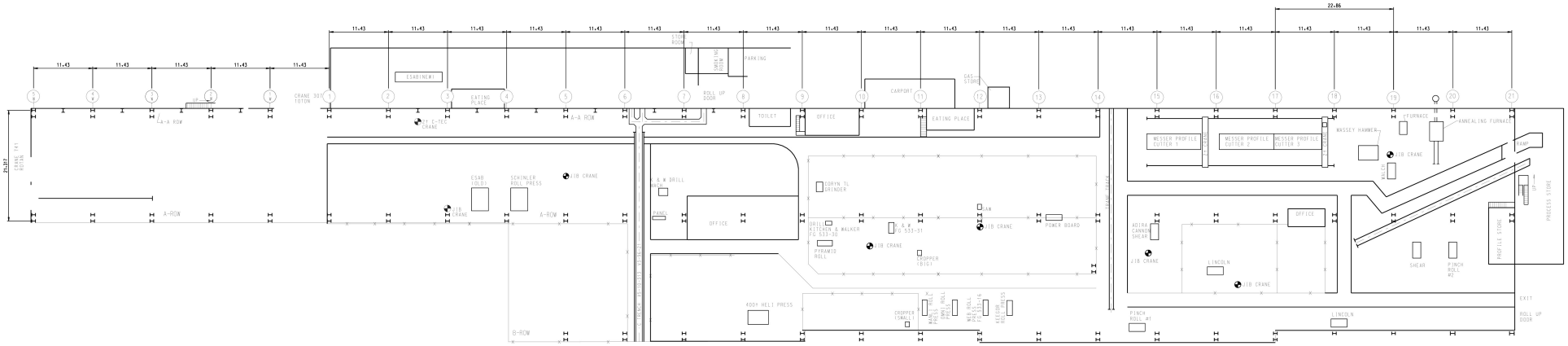


Figure B.1: Mechanical Fabrication Floor Layout - with Relevant Dimensions

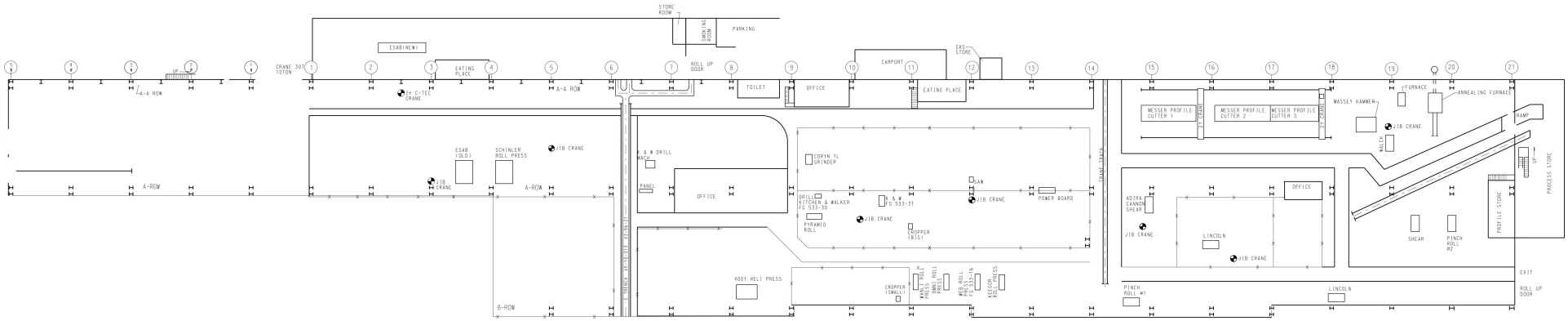


Figure B.2: Mechanical Fabrication Floor Layout

B.2 Sections in Job Shop

To adequately provide an overview regarding the proceedings and operations within the job shop, a specific number is assigned to each area, as can be seen in Figure B.3.

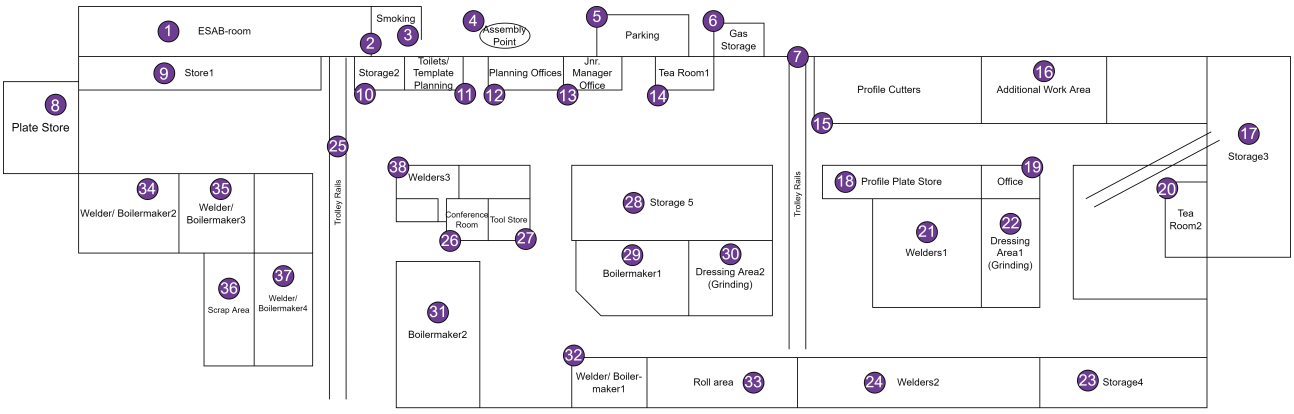


Figure B.3: Schematic Floor Layout of Mechanical Fabrication Job Shop with Area Numbering

These areas are briefly defined as follows:

1. This area is identified as the “ESAB-room” because of the machine that is present within this area. Further elaboration regarding the operations that are conducted within this room is discussed in Section 2.2.3.
2. An opening that enables access from and to the “ESAB-room”.
3. An area that is identified where employees can smoke.
4. A designated area where employees must assemble during a safety emergency.
5. Parking area where employees can park.
6. Storage area for all gas-related cylinders.
7. The main entrance to MechFab job shop.
8. Designated area where steel plates are stored to be used for different jobs.
9. Area where low-utilised equipment is stored.
10. Another area where equipment is stored that is not utilised often.
11. This area has a second floor. The ground floor has ablution facilities, whereas the upper floor is a drafting office.
12. Offices where the main activity is to plan upcoming jobs.
13. The office of the Manager of the job shop.
14. An area where employees go to have tea breaks or have lunch.
15. Profile-cutting machines are located in this area.
16. Additional working space to do various jobs.
17. This area is used to store manufacturing equipment and material.
18. A store that is used to store steel plates that are cut and/or are already cut by profile cutters at work area 15.

19. Office space.

20. A designated room where employees take tea breaks and have lunch.

21. Work area that are used for welding.

22. Area that is used by “dresser”. The term “dresser” refers to an employee that works with the steel to perform tasks such as smoothing, grinding, and deburring the surface of the steel to achieve a desired finish.

23. Storage area for underutilised equipment.

24. Designated workspace for employees specifically assigned to welding tasks.

25. Rails on which a trolley runs to transfer material from and to the various bays. Elaboration regarding this transport method is to follow.

26. A conference room where meetings are held.

27. Store where tools are stored.

28. Space that is utilised to store equipment and/ or materials that are not used regularly.

29. Workspace that is utilised by boilermakers to attend to tasks.

30. Area that is used by “dressers”.

31. Work area specifically designated for boilermakers to carry out tasks.

32. This area is assigned to boilermakers and/or welders to attend to their jobs.

33. Space utilised for rolling plates of various sizes.

34. This designated area is assigned to boilermakers and/or welders, enabling the employees to efficiently carry out allocated tasks.

35. Boilermakers and/or welders attend to jobs or tasks in this area.

36. Area that is utilised for discarded material, mainly off-cuts of steel.

37. This area is designated for the assignment of boilermakers and/or welders to carry out specific jobs or tasks.

38. Designated area assigned to welders to attend to tasks.

B.3 Machine Allocations

A brief description regarding the types of machines is provided and elaborated on in Table B.1.

Table B.1: Brief Discussion of Machine Types in Job Shop

Machine Type	Symbol	Description
Roll		The **roll** machines within the job shop are mainly used for bending or shaping sheets, plates, or pipes into desired forms or curves. The machine applies pressure or force to the material, causing it to deform and take on the desired shape.
Profile cutters		The **profile cutters** employ flame-cutting techniques to precisely shape or sever intricate profile designs on steel sheets.
Guillotine		The **guillotine** machines are used for the straight cutting of various steel sheets. The machine employs a sharp, vertical blade that descends in a vertical motion to shear the material, enabling precise and efficient cutting operations.
Press		The **press** machines are used for shaping or forming steel materials through the application of force. The machine utilizes hydraulic or mechanical pressure enabling the production of various steel components and parts with desired shapes and specifications.
Crop shear		A crop **shear machine**, a type of shearing machine, is primarily employed to cut steel sheets to obtain the desired width, ensuring the desired size and dimensions are achieved.
ESAB welder		The ESAB-machines refer to a range of equipment manufactured by ESAB, a well-known company specializing in welding and cutting technologies. The machines are utilized for welding operations within the job shop.
Drill		The **drill** machines are used to create precise holes in the steel materials. The machine employs rotating cutting tools, such as drill bits, to remove material and enable various operations. Some of these operations include machining, assembly, or fastening during the manufacturing of jobs.
Lincoln		**Lincoln** profile cutting machine designed to precisely cut materials.

B.4 Machine Visualisation

(a) Roll

(b) Profile Cutter

(c) Guillotine

(d) Press

(e) Crop Shear

(f) ESAB Welder

(g) Drill

(h) Lincoln Profile Cutter

Figure B.4: Visual Demonstration of Machine Types in MechFab

B.5 Workforce Allocations

Information regarding the personnel's designations is provided in Table B.2. This information is necessary for gaining a clearer understanding of the operational aspects and workforce composition within the job shop.

Table B.2: Information Regarding Workforce

Number of Employees	Designation	Related Icon	Description
7	Boilermaker	B	The **boilermakers** are trades personnel that fabricates steel materials. The personnel are responsible for assembling, installing, maintaining, and attending to various jobs within the MechFab job shop.
9	Welder	W	Within a job shop setting, **welders** play an important role as highly skilled tradespeople who employ diverse welding techniques to effectively join metal components. Their primary responsibility encompasses fabricating, repairing, and maintaining structures, parts, and equipment to guarantee optimal strength and integrity.
4	Operator dresser	D	An **operator dresser** in the job shop environment sets up and operates dressing machines, namely grinders, to shape and finish steel materials. Dressers contribute to the preparation of steel components for further processing and achieving the desired appearance and dimensions.
2	Profile operator	P	A **profile operator** in the job shop environment operates the profile-cutting machines to accurately cut and shape metal workpieces according to specific designs or profiles.

Appendix C

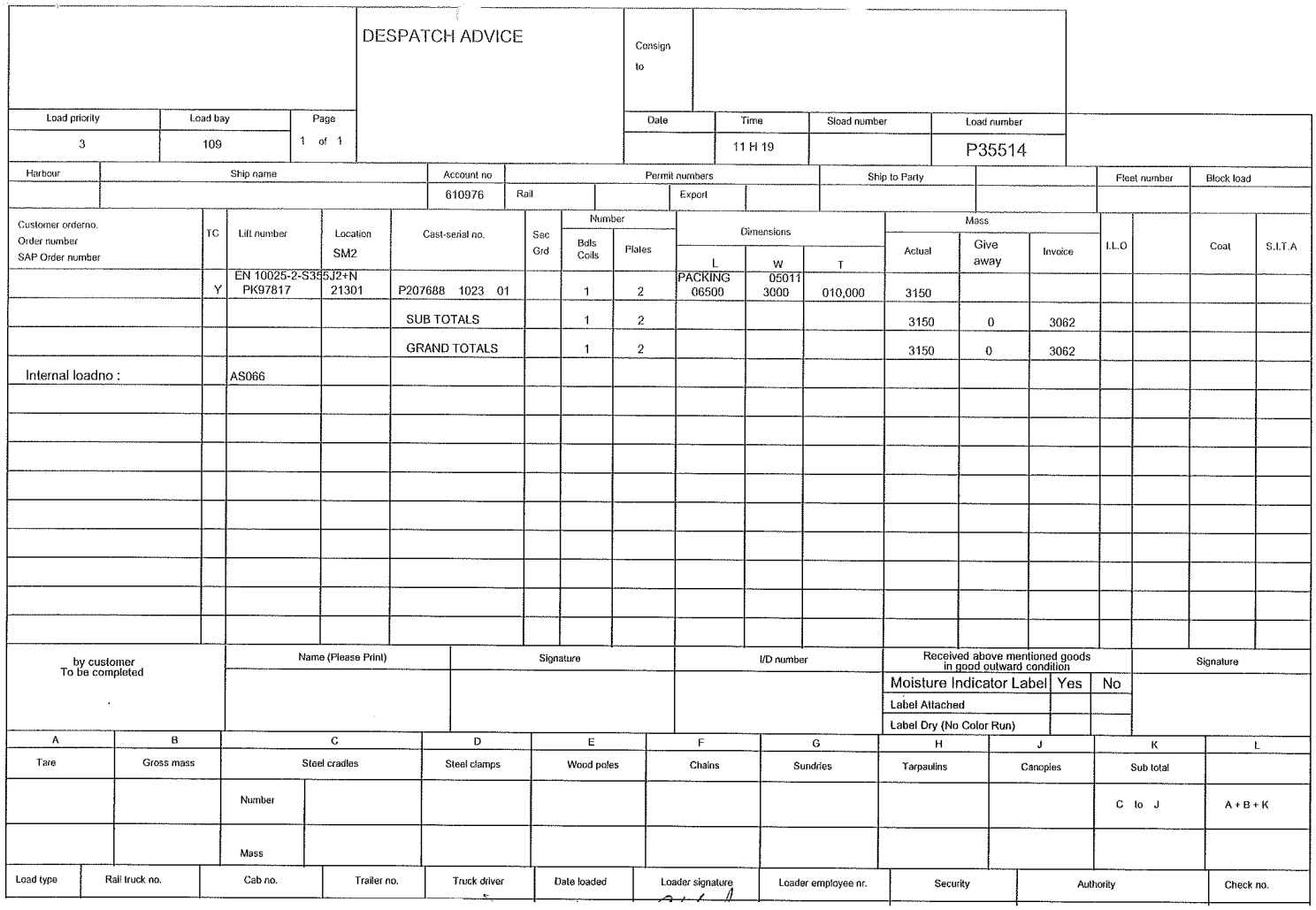
Plate Specifications

The structural steel grade of the plates used is S355J2 EN 10025. This steel grade has a minimum yield strength of $355 N/mm^2$. High yield and tensile strength are present and the steel is also supplied with a variety of treatments to ensure that it is highly usable in a range of projects.

The chemical analysis for the plates used can be seen in Table C.1

Table C.1: Chemical Elements of Plates

Element	Min/Max	Percentage (%)
C (Carbon)	Max	0.2000
Mn (Manganese)	Max	1.6000
P (Phosphorus)	Max	0.0250
S (Sulfur)	Max	0.0250
Si (Silicon)	Max	0.5500
Cu (Copper)	Max	0.5500
Al (Aluminium)	Min	0.0200



DESPATCH ADVICE

Consign to

Load priority	Load bay	Page	Date	Time	Sload number	Load number
3	109	1 of 1		11 H 19		P35514

Harbour	Ship name	Account no	Permit numbers				Ship to Party		Fleet number	Block load
		610976	Rail		Export					

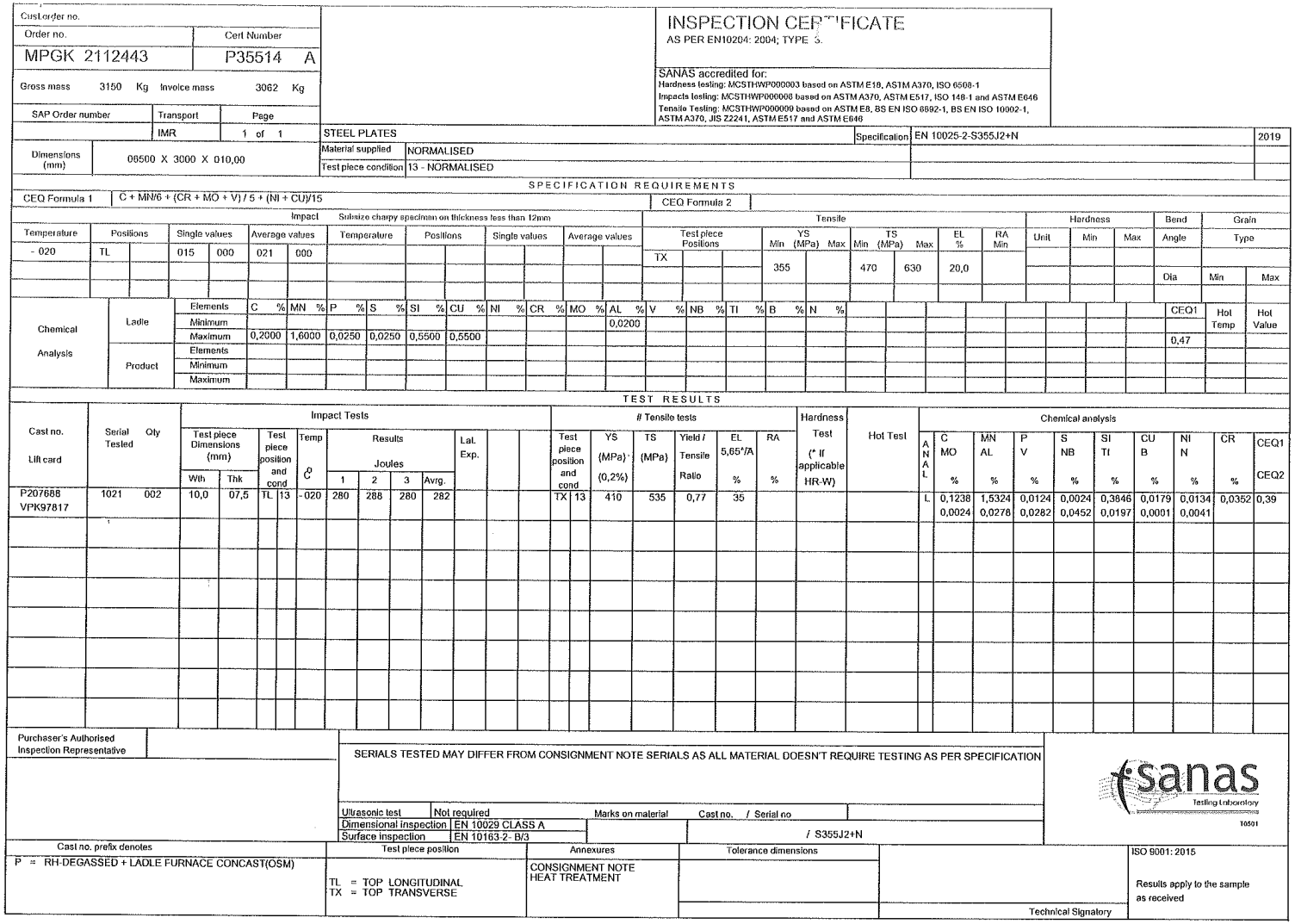
Customer orderno. / Order number / SAP Order number	TC	Lift number	Location	Cast-serial no.	Sec Grd	Number Bdls Coils	Number Plates	Dimensions L	Dimensions W	Dimensions T	Mass Actual	Mass Give away	Mass Invoice	I.L.O		Coat	S.I.T.A
	Y	EN 10025-2-S355J2+N PK97817	SM2 21301	P207688 1023 01		1	2	PACKING 06500	05011 3000	010,000	3150						
				SUB TOTALS		1	2				3150	0	3062				
				GRAND TOTALS		1	2				3150	0	3062				
Internal loadno :		AS066															

by customer To be completed	Name (Please Print)	Signature	I/D number	Received above mentioned goods in good outward condition			Signature
				Moisture Indicator Label	Yes	No	
				Label Attached			
				Label Dry (No Color Run)			

A	B	C		D	E	F	G	H	J	K	L
Tare	Gross mass	Steel cradles		Steel clamps	Wood poles	Chains	Sundries	Tarpaulins	Canopies	Sub total	
		Number								C to J	A + B + K
		Mass									

Load type	Rail truck no.	Cab no.	Trailer no.	Truck driver	Date loaded	Loader signature	Loader employee nr.	Security	Authority	Check no.

Figure C.1: Material Certificate (1)



Cust.order no.

Order no. MPGK 2112443 | Cert Number P35514 A

Gross mass 3150 Kg | Invoice mass 3062 Kg

SAP Order number | Transport IMR | Page 1 of 1

Dimensions (mm): 06500 X 3000 X 010,00

INSPECTION CERTIFICATE

AS PER EN10204: 2004; TYPE 3.

SANAS accredited for:
Hardness testing: MCSTHWP000003 based on ASTM E18, ASTM A370, ISO 6508-1
Impacts testing: MCSTHWP000008 based on ASTM A370, ASTM E517, ISO 148-1 and ASTM E646
Tensile Testing: MCSTHWP000009 based on ASTM E8, BS EN ISO 6892-1, BS EN ISO 10002-1, ASTM A370, JIS Z2241, ASTM E517 and ASTM E646

STEEL PLATES | Specification: EN 10025-2-S355J2+N | 2019

Material supplied: NORMALISED

Test piece condition: 13 - NORMALISED

SPECIFICATION REQUIREMENTS

CEQ Formula 1: C + MN/6 + (CR + MO + V) / 5 + (NI + CU)/15

CEQ Formula 2:

Impact — Subsize charpy specimen on thickness less than 12mm

Temperature	Positions	Single values		Average values	
- 020	TL	015	000	021	000

Temperature	Positions	Single values		Average values	

Tensile

Test piece Positions	YS Min (MPa)	YS Max	TS Min (MPa)	TS Max	EL %	RA Min
TX	355		470	630	20,0	

Hardness

Unit	Min	Max

Bend: Angle | Dia | Min | Max

Grain: Type

Chemical Analysis

		Elements	C %	MN %	P %	S %	SI %	CU %	NI %	CR %	MO %	AL %	V %	NB %	TI %	B %	N %	CEQ1	Hot Temp	Hot Value
Chemical Analysis	Ladle	Minimum										0,0200								
		Maximum	0,2000	1,6000	0,0250	0,0250	0,5500	0,5500										0,47		
	Product	Elements																		
		Minimum																		
		Maximum																		

TEST RESULTS

Cast no. / Lift card	Serial Qty Tested	Impact Tests: Test piece Dimensions (mm) Wth	Thk	Test piece position and cond	Temp °C	Results Joules 1	2	3	Avrg.	Lat. Exp.	# Tensile tests: Test piece position and cond	YS (MPa) (0,2%)	TS (MPa)	Yield / Tensile Ratio	EL 5,65*√A %	RA %	Hardness Test (* If applicable HR-W)	Hot Test
P207688 VPK97817	1021 002	10,0	07,5	TL 13	- 020	280	288	280	282		TX 13	410	535	0,77	35			

Chemical analysis	C MO %	MN AL %	P V %	S NB %	SI TI %	CU B %	NI N %	CR %	CEQ1 CEQ2
ANAL L	0,1238	1,5324	0,0124	0,0024	0,3646	0,0179	0,0134	0,0352	0,39
	0,0024	0,0278	0,0282	0,0452	0,0197	0,0001	0,0041		

Purchaser's Authorised Inspection Representative

SERIALS TESTED MAY DIFFER FROM CONSIGNMENT NOTE SERIALS AS ALL MATERIAL DOESN'T REQUIRE TESTING AS PER SPECIFICATION

Ultrasonic test	Not required	Marks on material	Cast no. / Serial no
Dimensional inspection	EN 10029 CLASS A		/ S355J2+N
Surface inspection	EN 10163-2- B/3		

sanas Testing Laboratory T0501

Cast no. prefix denotes: P = RH-DEGASSED + LADLE FURNACE CONCAST(OSM)

Test piece position: TL = TOP LONGITUDINAL; TX = TOP TRANSVERSE

Annexures: CONSIGNMENT NOTE; HEAT TREATMENT

Tolerance dimensions

Technical Signatory

ISO 9001: 2015

Results apply to the sample as received

Figure C.2: Material Certificate (2)

Appendix D

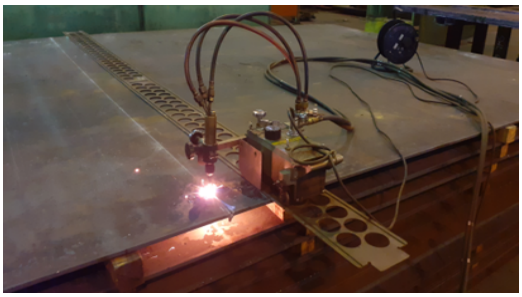
Details Regarding the Pipe Manufacturing Process

D.1 Straight Line Cutting

Before an explanation is provided regarding the operations at the *s_line* workstation, it will be beneficial to provide an overview of the equipment that is used to obtain the workstation's objective. First, it is required that the plate is measured and marked accordingly. Therefore, *measuring and marking* tools are used to measure and mark the plates accordingly, ensuring accurate cuts that match the required specifications. The tools consist of one 10 m measuring tape, one right-angled triangle, and chalk that enables the employee to make the required marks on the plates where the cuts are to take place. The tools used are displayed in Figure D.1.

Figure D.1: Measuring and Marking Tools for *s_line* Workstation

After the required marks are made on the plate, a *motor-driven portable flame cutter* is used to cut to retrieve two plates. The machine uses a combination of heat, oxygen, and a motorised driven system to melt and remove the steel in a controlled manner. A visual demonstration of the flame cutter is provided in Figure D.2, which is also a representation of the operations that take place at the *s_line* workstation.



(a) Representation 1 of *s_line* workstation

(b) Representation 2 of *s_line* workstation

Figure D.2: Representation of *s_line* workstation

The following main steps are followed to achieve the workstation's objective:

1. Mark the plate: The cutting line is first marked on the plate with the use of the adequate measuring tools.
2. Setting up the equipment: After marking the line, the gas cutter is prepared and adjusted according to the plate's thickness and material. This may consist of altering the voltage, amperage, and gas and oxygen pressure.
3. Cutting the plate: The gas cutter is placed along the cutting line, and the operator starts the cut. The cutting line is followed by the torch as it moves along, cutting the plate according to the correct dimensions.

D.2 Deburring of Plates

The main reasons for grinding the plates' edges include the following:

1. Burr removal: Burrs, or rough edges or small fragments of leftover steel from the straight line cutting process, may be present on the edges of the plates. These burrs, which can obstruct welding operations and harm the interior of the pipe, are removed by grinding the plates' edges.
2. Smooth edges: By grinding the plates' edges, a smooth surface is obtained. This ensures that the pipes' edges fit together uniformly and firmly. The pipe's structural integrity and ability to resist the necessary stress and pressure are dependent on ensuring that the edges are ground accordingly.
3. Enhancing welding quality: By providing a clean surface for welding, the smooth edges achieved by grinding can enhance the quality of the welds. This decreases the possibility of weld flaws like cracks or porosity, which can negatively impact the pipe's structural integrity.
4. Assisting the welding process: By ensuring a bevelled edge, which enables a deeper penetration of the weld and enhances its strength and quality, assist in facilitating the welding process.

For demonstration purposes, a visual demonstration of the *dress* workstation's operations is depicted in Figure D.3.

(a) Representation 1 of *dress* workstation

(b) Representation 2 of *dress* workstation

Figure D.3: Representation of *dress* workstation

D.3 Roll

A visual demonstration of the machine used at the *roll* workstation is provided in Figure D.4.

(a) Representation 1 of *roll* workstation

(b) Representation 2 of *roll* workstation

Figure D.4: Representation of *roll* workstation

The process of rolling a steel sheet to form a cylindrical shape, known as a 'canister', includes the following main steps:

1. Setting up the roll press machine: Ensure that all equipment, and the machine itself, are functional and in working condition.

2. Feeding the sheet into the roll press machine: The steel sheet is progressively fed through the rollers as it is fed into the roll press machine from one end. The sheet is subjected to pressure from the rollers, which gradually bends the sheet into the desired cylindrical shape. The crane is used to assist to transfer the steel sheet into position.

3. Adjusting the rollers: The rollers' settings are adjusted during the process to ensure that the canister is formed according to the correct shape and size. This could entail altering the roller dies or the shape between the rollers. The operator uses a pattern to ensure that the adequate shape and size are retrieved. This pattern is placed next to the inside seam of the canister to analyse the shape. The roll press is further used to alter the canister to adhere to the pattern's dimensions and consequently the specifications of the desired canister's dimensions. Please refer to Figure D.5 for a visual demonstration of the use of the pattern.

Figure D.5: Assessment of Canister's Shape

4. Welding the seams: After the canister is formed, the edges are joined together with the use of tack welding. Before the rest of the seam is tack welded, it is necessary to ensure that the two sides of the plates align correctly. To adhere to this objective, the *dog-and-wedge* tools are utilised. During the manufacturing of the pipes, the *dog-and-wedge* are two separate tools used collectively to ensure that two corresponding plates/ canisters are lined up accordingly. If welding onto the work-piece is deemed permissible, the *dog-and-wedge* method is used to align and secure the parts in place for tack-welding the parts together, which serves as preparation for welding operations. The operation of the *dog-and-wedge* tools is further briefly explained in Figure D.6 and visualised in Figure D.7.

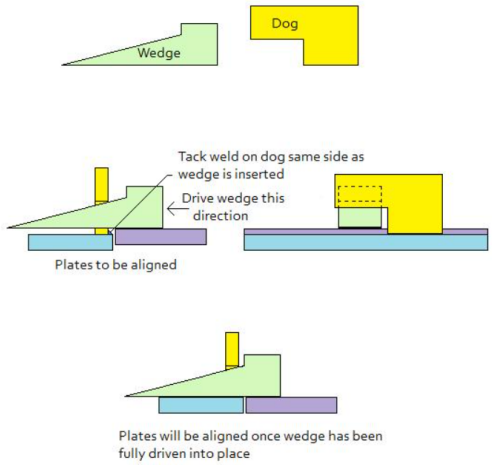


Figure D.6: Brief demonstration Regarding the Use of *dog and wedge* [2]

Figure D.7: Visual Demonstration of the *dog and wedge* Tools Being Used

5. Removing the canister: After the canister is rolled and the seam is tack welded, the canister is removed from the machine. The roller-die lifts at an approximate angle of 45 degrees which makes it possible for the operator to remove the canister with the use of the overhead crane.

D.4 Long-weld

For illustration purposes, refer to Figure D.8, which provides a visual demonstration of the ESAB welding machine.

(a) Representation 1 of *l_weld* workstation

(b) Representation 2 of *l_weld* workstation

Figure D.8: Representation of *l_weld* workstation

The ESAB#1 welding machine is utilised to weld the seam of the canister. The main steps of this process is as follow:

1. Preparing the canister: This phase may involve ensuring that the surface is cleaned and/or prepared where the welding takes place. Furthermore, any rough or uneven edges may be deburred or smoothed.

2. Setting up the ESAB#1 welding machine: The next step is to ensure that the ESAB#1 welding machine is set up accordingly. To ensure this, the correct welding parameters must be identified based on the thickness and kind of quality of the material being used. Such parameters include the current, voltage, and wire feed speed.

3. Positioning the canister: The canister is positioned in a welding fixture to keep the canister in position throughout the welding procedure.

4. Starting the weld: The welding process starts by positioning the ESAB#1 welding gun at the start of the weld joint on the inside of the canister. After the operator ensured that all the equipment is in order, the operator starts the machine. The welding gun moves along the joint, welding the two sides together.

5. Welding the entire joint: To achieve a uniform weld bead, the weld is carried over the full length of the joint by moving the welding gun at a constant pace. Typically, the weld is done in many passes, with each pass covering a smaller piece of the connection until the whole joint is welded.

6. Post-welding steps: The canister is analysed and inspected for any defects or anomalies once the welding is complete. Any rough edges or abnormalities are cleaned and smoothed out as needed, typically with the use of a grinder.

The procedure of welding the seam takes place on the inside and outside of the canister's seam. This ensures a strong and secure bond, enhancing structural integrity and preventing any potential leaks or openings.

D.5 Inspection

The inspection technique used is NDT, which is a non-invasive evaluation and analysis method used by industries to assess the properties of materials, components, structures, or systems for distinctive variations [74]. The technique is also used to detect welding defects and discontinuities, all without inflicting damage to the original part [74].

The NDT method used during the inspection of the pipes includes Magnetic Particle Testing (MPT). As explained by Zolfaghari et.al [75], MPT is a method used to identify and detect surface and near-surface defects in ferromagnetic materials. Iron oxide particles are added to the surface of the material and subjected to a magnetic field. The particles accumulate in the areas where the magnetic field is distorted, indicating that a defect is present [75].

Appendix E

Transportation Methods

Table E.1: Information Regarding Transportation Methods

Transportation Mode	**Visual Illustration**	**Description**
Overhead crane(s)		Overhead crane(s) are utilized in the various bays, mainly transferring plates, canisters, 9-metre and 12-metre pipes from and to the various workstations.
Trolley		Trolley are used to transfer plates from Bay 2 to Bay 1.
Vehicle		The vehicle is utilized to: - Transfer canisters from and to the *roll* workstation - Transfer canisters from *l_weld* workstation to Bay 2, where the overhead crane 741 is to transfer canisters further to *assem* workstation.
Roll canister on floor		After the canisters are re-rolled, the canisters are rolled on the floor to Bay 2, where the overhead crane 741 is utilized to transfer the canisters to assem workstation.
Lowbed		Utilised to transfer 12-metre pipes to customers.

Appendix F

Probability Distribution Functions

Table F.1: Distribution Information (1) [3] [4]

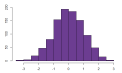
Distribution	Properties	Parameters	Illustration
Normal	- Symmetry around the mean. - Unimodal with a single peak at the mean. - Empirical Rule: **68-95-99.7 rule** - about 68% within 1 SD, 95% within 2 SD, and 99.7% within 3 SD. - **Central limit theorem** - sum or average of many variables approaches normal.	Normal(μ, σ) - μ: mean - σ: standard deviation	
Uniform	- Constant Probability in the range $[a, b]$. - **Flat PDF** - all outcomes equally likely. - Mean: $\frac{a+b}{2}$, Variance: $\frac{(b-a)^2}{12}$.	Uniform(a, b) - a: Lower bound - b: Upper bound	

Table F.2: Distribution Information (2) [5] [6] [7]

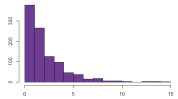
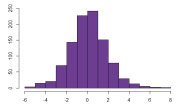
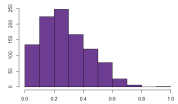
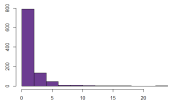
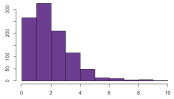
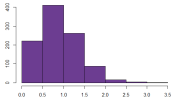
Distribution	Properties	Parameters	Illustration
Exponential	- Memorylessness: the probability of an event happening within a specific time interval remains constant and does not rely on the amount of time already passed. This characteristic is often referred to as "lack of aging," indicating that past events do not influence the likelihood of future occurrences. - Exponential Decay: As time passes, there is a decreasingly smaller chance of waiting for the next occurrence at a specific time. This attribute is useful when modelling processes with consistent failure rates. - Mean and Variance: The mean (μ) of the exponential distribution is $\frac{1}{\lambda}$, and the variance (σ^2) is $\frac{1}{\lambda^2}$.	Exponential(λ) - λ: rate	
Logistic	- Symmetry and S-shaped Curve: The distribution has an S-shaped curve and is symmetric about its mean, with the tails asymptotically moving toward zero. - Mean, variance, and moments: The mean is identified as μ, and its variance is $\frac{\pi^2 s^2}{3}$. The distribution has higher moments, allowing for the calculation of skewness and kurtosis. - Relationship to the Standard Logistic Distribution: The standard logistic distribution, where $\mu = 0$ and $s = 1$ is a specific instance of the logistic distribution.	Logistic(μ, s) - μ: location - s: scale	
Beta	- Bounded support: Since the beta distribution is based on the range [0,1], the distribution may be used to simulate random variables with constrained changes. - Flexibility: The distribution can adopt a variety of forms, including symmetric, skewed, and U-shaped, by adjusting the shape parameters. - Uniform and Symmetric Case: The special case of the beta distribution where $\alpha = \beta = 1$ corresponds to the uniform distribution on the interval $[0, 1]$.	Beta(α, β) - α and β: shape parameters	

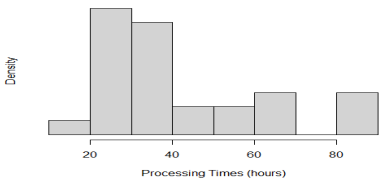
Table F.3: Distribution Information (3) [8] [9] [10]

Distribution	Properties	Parameters	Illustration
Lognormal	- Positivity: Defined only for positive values of x, making the distribution suitable for modelling variables with lower bounds at zero. - Asymmetry: The distribution's relationship to the normal distribution causes the distribution to be favourably skewed. Depending on the value of μ and σ , a broad range of skewness is permitted. - Moments: When compared to other distributions, the lognormal distribution's moments are more difficult to compute, which can have an influence on modelling and statistical analysis.	Lognormal(μ, σ) - μ: location - σ: scale	
Gamma	- Flexibility: The gamma distribution can efficiently mimic a variety of real-world occurrences because of the shape parameter k, which allows for a broad range of skewness and forms. - Special Cases: The gamma distribution reduces to the exponential distribution when k is equal to 1, however, it corresponds with the Erlang distribution for integer values of k. - Moments: It is relatively simple to compute the moments of the gamma distribution, which helps with parameter estimation and statistical inference.	Gamma(k, θ) - k: shape - θ: scale	
Weibull	- Flexibility: The distribution can simulate increasing and decreasing failure rates because the shape parameter k allows for a wide range of failure characteristics. - Special Cases: The Weibull distribution simplifies to the exponential distribution when k is equal to 1, which represents continuous failure rates throughout time. The distribution represents rising failure rates for values of $k > 1$, and decreasing failure rates for values of $k < 1$. - Hazard Function: The Weibull PDF may be utilised to generate the hazard function.	Weibull(λ, k) - λ: scale - k: shape	

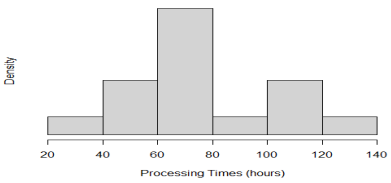
Appendix G

Data Analysis

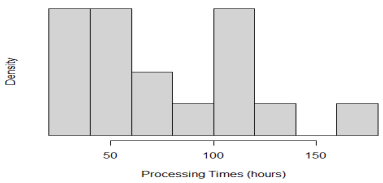
G.1 Exploratory Data Analysis Plots



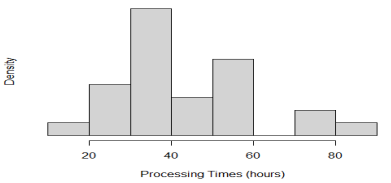
(a) Dress



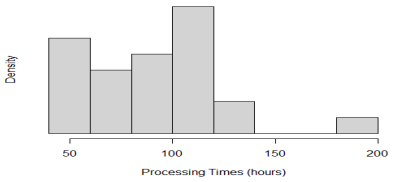
(b) Roll



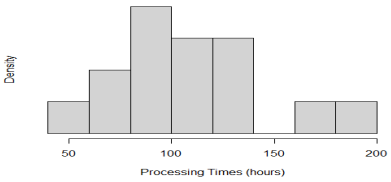
(c) Long weld



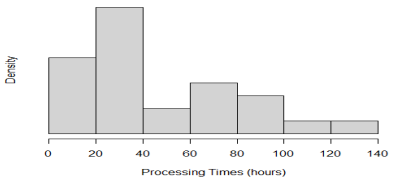
(d) Re-roll



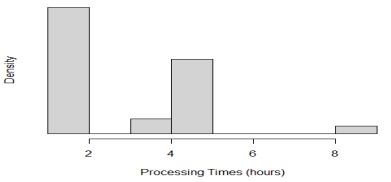
(e) Assemble canisters



(f) Circular weld canisters into 12-metre pipes

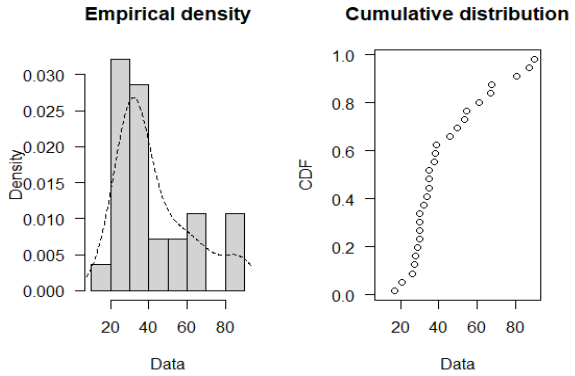


(g) Cosmetics done on pipes

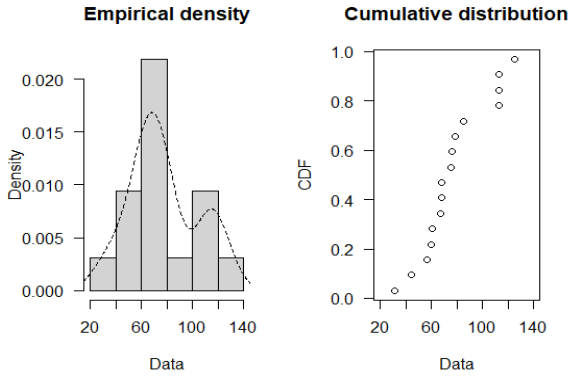


(h) Dispatch to customer

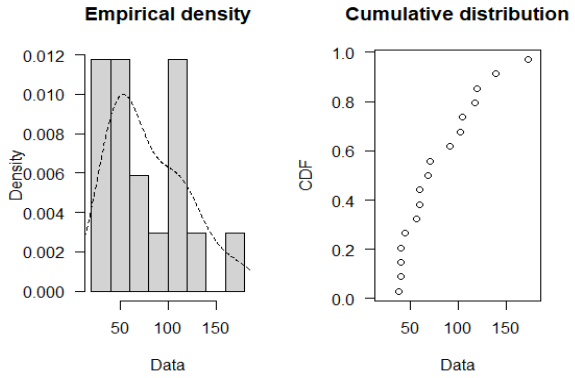
Figure G.1: Histograms of processing times' data



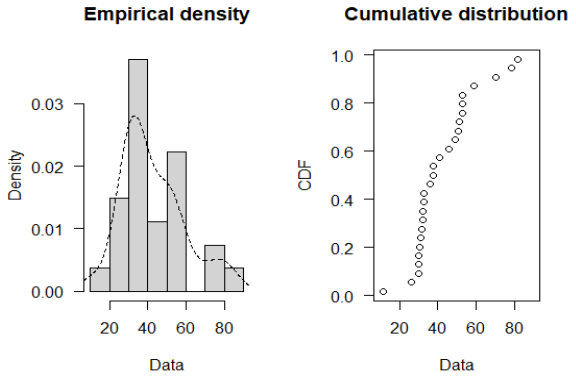
(a) Dress



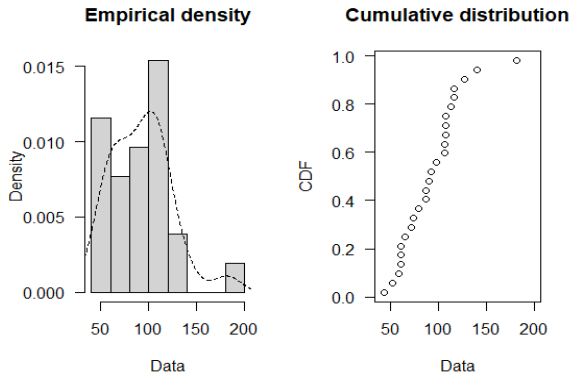
(b) Dress



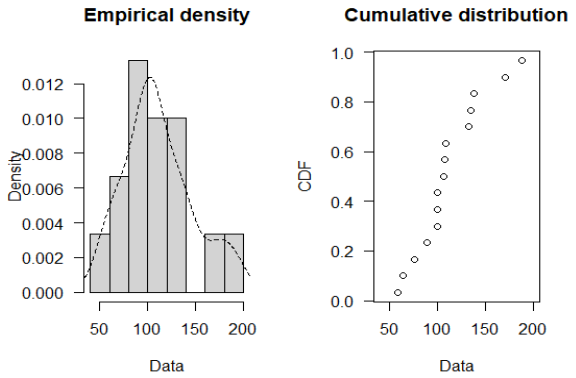
(c) Dress



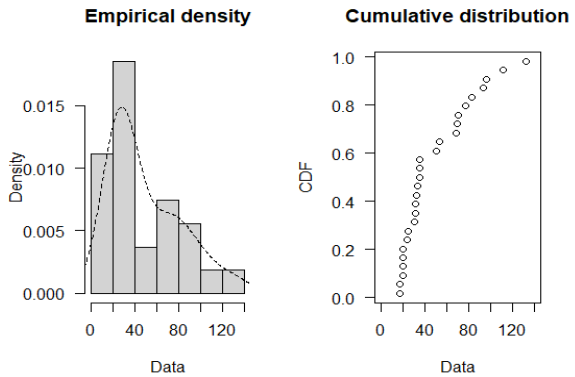
(d) Dress



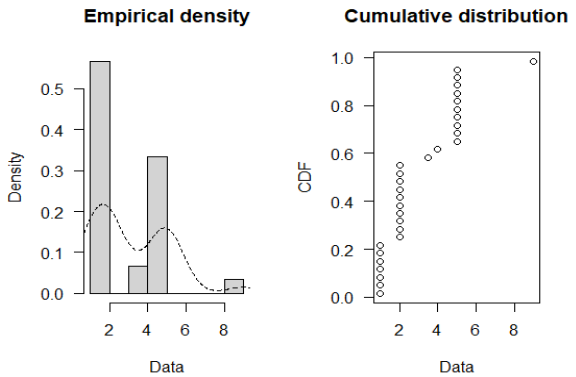
(e) Dress



(f) Dress

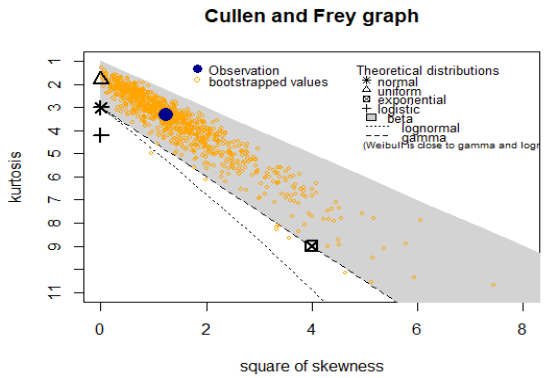


(g) Dress

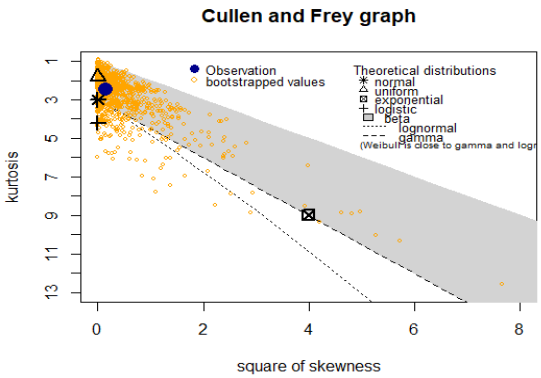


(h) Dress

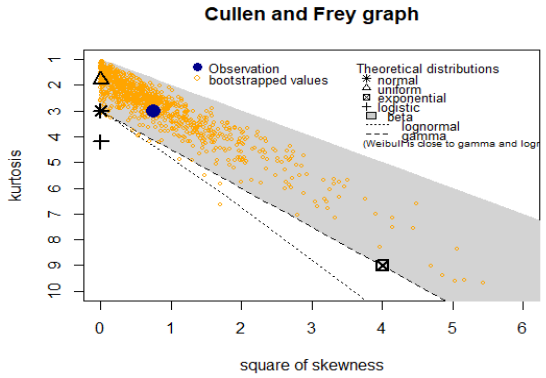
Figure G.2: Empirical Density and Cumulative Distribution of Operations' Process Times



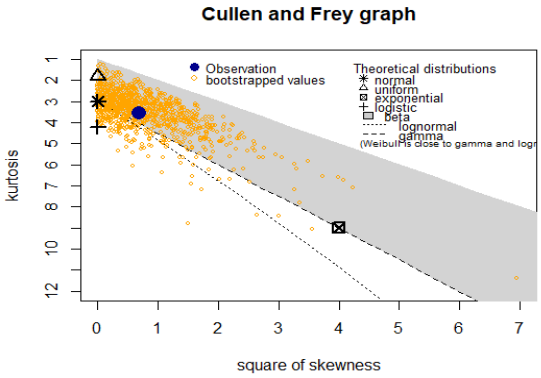
(a) Dress



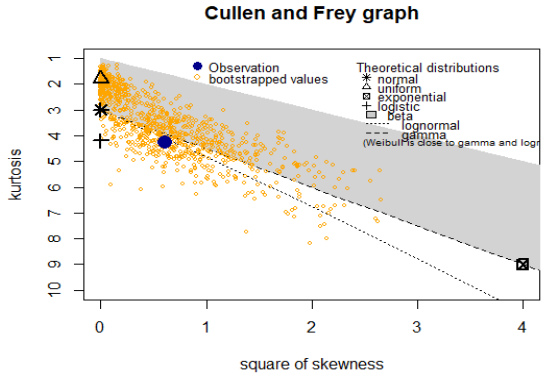
(b) Roll



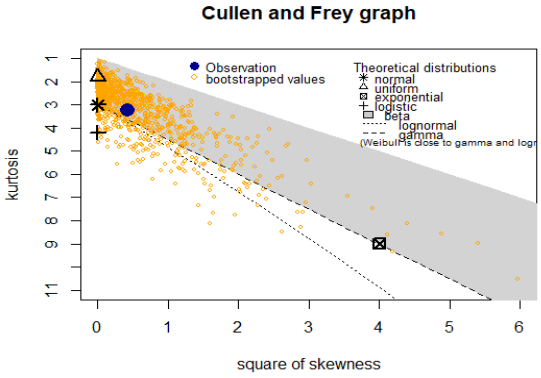
(c) Long weld



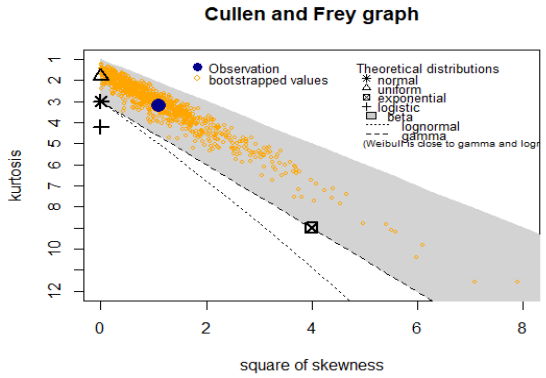
(d) Re-roll



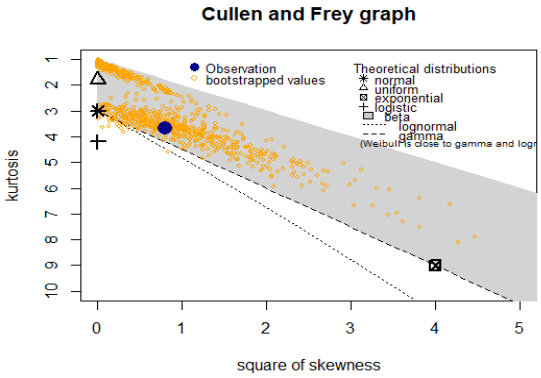
(e) Assemble of canisters



(f) Circular weld

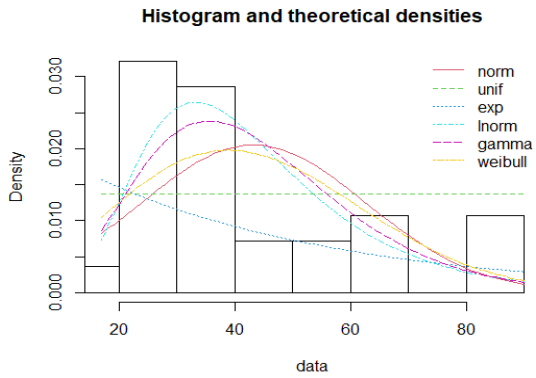


(g) Remove splatter

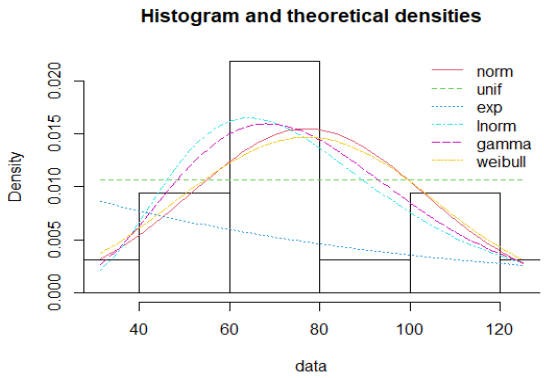


(h) Dispatch 12 metre pipe

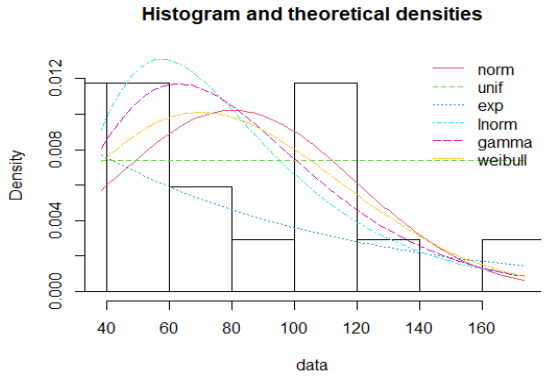
Figure G.3: Cullen and Frey Graphs of processing times' data



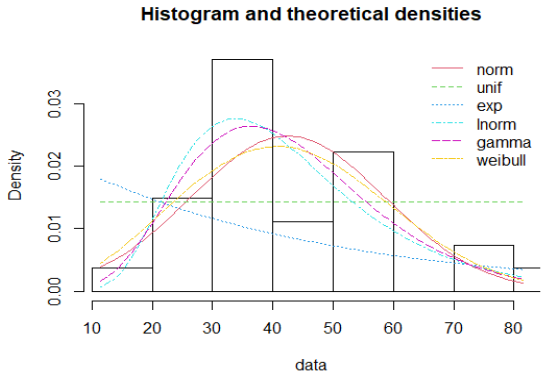
(a) Dress



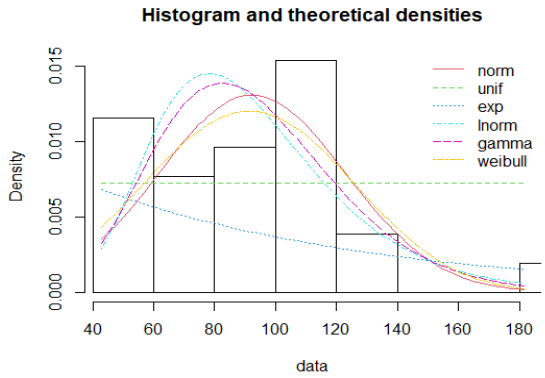
(b) Roll



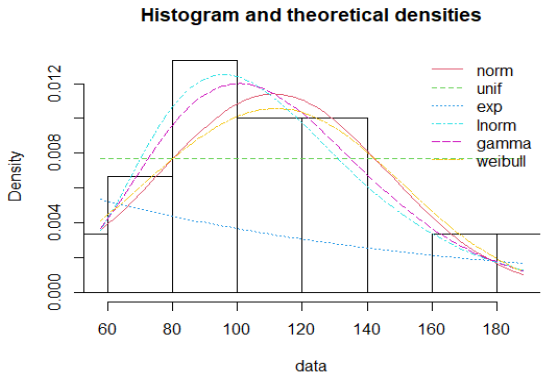
(c) Long weld of canisters



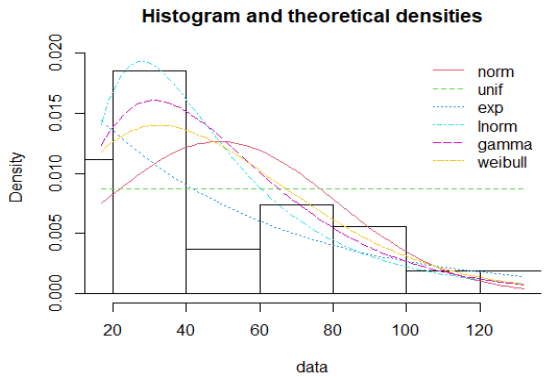
(d) Re-roll canisters



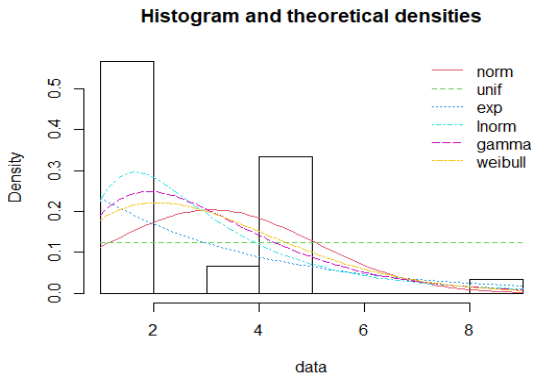
(e) Assemble canisters



(f) Circular weld 12 metre pipe

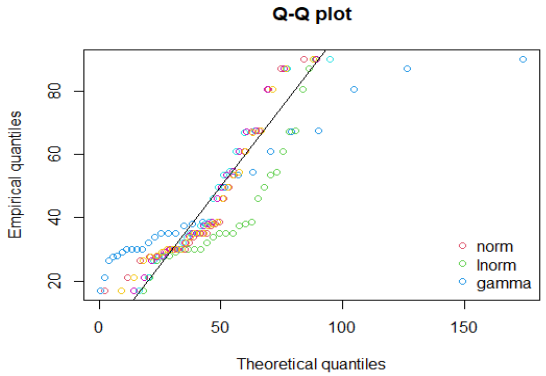


(g) Remove splatter from pipes

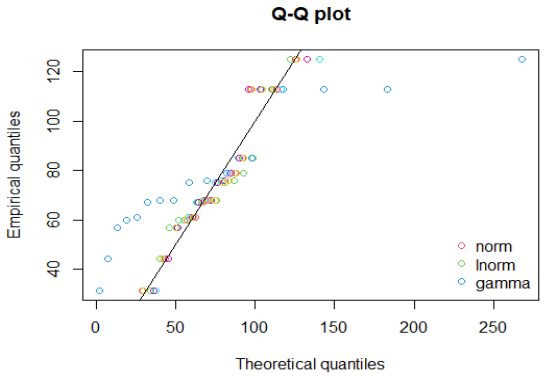


(h) Dispatch pipes to customers

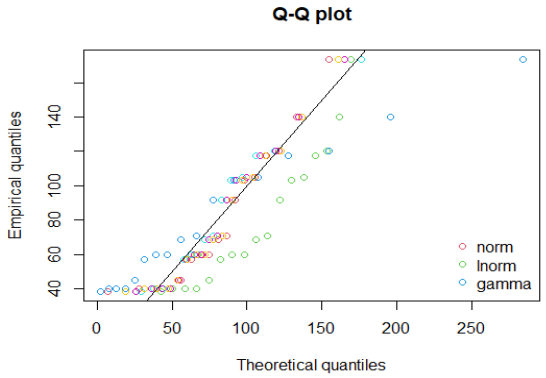
Figure G.4: Histogram and Theoretical Densities of Operations' Process Times



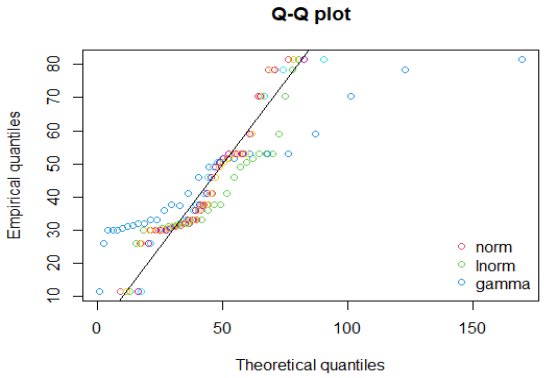
(a) Dress



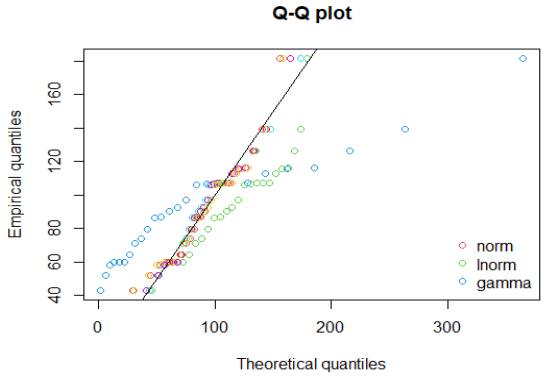
(b) Dress



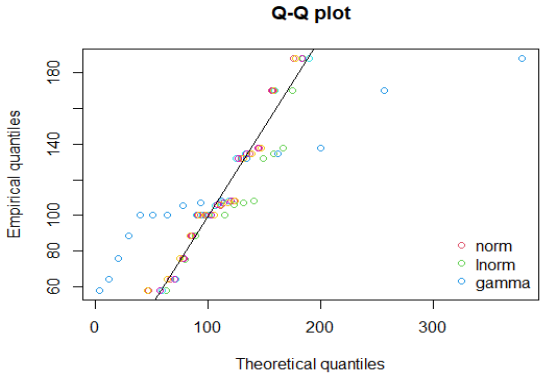
(c) Dress



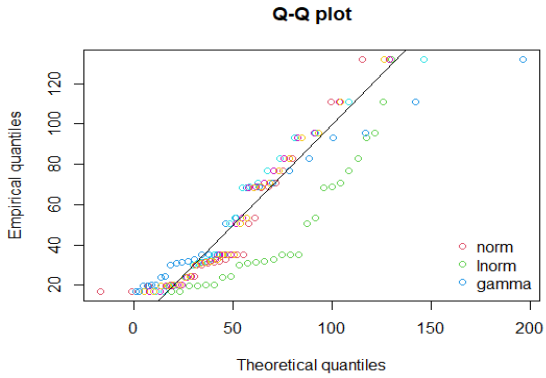
(d) Dress



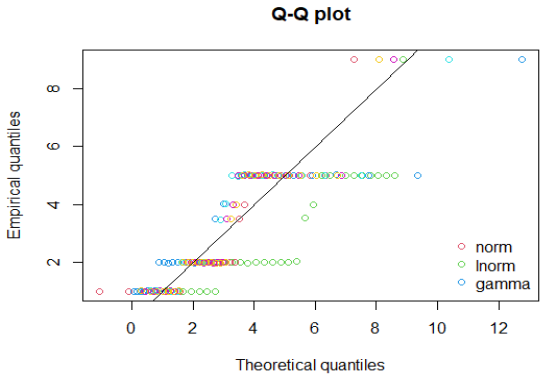
(e) Dress



(f) Dress



(g) Dress



(h) Dress

Figure G.5: Q-Q Plots of Operations' Process Times

Table G.1: Distribution Identification for Workstations' Processing Time

Workstation	Cullen and Frey identified distributions	chisq <chisq_c ?	P-value	Identified distribution fit
s_line	Weibull	TRUE	0.2000155	Gamma(6.1342, 0.2392)
	Gamma	TRUE	0.2128589	
	Lognormal	TRUE	0.1867302	
dress	Lognormal	FALSE	0	Random.Discrete (0.425, 0, 0.881, 0.536, 1.34, 0.75, 1.79, 0.893, 2.25, 1)
	Gamma	FALSE	0	
	Weibull	FALSE	0	
roll	Lognormal	TRUE	0.1848178	Gamma(8.4466, 0.2286)
	Gamma	TRUE	0.2945342	
	Weibull	TRUE	0.2454064	
l_weld	Lognormal	TRUE	0.07953934	Weibull(2.224955, 2.288199)
	Gamma	TRUE	0.1121951	
	Weibull	TRUE	0.1586613	
roll (re-roll)	Lognormal	FALSE	0.000168205	Gamma(6.7164, 0.1582)
	Gamma	TRUE	0.499832	
	Weibull	FALSE	0	
assem (9 m)	Gamma	TRUE	0.561677	Gamma(8.8413, 0.7831)
	Lognormal	TRUE	0.5051147	
	Weibull	TRUE	0.4777362	
c_weld (9 m)	Lognormal	FALSE	0.01951081	Gamma(9.4628, 0.8820)
	Gamma	TRUE	0.5553709	
	Weibull	TRUE	0.3127234	
assem (12 m)	Gamma	FALSE	0.02305088	Random.Discrete (1.08, 0, 1.94, 0.346, 2.81, 0.769, 3.67, 0.962, 4.54, 1)
	Lognormal	FALSE	0.03987296	
	Weibull	FALSE	0.0008780395	
c_weld (12 m)	Lognormal	TRUE	0.6578113	Lognormal(0.9740001, 0.3163829)
	Gamma	TRUE	0.6550476	
	Weibull	TRUE	0.5664864	
inspect	Lognormal	TRUE	0.5102275	Weibull(1.6829, 5.5533)
	Gamma	TRUE	0.6121028	
	Weibull	TRUE	0.7252747	
disp	Lognormal	FALSE	3.71E-05	Random.Discrete (0.1, 0, 0.3, 0.567, 0.5, 0.633, 0.7, 0.967, 0.9, 1)
	Gamma	FALSE	5.39E-05	
	Weibull	FALSE	7.87E-05	

G.2 Coding Done in RStudio

```
################################################################################
############################## Masters_Processing Time ##########################
################################################################################

################### IMPORT DATA ##################

## Back-up of the dataset ##
data <- read.csv("C:/Users/10018078/Documents/Masters/Data/Processing Times/Processing_Time1.csv", header = TRUE) #Import data
View(data)

## Data to be used ##
dfp<- data
View(dfp)
str(dfp)

#####################################################
################### Data Cleansing ##################
#####################################################
## Data Cleaning in R with Real Campaign Data! ##
#install.packages("dplyr")
library(dplyr)
library(tidyverse)
library(openxlsx) #open up text file
library(plotly)
library(zoo)


#3) Remove missing values/ nulls
#Identify NA's with completed cases function:
df1p <- dfp
complete.cases(df1p)
#Alternative method - use summary function - exact number and columns
summary(df1p)

#4) Remove/ Correct Garbage Data
str(df1p)
df_toetsp <- df1p

df_newp <- df_toetsp[apply(df_toetsp!=0, 1, all),] #remove rows with 0 in
str(df_newp)
summary(df_newp)

#5) Quick Insights - get frequency distribution
hist(df_newp$Actual_work)

# Calculate number of na values
str(df_newp)
summary(df_newp)
sum(is.na(df_newp))


#####################################################################
##### SPLIT DATA TO BE USED IN SERVERS! #####
View(df_newp)
summary(df_newp)

df_split_work <- split(df_newp, df_newp$Work_center)
df_split_work

df_split_short <- split(df_newp, df_newp$Op_short_text)
df_split_short



# ESMS6PLN (DO PLANNING)
planning_w <- df_split_work$ESMS6PLN
planning_w

planning_s <- df_split_short$`DO PLANNING`
planning_s

planning <- planning_s

# ESMS6BLM (MARK-OFF & S/LINE CUT PLATES)
sline_w <- df_split_work$ESMS6BLM
sline_w

sline_s <- df_split_short$`MARK-OFF AND S/LINE CUT PLATES`
sline_s

sline <- sline_s

# ESMS6MOP (DRESS PLATES FOR ROLLING)
dress_w <- df_split_work$ESMS6MOP
dress_w

dress_s <- df_split_short$`DRESS PLATES FOR ROLLING`
dress_s

dress <- dress_s

```

```
# ESMS6BLM (SET & ROLL PIPE SECTIONS)
roll_s <- df_split_short$`SET AND ROLL PIPE SECTIONS`
roll_s

roll <- roll_s

# ESMS6WLD (WELD PIPE SECTIONS) aka Long Weld
lweld_s <- df_split_short$`WELD PIPE SECTIONS`
lweld_s

lweld <- lweld_s

# ESMS6BLM (RE-ROLL PIPE SECTIONS)
reroll_s <- df_split_short$`RE-ROLL PIPE SECTION`
reroll_s

reroll <- reroll_s

# ESMS6BLM (ASSY PIPE SECTIONS 12 M LONG & NUMBER)
assem_s <- df_split_short$`ASSY PIPE SECTIONS 12 M LONG AND NUMBER`
assem_s

assem <- assem_s

# ESMS6WLD (FINAL WELD 12M LONG PIPES)
sweld_s <- df_split_short$`FINAL WELD PIPES 12 M LONG`
sweld_s

sweld <- sweld_s

# ESMS6MOP (DRESS & REMOVE ALL SPLATTER,UNDER CUTS)
splatter_s <- df_split_short$`DRESS & REMOVE ALL SPLATTER,UNDER CUTS`
splatter_s

splatter <- splatter_s

# ESMS6PLN (INPECTOR INSPECT EACH PIPE AND NUMBER)
inspect_s <- df_split_short$`INPECTOR INSPECT EACH PIPE AND NUMBER`
inspect_s

inspect <- inspect_s

# ESMS5DES (DESPT TO SECTION)
disp_s <- df_split_short$`DESPT TO SECTION`
disp_s

disp <- disp_s

########################################################################
########## OUTLIERS (Export R to Excel) ##########
# I = [q0.25 - 1.5*IQR; q0.75+1.5*IQR]
library(ggplot2)
library(openxlsx)
library(rio)

## ESMS6PLN (DO PLANNING)
summary(planning$Actual_work)
boxplot(planning$Actual_work)
planning_out <- boxplot.stats(planning$Actual_work)$out
planning_ind <- which(planning$Actual_work %in% c(planning_out)) #say in which rows is outliers
planning_ind

planning0 <- planning[-c(1),]
boxplot(planning0$Actual_work)
# No Distribution!!


## ESMS6BLM (MARK-OFF & S/LINE CUT PLATES)
summary(sline$Actual_work)
boxplot(sline$Actual_work)
sline_out <- boxplot.stats(sline$Actual_work)$out
sline_ind <- which(sline$Actual_work %in% c(sline_out)) #say in which rows is outliers
sline_ind

sline0 <- sline[-c(3),]
boxplot(sline0$Actual_work)

summary(sline0)
View(sline0)

## ESMS6MOP (DRESS PLATES FOR ROLLING)
summary(dress$Actual_work)
boxplot(dress$Actual_work)
dress_out <- boxplot.stats(dress$Actual_work)$out
dress_ind <- which(dress$Actual_work %in% c(dress_out)) #say in which rows is outliers
dress_ind

```

```
dress0 <- dress[-c(4),]
boxplot(dress0$Actual_work)

## ESMS6BLM (SET & ROLL PIPE SECTIONS)
summary(roll$Actual_work)
boxplot(roll$Actual_work)
roll_out <- boxplot.stats(roll$Actual_work)$out
roll_ind <- which(roll$Actual_work %in% c(roll_out)) #say in which rows is outliers
roll_ind

#roll0 <- roll[-c(),]
#boxplot(roll0$Actual_work)
# No outliers

roll0 <- roll

## ESMS6WLD (WELD PIPE SECTIONS) aka Long Weld
summary(lweld$Actual_work)
boxplot(lweld$Actual_work)
lweld_out <- boxplot.stats(lweld$Actual_work)$out
lweld_ind <- which(lweld$Actual_work %in% c(lweld_out)) #say in which rows is outliers
lweld_ind

#lweld0 <- lweld[-c(),]
#boxplot(lweld0$Actual_work)
# No outliers

lweld0 <- lweld

## ESMS6BLM (RE-ROLL PIPE SECTIONS)
summary(reroll$Actual_work)
boxplot(reroll$Actual_work)
reroll_out <- boxplot.stats(reroll$Actual_work)$out
reroll_ind <- which(reroll$Actual_work %in% c(reroll_out)) #say in which rows is outliers
reroll_ind

reroll0 <- reroll[-c(2),]
boxplot(reroll0$Actual_work)

## ESMS6BLM (ASSY PIPE SECTIONS 12 M LONG & NUMBER)
summary(assem$Actual_work)
boxplot(assem$Actual_work)
assem_out <- boxplot.stats(assem$Actual_work)$out
assem_ind <- which(assem$Actual_work %in% c(assem_out)) #say in which rows is outliers
assem_ind

assem0 <- assem[-c(12, 18),]
boxplot(assem0$Actual_work)

## ESMS6WLD (FINAL WELD 12M LONG PIPES)
summary(sweld$Actual_work)
boxplot(sweld$Actual_work)
sweld_out <- boxplot.stats(sweld$Actual_work)$out
sweld_ind <- which(sweld$Actual_work %in% c(sweld_out)) #say in which rows is outliers
sweld_ind

sweld0 <- sweld[-c(9),]
boxplot(sweld0$Actual_work)

## ESMS6MOP (DRESS & REMOVE ALL SPLATTER,UNDER CUTS)
summary(splatter$Actual_work)
boxplot(splatter$Actual_work)
splatter_out <- boxplot.stats(splatter$Actual_work)$out
splatter_ind <- which(splatter$Actual_work %in% c(splatter_out)) #say in which rows is outliers
splatter_ind

#splatter0 <- splatter[-c(9),]
#boxplot(splatter0$Actual_work)
# No outliers

splatter0 <- splatter

## ESMS6PLN (INPECTOR INSPECT EACH PIPE AND NUMBER)
summary(inspect$Actual_work)
boxplot(inspect$Actual_work)
inspect_out <- boxplot.stats(inspect$Actual_work)$out
inspect_ind <- which(inspect$Actual_work %in% c(inspect_out)) #say in which rows is outliers
inspect_ind

inspect0 <- inspect[-c(6, 9),]
boxplot(inspect0$Actual_work)

# No Distribution !!

## ESMS5DES (DESPT TO SECTION)
summary(disp$Actual_work)
boxplot(disp$Actual_work)
disp_out <- boxplot.stats(disp$Actual_work)$out
disp_ind <- which(disp$Actual_work %in% c(disp_out)) #say in which rows is outliers
disp_ind
```

```
disp0 <- disp[-c(18, 22),]
boxplot(disp0$Actual_work)

#####################################################################
################### Exploratory Data Analysis (ERP) ##################
#####################################################################

################################################
############ #Example From: http://www.di.fc.ul.pt/~jpn/r/distributions/fitting.html

####################### ESMS6BLM (MARK-OFF & S/LINE CUT PLATES) ########################################
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
library(latex2exp)
library(ggplot2)
summary(sline0)

View(sline0)

sline0$Actual_work <- sline0$Actual_work /120 #this is the amount of time PER METER! ################## Check hier vir 120m
sline0$Actual_work <- sline0$Actual_work * 3 #amount hours per plate

head(sline0)
mean(sline0$Actual_work)

sline0$Work <- sline0$Work /120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!
sline0$Work <- sline0$Work * 3 #amount hours per plate

mean(sline0$Work)

View(sline0)


export(sline0, "ProcessingTime_sline_StatFit.xlsx")

fit_sline0 <- fitdistr(sline0$Actual_work, densfun = "gamma")

hist(sline0$Actual_work, pch = 20, breaks = 5, prob = T,
     main = NULL,
     ylab = TeX("Frequency"),
     xlab = TeX("Processing Time (hours)"),
    col = "#d3d3d3")

curve(dgamma(x, fit_sline0$estimate[1], fit_sline0$estimate[2]), col = " rgb(108, 61, 145)", lwd = 1.5, add = T)

x <- sline0$Actual_work

a<- 6.1342
b <- 0.2392

hist(x, pch = 20, breaks = 5, prob = F,
     main = NULL,
     ylab = TeX("Frequency"),
     xlab = TeX("Processing Time (hours)"),
     col = "#d3d3d3")

curve(dgamma(x,a,b),from=0,to=160,col="red",lwd=2,add=T)


x <- sline0$Actual_work
library(fitdistrplus)
model<-fitdist(x, "gamma")
print(model$estimate)
#     shape      rate
#    6.2742206 0.1068711
a=model$estimate[1]
b=model$estimate[2]

# Get maximum y value of the curve
max_curve_y <- max(curve_data$y)

h<-hist(x, breaks=5,probability = T,
        main = NULL,
        ylab = TeX("Density"),
        xlab = TeX("Processing Time (hours)"),
        col = "#d3d3d3")
curve(dgamma(x,a,b),from=20,to=120,col="#6C3D91",lwd=2,add=T)


################ Draw Gamma over Histogram
# Your existing code for histogram
h <- hist(x, breaks = 5, probability = TRUE,
          main = NULL,
          ylab = TeX("Density"),
          xlab = TeX("Processing Time (hours)"),
          col = "#d3d3d3")
```

```r
# Fitting the gamma distribution parameters
fit_params <- fitdistr(x, densfun = "gamma")
a <- fit_params$estimate[1]
b <- fit_params$estimate[2]

# Plot the histogram
plot(h, xlim = c(20, 120),
     main = NULL,
     ylab = TeX("Frequency"),
     xlab = TeX("Processing Time (hours)"),
     col = "#d3d3d3")

# Adding the gamma distribution curve
#curve(dgamma(x, a, b), from = 20, to = 120, col = "#6C3D91", lwd = 2, add = TRUE)

# Adjust y-axis limits
curve_data <- data.frame(x = seq(20, 120, length.out = 100))
curve_data$y <- dgamma(curve_data$x, a, b)
max_curve_y <- max(curve_data$y)

# Set y-axis limits to accommodate the curve
ylim <- c(0, max_curve_y * 1.1)  # Adjust the multiplier as needed
par(usr = c(par()$usr[1:2], ylim))  # Set new y-axis limits

# Adding the gamma distribution curve again with adjusted y-axis limits
curve(dgamma(x, a, b), from = 20, to = 120, col = "#6C3D91", lwd = 2, add = TRUE)

# Adding label with color stripe


labels <- c("Gamma(6.13, 0.24)")
legend("topright", legend = labels,
       cex = 0.8,
       inset = 0.01,
       pch = 15,
       bty = "n",
       col = "#6C3D91")

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(sline0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(sline0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_sline0_n <- fitdist(sline0$Actual_work, "norm")
fit_sline0_u <- fitdist(sline0$Actual_work, "unif")
fit_sline0_e <- fitdist(sline0$Actual_work, "exp")
fit_sline0_ln <- fitdist(sline0$Actual_work, "lnorm")
fit_sline0_g <- fitdist(sline0$Actual_work, "gamma")
fit_sline0_w <- fitdist(sline0$Actual_work, "weibull")
#fitdistr(sline0$Actual_work, densfun = "weibull", lower = 0)
summary(fit_sline0_g) #parameters of distributions

#### DETERMINE PARAMETERS OF GAMMA DISTRIBUTION ####

# Calculate sample mean and sample variance
sample_mean <- mean(sline0$Actual_work)
sample_variance <- var(sline0$Actual_work)

# Calculate parameters using method of moments
alpha <- (sample_mean^2) / sample_variance
beta <- sample_variance / sample_mean

# Print the estimated parameters
print(paste("Shape (alpha):", alpha))
print(paste("Scale (beta):", beta))

## Method 2 ##
library(MASS)

# Fit gamma distribution using MLE
fit_params <- fitdistr(sline0$Actual_work, "gamma")

# Print the estimated parameters
print(fit_params$estimate)

#### Distributions Fit ####
## AIC
fit_sline0_n$aic
fit_sline0_ln$aic
fit_sline0_g$aic
fit_sline0_w$aic
fit_sline0_u$aic
fit_sline0_e$aic
fit_sline0_w$bic

```

```r
#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_sline0_n, fit_sline0_u, fit_sline0_e, fit_sline0_ln, fit_sline0_g, fit_sline0_w), legendtext = plot.legend)
cdfcomp(list(fit_sline0_n, fit_sline0_u, fit_sline0_e, fit_sline0_ln, fit_sline0_g, fit_sline0_w), legendtext = plot.legend)
qqcomp(list(fit_sline0_n, fit_sline0_u, fit_sline0_e, fit_sline0_ln, fit_sline0_g, fit_sline0_w), legendtext = plot.legend)
ppcomp(list(fit_sline0_n, fit_sline0_u, fit_sline0_e, fit_sline0_ln, fit_sline0_g, fit_sline0_w), legendtext = plot.legend)


##########################################################################
#### Goodness of Fit Test - Chi-squared ##################################
##########################################################################
#Compare Empirical to Theoretical Frequencies
# Here we need to compare the frequencies of observations (o) in k bins
# to the theoretical frequencies (e for expected) we would get from a fitted distribution function.

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sline0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_sline0_w)

shape <- 2.752008
scale <- 1.654079

x <- dweibull(dress0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,120, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,120, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq


# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA
```

```
#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sline0$Actual_work, pch = 500, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
labels <- c("Fitted Gamma Distribution")
legend("topright", legend = labels,
       cex = 0.8,
       inset = 0.01,
       pch = 15,
       bty = "n",
       col = "red")

summary(fit_sline0_g)

shape <- 6.271358
scale <- 4.273968

x <- dgamma(sline$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,120, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "red", lwd = 2)


# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,120, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq


# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

```

```r
#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p



################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sline0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_sline0_ln)

mu <- 0.3016432
sigma <- 0.4123197


x <- dlnorm(sline$Actual_work, mu, sigma)
n <- length(x)



#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,120, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)
#plot(y_dlnorm)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq


# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
```

```
p <- 1 - pchisq(chisq, df); p

######################## ESMS6MOP (DRESS PLATES FOR ROLLING) ######################################
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
library(latex2exp)
library(rio)
library(openxlsx)
summary(dress0)

View(dress0)

summary(dress0$Actual_work)

dress0$Actual_work <- dress0$Actual_work /120 #this is the amount of time PER METER! ################## Check hier vir 120m
dress0$Actual_work <- dress0$Actual_work * 3 #amount hours per plate

head(dress0)
mean(dress0$Actual_work)

dress0$Work <- dress0$Work /120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!
dress0$Work  <- dress0$Work * 3 #amount hours per plate

mean(dress0$Work)

export(dress0, "ProcessingTime_dress_StatFit.xlsx")

fit_dress0 <- fitdistr(dress0$Actual_work, densfun = "normal")
hist(dress0$Actual_work, pch = 20, breaks = 4, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = TeX("Processing Times (hours)"),
     col = "#d3d3d3")
title(ylab =TeX("Density"), mgp = c(1,1,0))
curve(dnorm(x, fit_dress0$estimate[1], fit_dress0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(dress0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(dress0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_dress0_n <- fitdist(dress0$Actual_work, "norm")
fit_dress0_u <- fitdist(dress0$Actual_work, "unif")
fit_dress0_e <- fitdist(dress0$Actual_work, "exp")
fit_dress0_ln <- fitdist(dress0$Actual_work, "lnorm")
fit_dress0_g <- fitdist(dress0$Actual_work, "gamma")
fit_dress0_w <- fitdist(dress0$Actual_work, "weibull")
summary(fit_dress0_w) #parameters of distributions

#### Distributions Fit ####
## AIC
fit_dress0_n$aic
fit_dress0_u$aic
fit_dress0_e$aic
fit_dress0_ln$aic
fit_dress0_g$aic
fit_dress0_w$aic

summary(fit_dress0_n)


#### PLOT THE RESULTS ####
#par(mfrow = c(2,2)) #laat die grafieke langs mekaar
plot.legend <- c("lnorm", "gamma", "weibull")
denscomp(list(fit_dress0_ln, fit_dress0_g, fit_dress0_w), legendtext = plot.legend)
cdfcomp(list(fit_dress0_ln, fit_dress0_g, fit_dress0_w), legendtext = plot.legend)
qqcomp(list(fit_dress0_ln, fit_dress0_g, fit_dress0_w), legendtext = plot.legend)
ppcomp(list(fit_dress0_ln, fit_dress0_g, fit_dress0_w), legendtext = plot.legend)


################################################################################
#### Goodness of Fit Test - Chi-squared ########################################
################################################################################

#################################
## TEST LOGNORMAL DISTRIBUTION ##
#################################
# store histogram as an object
x_hist <- hist(dress0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
```

```r
#mu <- mean(sline0$Actual_work)

#sigma <- sqrt(sum((sline0$Actual_work-mean(sline0$Actual_work))^2/(length(sline0$Actual_work)-1)))

x <- dlnorm(dress0$Actual_work, mu, sigma)
n <- length(x)


summary(fit_dress0_ln)
mu <- 3.6762231
sigma <- 0.4168786


#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,120, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)
#plot(y_dlnorm)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq


# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(dress0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_dress0_g)

shape <- 5.6975531
```

```r
scale <- 0.1318078

x <- dgamma(dress0$Actual_work, shape, scale)
n <- length(x)



#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,120, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)


# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,120, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p


################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(dress0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_dress0_w)

shape <- 2.372913
scale <- 48.992790

x <- dweibull(dress0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,120, by = 0.1)
```

```r
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,120, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

######################### ESMS6BLM (SET & ROLL PIPE SECTIONS) ########################################
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(roll0)
View(roll0)

summary(roll0$Actual_work)

roll0$Actual_work <- roll0$Actual_work /120 #this is the amount of time PER METER! ################### Check hier vir 120m
roll0$Actual_work <- roll0$Actual_work * 3 #amount hours per plate

View(roll0)

mean(roll0$Actual_work)

roll0$Work <- roll0$Work /120 #this is the amount of time PER METER! ################### Check hier vir 120m omskakeling!
roll0$Work  <- roll0$Work * 3 #amount hours per plate

mean(roll0$Work)

fit_roll0 <- fitdistr(roll0$Actual_work, densfun = "normal")
hist(roll0$Actual_work, pch = 20, breaks = 6, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = TeX("Processing Times (hours)"),
     col = "#d3d3d3")
title(ylab =TeX("Density"), mgp = c(1,1,0))
curve(dnorm(x, fit_roll0$estimate[1], fit_roll0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(roll0$Actual_work, histo = TRUE, demp = TRUE, las = 1)
```

```
#### CULLEN AND FREY GRAPH ####
descdist(roll0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_roll0_n <- fitdist(roll0$Actual_work, "norm")
fit_roll0_u <- fitdist(roll0$Actual_work, "unif")
fit_roll0_e <- fitdist(roll0$Actual_work, "exp")
fit_roll0_ln <- fitdist(roll0$Actual_work, "lnorm")
fit_roll0_g <- fitdist(roll0$Actual_work, "gamma")
fit_roll0_w <- fitdist(roll0$Actual_work, "weibull")
summary(fit_roll0_g) #parameters of distributions

#### DETERMINE PARAMETERS OF GAMMA DISTRIBUTION ####

# Calculate sample mean and sample variance
sample_mean <- mean(roll0$Actual_work)
sample_variance <- var(roll0$Actual_work)

# Calculate parameters using method of moments
alpha <- (sample_mean^2) / sample_variance
beta <- sample_variance / sample_mean

# Print the estimated parameters
print(paste("Shape (alpha):", alpha))
print(paste("Scale (beta):", beta))

#### Distributions Fit ####
## AIC
fit_roll0_n$aic
fit_roll0_u$aic
fit_roll0_e$aic
fit_roll0_ln$aic
fit_roll0_g$aic
fit_roll0_w$aic

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("unif", "gamma", "weibull")
denscomp(list(fit_roll0_u, fit_roll0_g, fit_roll0_w), legendtext = plot.legend)
cdfcomp(list(fit_roll0_u, fit_roll0_g, fit_roll0_w), legendtext = plot.legend)
qqcomp(list(fit_roll0_u, fit_roll0_g, fit_roll0_w), legendtext = plot.legend)
ppcomp(list(fit_roll0_u, fit_roll0_g, fit_roll0_w), legendtext = plot.legend)

################################################################################
#### Goodness of Fit Test - Chi-squared ########################################
################################################################################

################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(roll0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_roll0_ln)

mu <- 0.5991363
sigma <- 0.3530581


x <- dlnorm(sline$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,120, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)
#plot(y_dlnorm)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")
```

```
1166
1167 #we use the diff() function to take the difference between cumulative prob values
1168 #evaluated at each bin interval value, Then, with the incremental prob within each bin
1169 #we multiply this by the total number of samples to get the theoretical frequencies.
1170 x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
1171 sum(x_e)
1172
1173 #################
1174 ## Chi-squared Test
1175 #################
1176 #Going to determine the Chi square statistic
1177 #Going to compare it to a critical Chi Square value,
1178 # to determine if our data are truly different from the Lognormal Distribution
1179
1180 #Null Hypothesis: Ho: Our data are from the Normal Distribution
1181 # ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
1182 chisq <- sum((x_o - x_e)^2/x_e)
1183 chisq
1184
1185 # Degrees of freedom: k -1-p where p is the # of parameter estimated
1186 # p = 2 for normal and logormal distributions (mu and sigma)
1187 df<- k - 1 - 2
1188
1189 #Calculate the critical Chisq value: chisq_c
1190 #for 0.05 significance level
1191 #(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
1192 chisq_c <-  qchisq(0.95, df); chisq_c
1193
1194 #Is our chisq value less than chisq_c??
1195 chisq < chisq_c
1196
1197 # If TRUE (less than) we fail to reject H0
1198 # If FALSE (greater than) we reject H0 and accept HA
1199
1200 #P(Chisq < c) : aka the p-value
1201 #Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
1202 #This is the probability that our data are NOT from a Nognormal Distribution
1203 p <- 1 - pchisq(chisq, df); p
1204
1205 ################################
1206 ## TEST GAMMA DISTRIBUTION ##
1207 ################################
1208 # store histogram as an object
1209 x_hist <- hist(roll0$Actual_work, pch = 20, breaks = 5, prob = T,
1210                main = NULL,
1211                yaxt = "n",
1212                ylab = " ",
1213                xlab = "Processing Times (hours)",
1214                col = "#d3d3d3")
1215 title(ylab ="Density", mgp = c(1,1,0))
1216
1217 summary(fit_roll0_g)
1218
1219 shape <- 8.634483
1220 scale <- 4.471018
1221
1222 x <- dgamma(roll0$Actual_work, shape, scale)
1223 n <- length(x)
1224
1225 #Plot GAMMA PDF over histogram
1226 x_dgamma <- seq(0,120, by = 0.1)
1227 y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
1228 lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)
1229
1230 # extract bin values
1231 x_bins <- x_hist$breaks
1232
1233 #extract counts in each bin (o_i)s
1234 x_o <- x_hist$counts
1235
1236 #Calculae number of bins,k
1237 k <- length(x_o)
1238
1239 #Now we calculate the theoretical frequencies (e) in each bin
1240 #using the normal CDF function to compare with observed frequencies, (o)
1241 X_gamma <- seq(0,120, by = 0.1)
1242 Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
1243 plot(X_gamma , Fx_gamma , type = "l")
1244 abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")
1245
1246 #we use the diff() function to take the difference between cumulative prob values
1247 #evaluated at each bin interval value, Then, with the incremental prob within each bin
1248 #we multiply this by the total number of samples to get the theoretical frequencies.
1249 x_e <- diff(pgamma(x_bins, shape, scale))*n
1250 sum(x_e)
1251
1252 #################
1253 ## Chi-squared Test
1254 #################
1255 #Going to determine the Chi square statistic
```

```
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(roll0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_roll0_w)

shape <- 3.270975
scale <- 2.156851

x <- dweibull(roll0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,140, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,120, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2
```

```
#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

####################### ESMS6WLD (WELD PIPE SECTIONS) aka Long Weld ########################################
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(lweld0)

lweld0$Actual_work <- lweld0$Actual_work /120 #this is the amount of time PER METER! ################## Check hier vir 120m
lweld0$Actual_work <- lweld0$Actual_work * 3 #amount hours per plate

mean(lweld0$Actual_work)

lweld0$Work <- lweld0$Work /120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!
lweld0$Work <- lweld0$Work * 3 #amount hours per plate


mean(lweld0$Work)

summary(lweld0)
View(lweld0)

#fit_lweld0 <- fitdistr(lweld0$Actual_work, densfun = "normal")
hist(lweld0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = TeX("Processing Times (hours)"),
     col = "#d3d3d3")
title(ylab =TeX("Density"), mgp = c(1,1,0))
#curve(dnorm(x, fit_lweld0$estimate[1], fit_lweld0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(lweld0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(lweld0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_lweld0_n <- fitdist(lweld0$Actual_work, "norm")
fit_lweld0_u <- fitdist(lweld0$Actual_work, "unif")
fit_lweld0_e <- fitdist(lweld0$Actual_work, "exp")
fit_lweld0_ln <- fitdist(lweld0$Actual_work, "lnorm")
fit_lweld0_g <- fitdist(lweld0$Actual_work, "gamma")
fit_lweld0_w <- fitdist(lweld0$Actual_work, "weibull")
summary(fit_lweld0_w) #parameters of distributions

View(lweld0)

library(MASS)

fit <- fitdistr(lweld0$Actual_work, "weibull")
shape_estimate <- fit$estimate["shape"]
scale_estimate <- fit$estimate["scale"]

shape_estimate
scale_estimate

k <- 2.224955
a<- 2.288199

# Calculate the mean
mean <- a * gamma(1 + 1/k)
print(mean)

#### Distributions Fit ####
## AIC
fit_lweld0_n$aic
fit_lweld0_u$aic
fit_lweld0_e$aic
fit_lweld0_ln$aic
fit_lweld0_g$aic
fit_lweld0_w$aic

#### PLOT THE RESULTS ####
```

```r
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_lweld0_n,fit_lweld0_u, fit_lweld0_e, fit_lweld0_ln, fit_lweld0_g, fit_lweld0_w), legendtext = plot.legend)
cdfcomp(list(fit_lweld0_n,fit_lweld0_u, fit_lweld0_e, fit_lweld0_ln, fit_lweld0_g, fit_lweld0_w), legendtext = plot.legend)
qqcomp(list(fit_lweld0_n,fit_lweld0_u, fit_lweld0_e, fit_lweld0_ln, fit_lweld0_g, fit_lweld0_w), legendtext = plot.legend)
ppcomp(list(fit_lweld0_n,fit_lweld0_u, fit_lweld0_e, fit_lweld0_ln, fit_lweld0_g, fit_lweld0_w), legendtext = plot.legend)

par(mfrow = c(2,2))
plot.legend <- dists_lweld0
denscomp(fit_lweld0, legendtext = plot.legend)
cdfcomp(fit_lweld0, legendtext = plot.legend)
qqcomp(fit_lweld0, legendtext = plot.legend)
ppcomp(fit_lweld0, legendtext = plot.legend)

##############################################################################
#### Goodness of Fit Test - Chi-squared ######################################
##############################################################################

################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(lweld0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

#mu <- mean(sline0$Actual_work)

#sigma <- sqrt(sum((sline0$Actual_work-mean(sline0$Actual_work))^2/(length(sline0$Actual_work)-1)))

summary(fit_lweld0_ln)
mu <- 0.5877083
sigma <- 0.4738802

x <- dlnorm(lweld0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,120, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)
```

```r
# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq


# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject HO when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c
```

```
#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(lweld0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))


summary(fit_lweld0_g)

shape <- 4.580628
scale <- 2.272700

x <- dgamma(lweld0$Actual_work, shape, scale)
n <- length(x)


#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,120, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculate number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,120, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
```

```
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(lweld0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_lweld0_w)

shape <- 2.224955
scale <- 2.288199

x <- dweibull(lweld0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,140, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,120, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

######################## ESMS6BLM (RE-ROLL PIPE SECTIONS) ########################################
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(reroll0)

reroll0$Actual_work <- reroll0$Actual_work /120 #this is the amount of time PER METER! ################### Check hier vir 120m
reroll0$Actual_work <- reroll0$Actual_work * 3 #amount hours per plate
```

```
View(reroll0)

mean(reroll0$Actual_work)

reroll0$Work<- reroll0$Work /120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!
reroll0$Work <- reroll0$Work * 3 #amount hours per plate

mean(reroll0$Work)

fit_reroll0 <- fitdistr(reroll0$Actual_work, densfun = "normal")
hist(reroll0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = TeX("Processing Times (hours)"),
     col = "#d3d3d3")
title(ylab =TeX("Density"), mgp = c(1,1,0))
#curve(dnorm(x, fit_reroll0$estimate[1], fit_reroll0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(reroll0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(reroll0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_reroll0_n <- fitdist(reroll0$Actual_work, "norm")
fit_reroll0_u <- fitdist(reroll0$Actual_work, "unif")
fit_reroll0_e <- fitdist(reroll0$Actual_work, "exp")
fit_reroll0_ln <- fitdist(reroll0$Actual_work, "lnorm")
fit_reroll0_g <- fitdist(reroll0$Actual_work, "gamma")
fit_reroll0_w <- fitdist(reroll0$Actual_work, "weibull")
summary(fit_reroll0_g) #parameters of distributions

#### DETERMINE PARAMETERS OF GAMMA DISTRIBUTION ####

# Calculate sample mean and sample variance
sample_mean <- mean(reroll0$Actual_work)
sample_variance <- var(reroll0$Actual_work)

# Calculate parameters using method of moments
alpha <- (sample_mean^2) / sample_variance
beta <- sample_variance / sample_mean

# Print the estimated parameters
print(paste("Shape (alpha):", alpha))
print(paste("Scale (beta):", beta))

#### Distributions Fit ####
## AIC
fit_reroll0_n$aic
fit_reroll0_u$aic
fit_reroll0_e$aic
fit_reroll0_ln$aic
fit_reroll0_g$aic
fit_reroll0_w$aic

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_reroll0_n,fit_reroll0_u, fit_reroll0_e, fit_reroll0_ln, fit_reroll0_g, fit_reroll0_w), legendtext = plot.lege
cdfcomp(list(fit_reroll0_n,fit_reroll0_u, fit_reroll0_e, fit_reroll0_ln, fit_reroll0_g, fit_reroll0_w), legendtext = plot.leger
qqcomp(list(fit_reroll0_n,fit_reroll0_u, fit_reroll0_e, fit_reroll0_ln, fit_reroll0_g, fit_reroll0_w), legendtext = plot.legend
ppcomp(list(fit_reroll0_n,fit_reroll0_u, fit_reroll0_e, fit_reroll0_ln, fit_reroll0_g, fit_reroll0_w), legendtext = plot.legend

##############################################################################
#### Goodness of Fit Test - Chi-squared ######################################
##############################################################################

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(reroll0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_reroll0_g)

shape <- 6.978965
scale <- 6.565555

x <- dgamma(reroll0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
```

```r
x_dgamma <- seq(0,120, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,120, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(reroll0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_reroll0_ln)

mu <- -0.01229644
sigma <- 0.39546849

x <- dlnorm(reroll0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,120, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculate number of bins,k
```

```r
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA
#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(reroll$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_reroll0_w)

shape <- 2.790038
scale <- 1.194196

x <- dweibull(reroll0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,140, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,120, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
```

```
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

######################## ESMS6BLM (ASSY PIPE SECTIONS 12 M LONG & NUMBER) ########################################
#### NINE METERS !! #### 999999999 ####
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(assem0)
View(assem0)

summary(assem0$Actual_work)

assem0$Actual_work <- assem0$Actual_work / 120 #this is the amount of time PER METER! ################### Check hier vir 120m
assem0$Actual_work <- assem0$Actual_work * 9  #amount hours per plate

View(assem0)

mean(assem0$Actual_work)

assem0$Work <- assem0$Work / 120 #this is the amount of time PER METER! ################### Check hier vir 120m omskakeling!
assem0$Work <- assem0$Work * 9  #amount hours per plate

mean(assem0$Work)

fit_assem0 <- fitdistr(assem0$Actual_work, densfun = "normal")
hist(assem0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = "Processing Times (hours)",
     col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
curve(dnorm(x, fit_assem0$estimate[1], fit_assem0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(assem0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(assem0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_assem0_n <- fitdist(assem0$Actual_work, "norm")
fit_assem0_u <- fitdist(assem0$Actual_work, "unif")
fit_assem0_e <- fitdist(assem0$Actual_work, "exp")
fit_assem0_ln <- fitdist(assem0$Actual_work, "lnorm")
fit_assem0_g <- fitdist(assem0$Actual_work, "gamma")
fit_assem0_w <- fitdist(assem0$Actual_work, "weibull")
summary(fit_assem0_g) #parameters of distributions

#### DETERMINE PARAMETERS OF GAMMA DISTRIBUTION ####
# Calculate sample mean and sample variance
sample_mean <- mean(assem0$Actual_work)
sample_variance <- var(assem0$Actual_work)

# Calculate parameters using method of moments
alpha <- (sample_mean^2) / sample_variance
beta <- sample_variance / sample_mean
```

```r
# Print the estimated parameters
print(paste("Shape (alpha):", alpha))
print(paste("Scale (beta):", beta))

#### Distributions Fit ####
## AIC
fit_assem0_n$aic
fit_assem0_u$aic
fit_assem0_e$aic
fit_assem0_ln$aic
fit_assem0_g$aic
fit_assem0_w$aic

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)
cdfcomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)
qqcomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)
ppcomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)

##########################################################################
#### Goodness of Fit Test - Chi-squared ##################################
##########################################################################

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(assem0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_assem0_g)

shape <- 9.406493
scale <- 1.358871

x <- dgamma(assem0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,210, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,210, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
```

```r
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(assem0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

#mu <- mean(sline0$Actual_work)

#sigma <- sqrt(sum((sline0$Actual_work-mean(sline0$Actual_work))^2/(length(sline0$Actual_work)-1)))

summary(fit_assem0_ln)
mu <- 1.8808812
sigma <- 0.3319485

x <- dlnorm(assem0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,120, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c
```

```
# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(assem0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_assem0_w)

shape <- 3.182807
scale <- 7.723710

x <- dweibull(assem0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,210, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,210, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

######################## ESMS6WLD (FINAL WELD 12M LONG PIPES) ########################################

#### CIRCULAR WELD 9 M ####!!!!!!!!
```

```
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(sweld0)

View(sweld0)

sweld0$Actual_work <- sweld0$Actual_work / 120 #this is the amount of time PER METER! ################## Check hier vir 120m
sweld0$Actual_work <- sweld0$Actual_work * 9  #amount hours per plate

mean(sweld0$Actual_work)

sweld0$Work <- sweld0$Work / 120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!
sweld0$Work <- sweld0$Work * 9  #amount hours per plate

mean(sweld0$Work)

View(sweld0)

fit_sweld0 <- fitdistr(sweld0$Actual_work, densfun = "normal")
hist(sweld0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = "Processing Times (hours)",
     col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
curve(dnorm(x, fit_sweld0$estimate[1], fit_sweld0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(sweld0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(sweld0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_sweld0_n <- fitdist(sweld0$Actual_work, "norm")
fit_sweld0_u <- fitdist(sweld0$Actual_work, "unif")
fit_sweld0_e <- fitdist(sweld0$Actual_work, "exp")
fit_sweld0_ln <- fitdist(sweld0$Actual_work, "lnorm")
fit_sweld0_g <- fitdist(sweld0$Actual_work, "gamma")
fit_sweld0_w <- fitdist(sweld0$Actual_work, "weibull")
summary(fit_sweld0_ln) #parameters of distributions

#### DETERMINE PARAMETERS OF GAMMA DISTRIBUTION ####

# Calculate sample mean and sample variance
sample_mean <- mean(sweld0$Actual_work)
sample_variance <- var(sweld0$Actual_work)

# Calculate parameters using method of moments
alpha <- (sample_mean^2) / sample_variance
beta <- sample_variance / sample_mean

# Print the estimated parameters
print(paste("Shape (alpha):", alpha))
print(paste("Scale (beta):", beta))


#### Distributions Fit ####
## AIC
fit_sweld0_n$aic
fit_sweld0_u$aic
fit_sweld0_e$aic
fit_sweld0_ln$aic
fit_sweld0_g$aic
fit_sweld0_w$aic

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)
cdfcomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)
qqcomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)
ppcomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)


##############################################################################
#### Goodness of Fit Test - Chi-squared ######################################
##############################################################################
################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sweld0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
```

```
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_sweld0_ln)
mu <- 2.0726124
sigma <- 0.3163829

x <- dlnorm(sweld0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,210, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sweld0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))


summary(fit_sweld0_g)

shape <- 10.317061
scale <- 1.235966

x <- dgamma(sweld0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
```

```r
x_dgamma <- seq(0,210, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,210, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq


# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sweld0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_sweld0_w)

shape <- 3.391036
scale <- 9.291728

x <- dweibull(sweld0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,210, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts
```

```r
#Calculate number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,210, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA
#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

######################## ESMS6BLM (ASSY PIPE SECTIONS 12 M LONG & NUMBER) ########################################
#### TWELVE METERS !! #### 12 12 12 12 12 12 12 ####
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(assem0)
View(assem0)

summary(assem0$Actual_work)

#Take back to 120m
assem0$Actual_work <- assem0$Actual_work / 9  #amount hours per plate
assem0$Actual_work <- assem0$Actual_work * 120 #this is the amount of time PER METER! ################## Check hier vir 120m

#Assemble one additional canister (thus 3 meters)
assem0$Actual_work <- assem0$Actual_work / 120 #this is the amount of time PER METER! ################## Check hier vir 120m
assem0$Actual_work <- assem0$Actual_work * 3  #amount hours per plate

View(assem0)

mean(assem0$Actual_work)

#Take back to 120m
assem0$Work <- assem0$Work / 9  #amount hours per plate
assem0$Work <- assem0$Work * 120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!

#Assemble one additional canister (thus 3 meters)
assem0$Work <- assem0$Work / 120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!
assem0$Work <- assem0$Work * 3  #amount hours per plate

mean(assem0$Work)

export(assem0, "ProcessingTime_assem0_12_StatFit.xlsx")

fit_assem0 <- fitdistr(assem0$Actual_work, densfun = "normal")
hist(assem0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = "Processing Times (hours)",
     col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
curve(dnorm(x, fit_assem0$estimate[1], fit_assem0$estimate[2]), col = "red", lwd = 1, add = T)
```

```r

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(assem0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(assem0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_assem0_n <- fitdist(assem0$Actual_work, "norm")
fit_assem0_u <- fitdist(assem0$Actual_work, "unif")
fit_assem0_e <- fitdist(assem0$Actual_work, "exp")
fit_assem0_ln <- fitdist(assem0$Actual_work, "lnorm")
fit_assem0_g <- fitdist(assem0$Actual_work, "gamma")
fit_assem0_w <- fitdist(assem0$Actual_work, "weibull")
summary(fit_assem0_g) #parameters of distributions

#### Distributions Fit ####
## AIC
fit_assem0_n$aic
fit_assem0_u$aic
fit_assem0_e$aic
fit_assem0_ln$aic
fit_assem0_g$aic
fit_assem0_w$aic

summary(fit_assem0_n)

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)
cdfcomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)
qqcomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)
ppcomp(list(fit_assem0_n,fit_assem0_u, fit_assem0_e, fit_assem0_ln, fit_assem0_g, fit_assem0_w), legendtext = plot.legend)

##############################################################################
#### Goodness of Fit Test - Chi-squared ######################################
##############################################################################

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(assem0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))


summary(fit_assem0_g)

shape <- 9.407090
scale <- 4.076052

x <- dgamma(assem0$Actual_work, shape, scale)
n <- length(x)


#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,210, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,210, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
```

```r
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq


# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(assem0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_assem0_ln)
```

```r
mu <- 0.7822689
sigma <- 0.3319485

x <- dlnorm(assem0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,120, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq
```

```
# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(assem0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_assem0_w)

shape <- 3.183160
scale <- 2.574351

x <- dweibull(assem0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,210, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,210, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

```

```
#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

####################### ESMS6WLD (FINAL WELD 12M LONG PIPES) #######################################

#### CIRCULAR WELD 12 M ####!!!!!!!!
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(sweld0)

#sweld0$Actual_work <- sweld0$Actual_work / 9  #amount hours per plate

#Take back to 120m
sweld0$Actual_work <- sweld0$Actual_work / 9  #amount hours per plate
sweld0$Actual_work <- sweld0$Actual_work * 120 #this is the amount of time PER METER! ################## Check hier vir 120m

sweld0$Actual_work <- sweld0$Actual_work / 120 #this is the amount of time PER METER! ################## Check hier vir 120m
sweld0$Actual_work <- sweld0$Actual_work * 3  #amount hours per plate

mean(sweld0$Actual_work)
View(sweld0)

#Take back to 120m
sweld0$Work <- sweld0$Work  / 9  #amount hours per plate
sweld0$Work  <- sweld0$Work  * 120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!

sweld0$Work  <- sweld0$Work  / 120 #this is the amount of time PER METER! ################## Check hier vir 120m omskakeling!
sweld0$Work  <- sweld0$Work * 3  #amount hours per plate

mean(sweld0$Work)

fit_sweld0 <- fitdistr(sweld0$Actual_work, densfun = "normal")
hist(sweld0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = "Processing Times (hours)",
     col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
curve(dnorm(x, fit_sweld0$estimate[1], fit_sweld0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(sweld0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(sweld0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_sweld0_n <- fitdist(sweld0$Actual_work, "norm")
fit_sweld0_u <- fitdist(sweld0$Actual_work, "unif")
fit_sweld0_e <- fitdist(sweld0$Actual_work, "exp")
fit_sweld0_ln <- fitdist(sweld0$Actual_work, "lnorm")
fit_sweld0_g <- fitdist(sweld0$Actual_work, "gamma")
fit_sweld0_w <- fitdist(sweld0$Actual_work, "weibull")
summary(fit_sweld0_ln) #parameters of distributions

#### Distributions Fit ####
## AIC
fit_sweld0_n$aic
fit_sweld0_u$aic
fit_sweld0_e$aic
fit_sweld0_ln$aic
fit_sweld0_g$aic
fit_sweld0_w$aic

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)
cdfcomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)
qqcomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)
ppcomp(list(fit_sweld0_n,fit_sweld0_u, fit_sweld0_e, fit_sweld0_ln, fit_sweld0_g, fit_sweld0_w), legendtext = plot.legend)

#############################################################################
#### Goodness of Fit Test - Chi-squared ####################################
#############################################################################
################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sweld0$Actual_work, pch = 20, breaks = 5, prob = T,
```

```
                main = NULL,
                yaxt = "n",
                ylab = " ",
                xlab = "Processing Times (hours)",
                col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_sweld0_ln)

mu <- 0.9740001
sigma <- 0.3163829

x <- dlnorm(sweld0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,210, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sweld0$Actual_work, pch = 20, breaks = 5, prob = T,
                main = NULL,
                yaxt = "n",
                ylab = " ",
                xlab = "Processing Times (hours)",
                col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_sweld0_g)

shape <- 10.31438
scale <- 3.70745
```

```
x <- dgamma(sweld0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,210, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,210, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution
#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq
# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(sweld0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_sweld0_w)

shape <- 3.390995
scale <- 3.097322

x <- dweibull(sweld0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,210, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
```

```
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,210, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA
#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

######################## ESMS6MOP (DRESS & REMOVE ALL SPLATTER,UNDER CUTS) ########################################
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(splatter0)

View(splatter0)

splatter0$Actual_work <- splatter0$Actual_work / 120 #this is the amount of time PER METER! ################### Check hier vir
splatter0$Actual_work <- splatter0$Actual_work *12  #amount hours per plate

mean(splatter0$Actual_work)

splatter0$Work <- splatter0$Work / 120 #this is the amount of time PER METER! ################### Check hier vir 120m omskakeli
splatter0$Work <- splatter0$Work *12  #amount hours per plate

mean(splatter0$Work)

View(splatter0)

fit_splatter0 <- fitdistr(splatter0$Actual_work, densfun = "normal")
hist(splatter0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = "Processing Times (hours)",
     col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
#curve(dnorm(x, fit_splatter0$estimate[1], fit_splatter0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(splatter0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(splatter0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_splatter0_n <- fitdist(splatter0$Actual_work, "norm")
fit_splatter0_u <- fitdist(splatter0$Actual_work, "unif")
fit_splatter0_e <- fitdist(splatter0$Actual_work, "exp")
fit_splatter0_ln <- fitdist(splatter0$Actual_work, "lnorm")
```

```
fit_splatter0_g <- fitdist(splatter0$Actual_work, "gamma")
fit_splatter0_w <- fitdist(splatter0$Actual_work, "weibull")
summary(fit_splatter0_ln) #parameters of distributions

library(MASS)
fit <- fitdistr(splatter0$Actual_work, "weibull")

shape_estimate <- fit$estimate["shape"]
scale_estimate <- fit$estimate["scale"]

shape_estimate
scale_estimate

# Define the parameters
k <- 1.6829
a <- 5.5533

# Calculate the mean
mean <- a * gamma(1 + 1/k)
print(mean)

#### Distributions Fit ####
## AIC
fit_splatter0_n$aic
fit_splatter0_u$aic
fit_splatter0_e$aic
fit_splatter0_ln$aic
fit_splatter0_g$aic
fit_splatter0_w$aic

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_splatter0_n,fit_splatter0_u, fit_splatter0_e, fit_splatter0_ln, fit_splatter0_g, fit_splatter0_w), legendtext
cdfcomp(list(fit_splatter0_n,fit_splatter0_u, fit_splatter0_e, fit_splatter0_ln, fit_splatter0_g, fit_splatter0_w), legendtext
qqcomp(list(fit_splatter0_n,fit_splatter0_u, fit_splatter0_e, fit_splatter0_ln, fit_splatter0_g, fit_splatter0_w), legendtext =
ppcomp(list(fit_splatter0_n,fit_splatter0_u, fit_splatter0_e, fit_splatter0_ln, fit_splatter0_g, fit_splatter0_w), legendtext =


##############################################################################
#### Goodness of Fit Test - Chi-squared ######################################
##############################################################################
################################
## TEST LOGNORMAL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(splatter0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_splatter0_ln)

mu <- -1.0854985
sigma <- 0.6159668

x <- dlnorm(splatter0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,210, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
```

```
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(splatter0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_splatter0_g)

shape <- 2.742987
scale <- 6.694810

x <- dgamma(splatter0$Actual_work, shape, scale)
n <- length(x)


#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,210, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,210, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq
```

```r
# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(splatter0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_splatter0_w)

shape <- 1.6828945
scale <- 0.4627734

x <- dweibull(splatter0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,210, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,210, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

```

```
#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

######################## ESMS5DES (DESPT TO SECTION) ########################################
####  HISTOGRAM ####
library(fitdistrplus)
library(MASS)
summary(disp0)
View(disp0)

summary(disp0$Actual_work)

disp0$Actual_work <- disp0$Actual_work / 120 #this is the amount of time PER METER! ################### Check hier vir 120m
disp0$Actual_work <- disp0$Actual_work  * 12  #amount hours per plate

mean(disp0$Actual_work)

disp0$Work <- disp0$Work / 120 #this is the amount of time PER METER! ################### Check hier vir 120m omskakeling!
disp0$Work <- disp0$Work  * 12  #amount hours per plate

mean(disp0$Work)

export(disp0, "ProcessingTime_disp0_StatFit.xlsx")

View(disp0)

fit_disp0 <- fitdistr(disp0$Actual_work, densfun = "normal")
hist(disp0$Actual_work, pch = 20, breaks = 8, prob = T,
     main = NULL,
     yaxt = "n",
     ylab = " ",
     xlab = "Processing Times (hours)",
     col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))
#curve(dnorm(x, fit_disp0$estimate[1], fit_disp0$estimate[2]), col = "red", lwd = 1, add = T)

####  EMPIRICAL DENSITY AND CUMULATIVE DISTRIBUTION ####
plotdist(disp0$Actual_work, histo = TRUE, demp = TRUE, las = 1)

#### CULLEN AND FREY GRAPH ####
descdist(disp0$Actual_work, discrete = FALSE, boot = 1000)

#### FIT SPECIFIC DISTRIBUTIONS ####
fit_disp0_n <- fitdist(disp0$Actual_work, "norm")
fit_disp0_u <- fitdist(disp0$Actual_work, "unif")
fit_disp0_e <- fitdist(disp0$Actual_work, "exp")
fit_disp0_ln <- fitdist(disp0$Actual_work, "lnorm")
fit_disp0_g <- fitdist(disp0$Actual_work, "gamma")
fit_disp0_w <- fitdist(disp0$Actual_work, "weibull")
summary(fit_disp0_ln) #parameters of distributions

#### Distributions Fit ####
## AIC
fit_disp0_n$aic
fit_disp0_u$aic
fit_disp0_e$aic
fit_disp0_ln$aic
fit_disp0_g$aic
fit_disp0_w$aic

summary(fit_disp0_n)

#### PLOT THE RESULTS ####
par(mfrow = c(2,2))
plot.legend <- c("norm", "unif", "exp", "lnorm", "gamma", "weibull")
denscomp(list(fit_disp0_n,fit_disp0_u, fit_disp0_e, fit_disp0_ln, fit_disp0_g, fit_disp0_w), legendtext = plot.legend)
cdfcomp(list(fit_disp0_n,fit_disp0_u, fit_disp0_e, fit_disp0_ln, fit_disp0_g, fit_disp0_w), legendtext = plot.legend)
qqcomp(list(fit_disp0_n,fit_disp0_u, fit_disp0_e, fit_disp0_ln, fit_disp0_g, fit_disp0_w), legendtext = plot.legend)
ppcomp(list(fit_disp0_n,fit_disp0_u, fit_disp0_e, fit_disp0_ln, fit_disp0_g, fit_disp0_w), legendtext = plot.legend)

##########################################################################
#### Goodness of Fit Test - Chi-squared ##################################
##########################################################################
#################################
## TEST LOGNORMAL DISTRIBUTION ##
#################################
# store histogram as an object
x_hist <- hist(disp0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
```

```
                col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

#mu <- mean(sline0$Actual_work)

#sigma <- sqrt(sum((sline0$Actual_work-mean(sline0$Actual_work))^2/(length(sline0$Actual_work)-1)))

summary(fit_disp0_ln)
mu <- 0.9287378
sigma <- 0.6620635

x <- dlnorm(disp0$Actual_work, mu, sigma)
n <- length(x)

#Plot lognormal PDF over histogram
x_dlnorm <- seq(0,210, by = 0.1)
y_dlnorm = dlnorm(x_dlnorm, mean = mu, sd = sigma)
lines(x_dlnorm, y_dlnorm, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_log <- seq(0,120, by = 0.1)
Fx_log <-plnorm(X_log, mean = mu, sd = sigma)
plot(X_log, Fx_log, type = "l")
abline(h = plnorm(x_bins, mean = mu, sd = sigma), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(plnorm(x_bins, mean = mu, sd = sigma))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST GAMMA DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(disp0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_disp0_g)

shape <- 2.5579416
scale <- 0.8207278
```

```r
x <- dgamma(disp0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dgamma <- seq(0,210, by = 0.1)
y_dgamma = dgamma(x_dgamma, shape, scale, log = FALSE)
lines(x_dgamma, y_dgamma, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_gamma <- seq(0,210, by = 0.1)
Fx_gamma <-pgamma(X_gamma, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_gamma , Fx_gamma , type = "l")
abline(h = pgamma(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pgamma(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2

#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p

################################
## TEST WEIBULL DISTRIBUTION ##
################################
# store histogram as an object
x_hist <- hist(disp0$Actual_work, pch = 20, breaks = 5, prob = T,
               main = NULL,
               yaxt = "n",
               ylab = " ",
               xlab = "Processing Times (hours)",
               col = "#d3d3d3")
title(ylab ="Density", mgp = c(1,1,0))

summary(fit_disp0_w)

shape <- 1.687803
scale <- 3.511924

x <- dweibull(disp0$Actual_work, shape, scale)
n <- length(x)

#Plot GAMMA PDF over histogram
x_dweibull <- seq(0,210, by = 0.1)
y_dweibull = dweibull(x_dweibull, shape, scale, log = FALSE)
lines(x_dweibull , y_dweibull, col = "purple", lwd = 2)

# extract bin values
x_bins <- x_hist$breaks

```

```r
#extract counts in each bin (o_i)s
x_o <- x_hist$counts

#Calculae number of bins,k
k <- length(x_o)

#Now we calculate the theoretical frequencies (e) in each bin
#using the normal CDF function to compare with observed frequencies, (o)
X_weibull <- seq(0,210, by = 0.1)
Fx_weibull <-pweibull(X_weibull, shape, scale, lower.tail = TRUE, log.p = FALSE)
plot(X_weibull , Fx_weibull , type = "l")
abline(h = pweibull(x_bins, shape, scale), lty = 2, col = "red")

#we use the diff() function to take the difference between cumulative prob values
#evaluated at each bin interval value, Then, with the incremental prob within each bin
#we multiply this by the total number of samples to get the theoretical frequencies.
x_e <- diff(pweibull(x_bins, shape, scale))*n
sum(x_e)

#################
## Chi-squared Test
#################
#Going to determine the Chi square statistic
#Going to compare it to a critical Chi Square value,
# to determine if our data are truly different from the Lognormal Distribution

#Null Hypothesis: Ho: Our data are from the Normal Distribution
# ALternative hypothesis: HA: Our data are not from the Lognormal Distribution
chisq <- sum((x_o - x_e)^2/x_e)
chisq

# Degrees of freedom: k -1-p where p is the # of parameter estimated
# p = 2 for normal and logormal distributions (mu and sigma)
df<- k - 1 - 2


#Calculate the critical Chisq value: chisq_c
#for 0.05 significance level
#(5% chance that we fail to reject H0 when data are indeed from Lognormal Dist)
chisq_c <-  qchisq(0.95, df); chisq_c

#Is our chisq value less than chisq_c??
chisq < chisq_c

# If TRUE (less than) we fail to reject H0
# If FALSE (greater than) we reject H0 and accept HA

#P(Chisq < c) : aka the p-value
#Calculate the chi-squared probability associated with a given alpha value!!!! aka p-value
#This is the probability that our data are NOT from a Nognormal Distribution
p <- 1 - pchisq(chisq, df); p


```

Appendix H

Model Development

H.1 Simio Library Objects

- **Source**: A Source is an object that allows the creation of entities at a specified rate, by a specified arrival pattern, on the firing of an event. The main functionality of the Source is to simulate systems where entities arrive, such as when clients enter a queue or when work orders arrive at a manufacturing factory.

- **Sink**: An object known as a Sink, removes entities from the simulated system. The Sink simulates the point at which system entities leave the system under investigation. For example, customers leaving a line or finished goods leaving a factory. An Input Buffer is present at the Sink. The main objective of the Sink object is to gather performance data for the entities leaving the system, such as waiting periods and throughput.

- **Server**: In Simio, a Server is a "resource that provides a specific service to the simulated system" [33]. The Server Object represents a capacitated resource with optional constrained input and output buffers. Simio enables users to specify numerous Server properties, including processing time, setup time, and availability. Simio users are able to define various characteristics of servers. Some examples of these characteristics include processing times, breakdown times, setup times, availability, and utilisation.

- **Combiner**: An object called a "Combiner" combines different flows of entities into a single flow. The Combiner has four stations with associated queues: a processing station, an output buffer station, a station for the parent input buffer, and a station for the member input buffer. Both unlimited and finite capacity queues are possible with the four-station approach. A Combiner's main purpose is to simulate systems where various parallel processes combine to form a single process, such as an assembly line that consists of multiple conveyors or a single lane of traffic.

- **Separator**: A Separator is an object that divides a single entity flow into several parallel flows. The Separator object represents a resource with one input buffer and two output buffers. The object has four stations with associated queues: an input buffer station, a processing station, a parent output buffer station, and a member output buffer station. Both unlimited and finite capacity queues are possible with the four-station approach. The main goal of a Separator is to simulate systems where one process splits into several parallel processes, such an example includes a conveyor belt that is divided into several assembly lines or a traffic lane that splits into several lanes.

- **Vehicle**: The Vehicle object is a mobile object that can move within a simulation model and transport entities from one location to another. When modelling different transportation systems, such as logistic networks and material handling systems, a vehicle can represent any type of transportation mechanism, such as trucks, cars, trains, or aeroplanes.

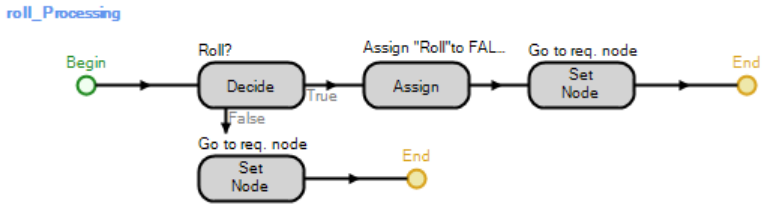
- **Worker**: The object that represents the Worker is a human or mechanical resource that carries out duties. Workers within a workforce environment are an important resource that is crucial

to form part of the system under investigation. Worker objects are classified according to their capacity, level of expertise, and availability schedules, which can be tailored to meet the particular requirements of the simulation model. The approach allows for the assignment of workers to particular workstations, tasks, and procedures.

- **Basic Node**: A Basic Node represents a node where entities can be allocated to an action of residing, waiting, or being processed. Various operational stations can be represented through the use of a Basic Node, such as queues, stations, or storage areas.
- **Transfer Node**: A Transfer Node transfers entities between distinct pathways. Transfer Nodes are frequently used to represent decision points where entities may be routed to various paths based on specific criteria or rules, such as junctions or intersections.
- **Connector**: Entities are able to travel between two nodes through the use of a Connector.
- **Path**: The pathway that entities follow as the entity travels through the simulation model is defined by a Path, which is a component that links two or more nodes. To reflect the material flow in which entities are transferred within the developed model, paths are used instead of connectors. The main reason for this is that paths take distance into consideration whereas connectors do not. Therefore to adequately reflect real-world operations, paths are utilised.

Appendix I

Process Development within Simio



(a) Steps Assigned

Properties: Roll? (Decide Step Instance)

Basic Logic	
Decide Type	ConditionBased
Condition Or Probability	**ModelEntity.Roll**
Advanced Options	
General	

(b) Decide Step

Properties: Assign "Roll"to FALSE (Assign Step Instance)

Basic Logic	
State Variable Name	**ModelEntity.Roll**
New Value	**False**
Assignments (More)	0 Rows
Advanced Options	
General	

(c) Assign Step

Properties: Go to req. node (SetNode Step Instance)

Basic Logic	
Destination Type	Specific
Node Name	**Input@l_weld**
Advanced Options	
General	

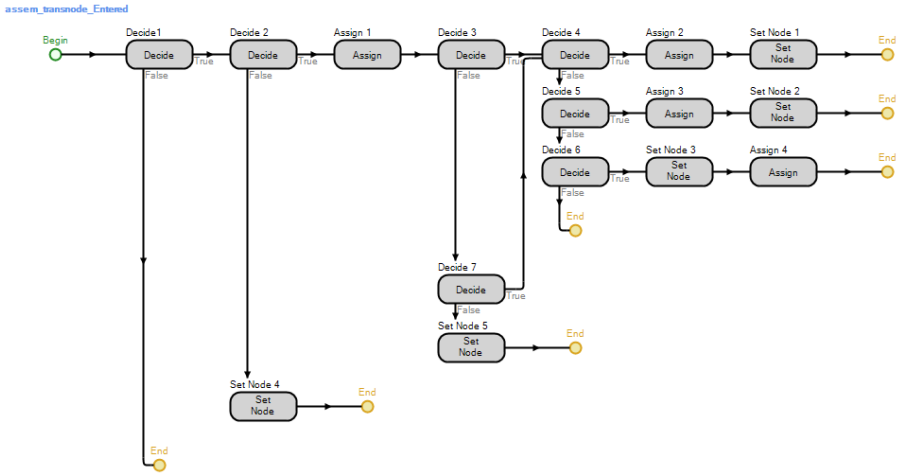
(d) Set Node Step - To Long Weld Workstation

Properties: Go to req. node (SetNode Step Instance)

Basic Logic	
Destination Type	Specific
Node Name	**assem_transnode**
Advanced Options	
General	

(e) Set Node Step - To Transfer Node for Assembly Workstation

Figure I.1: Roll Workstation - Two Operations Process



(a) Representation of process defined at *assem_transnode*

Properties: Decide1 (Decide Step Instance)

Basic Logic	
Decide Type	ConditionBased
Condition Or Probability	**Is.ModelEntity**
Advanced Options	
General	

(b) Decide 1

Properties: Decide 2 (Decide Step Instance)

Basic Logic	
Decide Type	ConditionBased
Condition Or Probability	**ModelEntity.transnode**
Advanced Options	
General	

(c) Decide 2

Properties: Assign 1 (Assign Step Instance)

Basic Logic	
State Variable Name	**ModelEntity.transnode**
New Value	**False**
Assignments (More)	0 Rows
Advanced Options	
General	

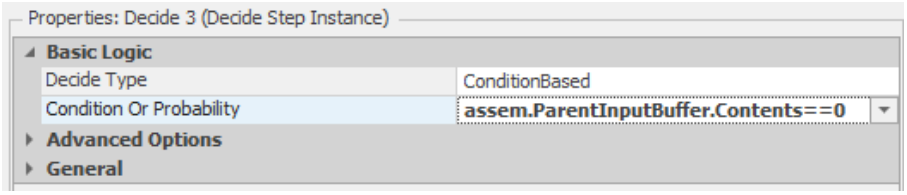
(d) Assign 1

Properties: Set Node 4 (SetNode Step Instance)

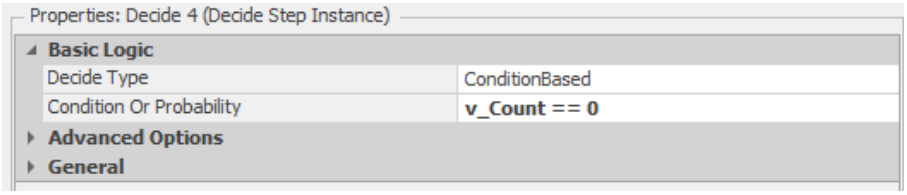
Basic Logic	
Destination Type	Specific
Node Name	**ParentInput@assem**
Advanced Options	
General	

(e) Set Node 4

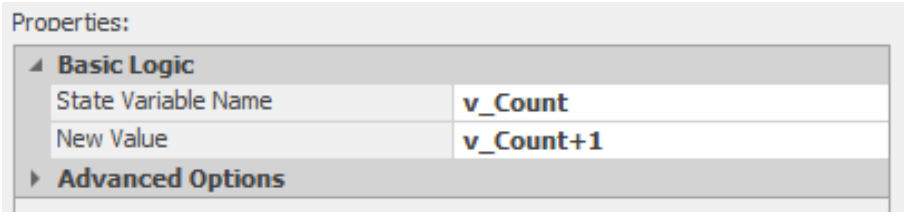
Figure I.2: Process at *assem_transnode*, First bundle of steps



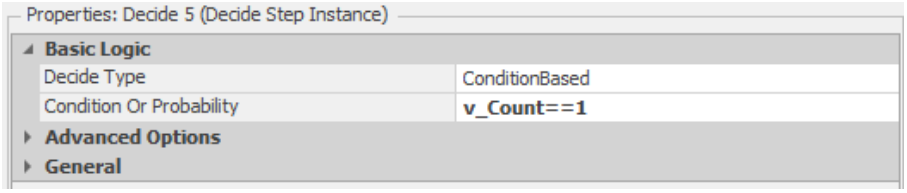
(a) Decide 3



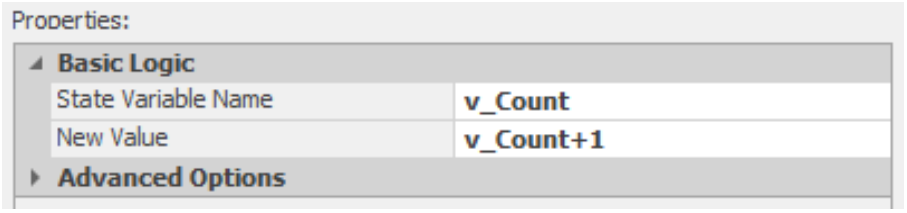
(b) Decide 4



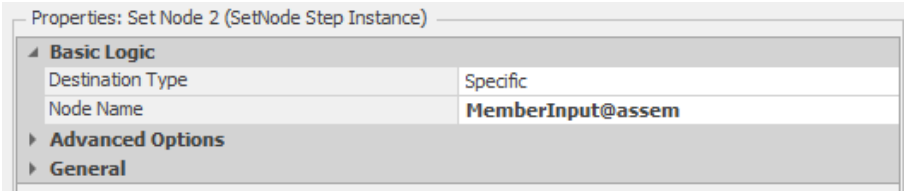
(c) Assign 2



(d) Decide 5

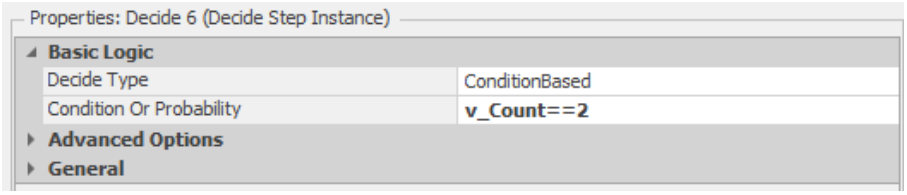


(e) Assign 3

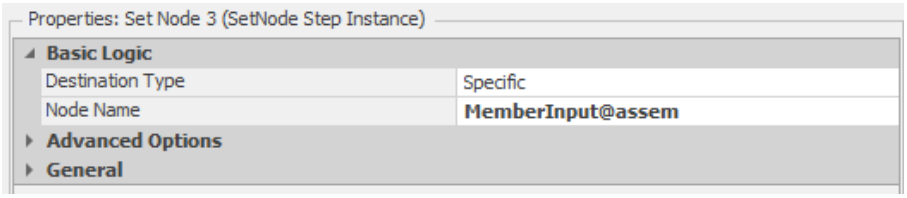


(f) Set Node 2

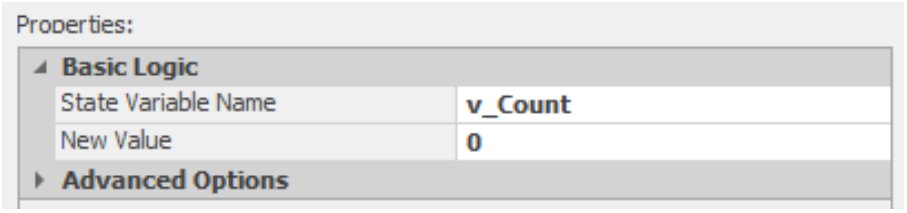
Figure I.3: Process at *assem_transnode*, Second bundle of steps



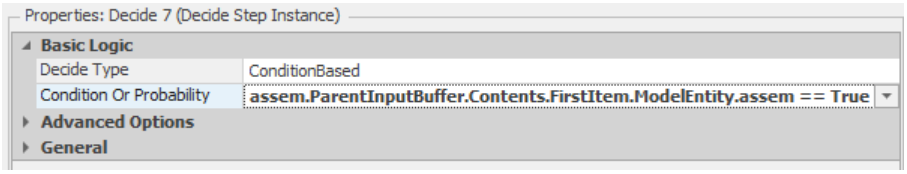
(a) Decide 6



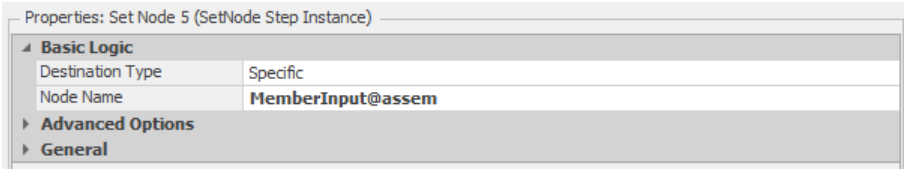
(b) Set Node 3



(c) Assign 4

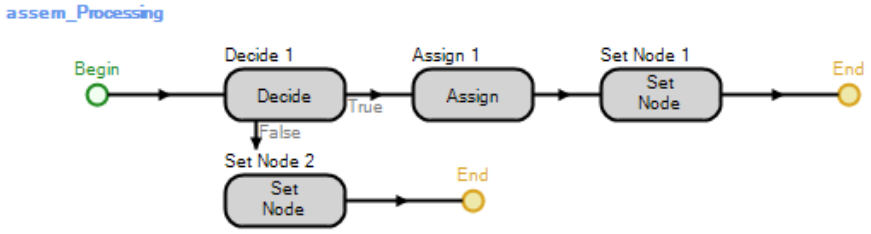


(d) Decide 7



(e) Set Node 5

Figure I.4: Process at *assem_transnode*, Third bundle of steps



(a) Representation of process defined as *assem_Processing*

Properties: Decide 1 (Decide Step Instance)

Basic Logic	
Decide Type	ConditionBased
Condition Or Probability	**ModelEntity.assem**
Advanced Options	
General	

(b) Decide 1

Properties: Assign 1 (Assign Step Instance)

Basic Logic	
State Variable Name	**ModelEntity.assem**
New Value	**False**
Assignments (More)	0 Rows
Advanced Options	
General	

(c) Assign 1

Properties: Set Node 1 (SetNode Step Instance)

Basic Logic	
Destination Type	Specific
Node Name	**Input@c_weld**
Advanced Options	
General	

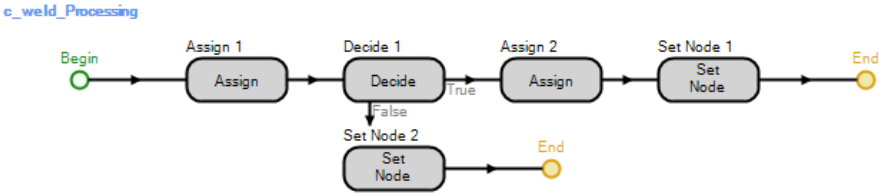
(d) Set Node 1

Properties: Set Node 2 (SetNode Step Instance)

Basic Logic	
Destination Type	Specific
Node Name	**Input@c_weld**
Advanced Options	
General	

(e) Set Node 2

Figure I.5: Process defined as *assem_Processing*



(a) Representation of Process Defined as *c_weld_Processing*

Properties: Assign 1 (Assign Step Instance)

Basic Logic	
State Variable Name	**ModelEntity.transnode**
New Value	**False**
Assignments (More)	0 Rows
Advanced Options	
General	

(b) Assign 1

Properties: Decide 1 (Decide Step Instance)

Basic Logic	
Decide Type	ConditionBased
Condition Or Probability	**ModelEntity.c_weld**
Advanced Options	
General	

(c) Decide 1

Properties: Assign 2 (Assign Step Instance)

Basic Logic	
State Variable Name	**ModelEntity.c_weld**
New Value	**False**
Assignments (More)	0 Rows
Advanced Options	
General	

(d) Assign 2

Properties: Set Node 1 (SetNode Step Instance)

Basic Logic	
Destination Type	Specific
Node Name	**assem_transnode**
Advanced Options	
General	

(e) Set Node 1

Properties: Set Node 2 (SetNode Step Instance)

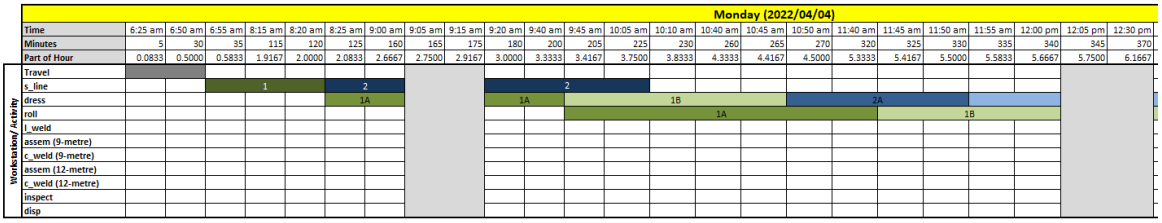
Basic Logic	
Destination Type	Specific
Node Name	**Input@inspect**
Advanced Options	
General	

(f) Set Node 2

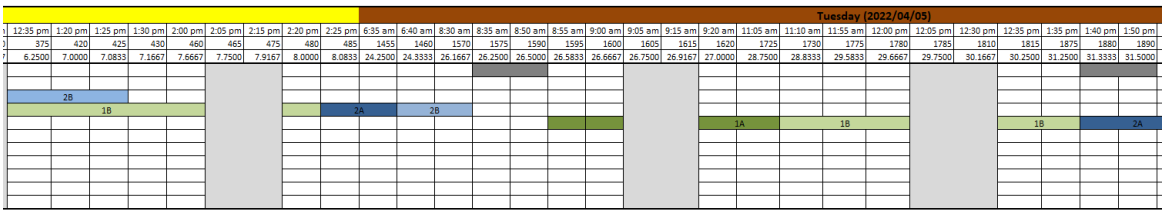
Figure I.6: Process defined as *assem_Processing*

Appendix J

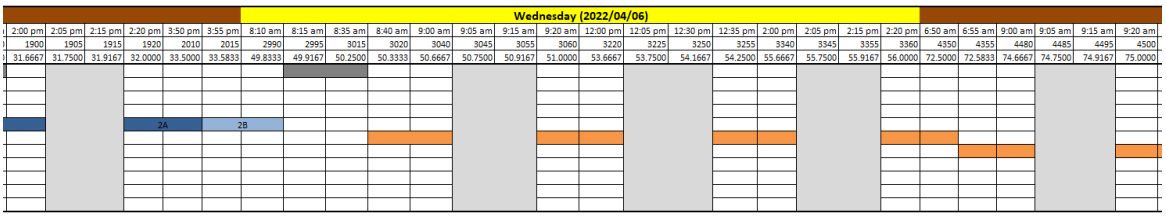
Validate Model-0.0



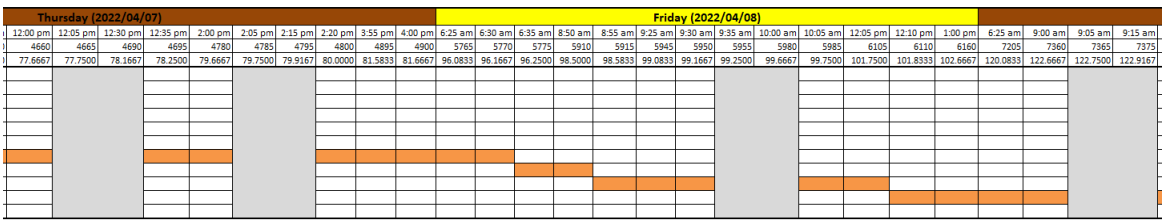
(a) Snipped 1 of Gantt Chart



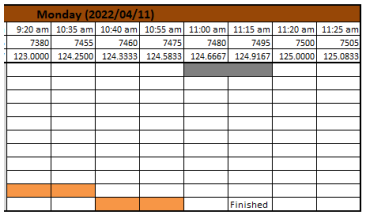
(b) Snipped 2 of Gantt Chart



(c) Snipped 3 of Gantt Chart



(d) Snipped 4 of Gantt Chart



(e) Snipped 5 of Gantt Chart

Figure J.1: Gantt Chart Developed for Validating Model-0.0

Appendix K

Opening in Wall

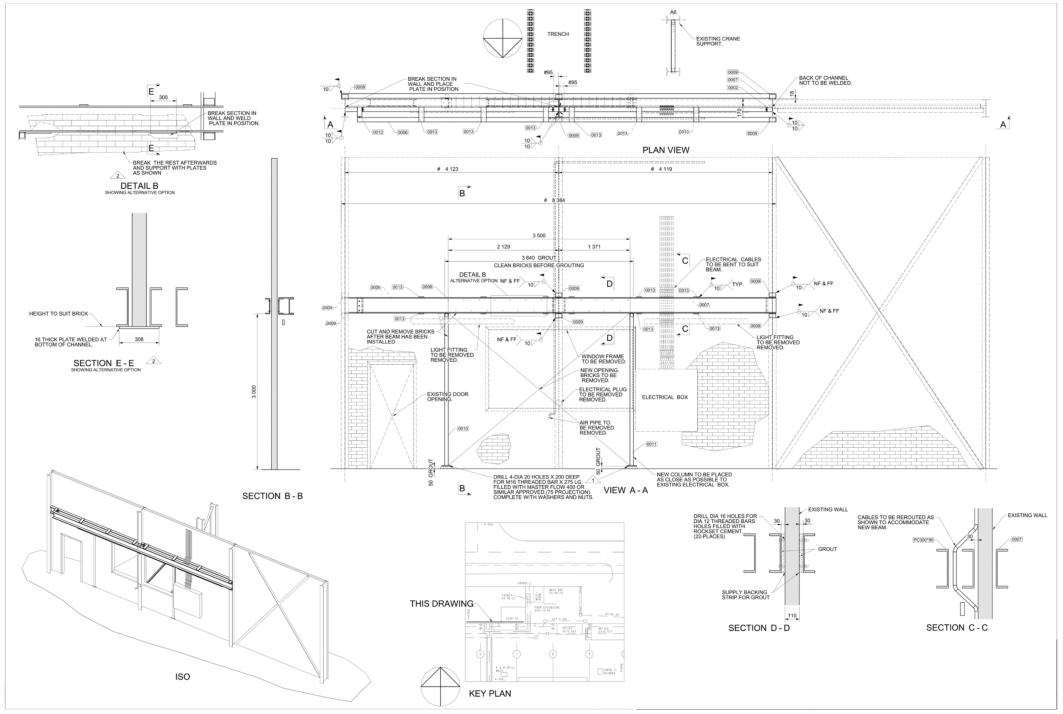


Figure K.1: Civil Drawing Regarding Opening in Wall to Improve Pipe Manufacturing Process Flow

Appendix L

Developed Model-1.0

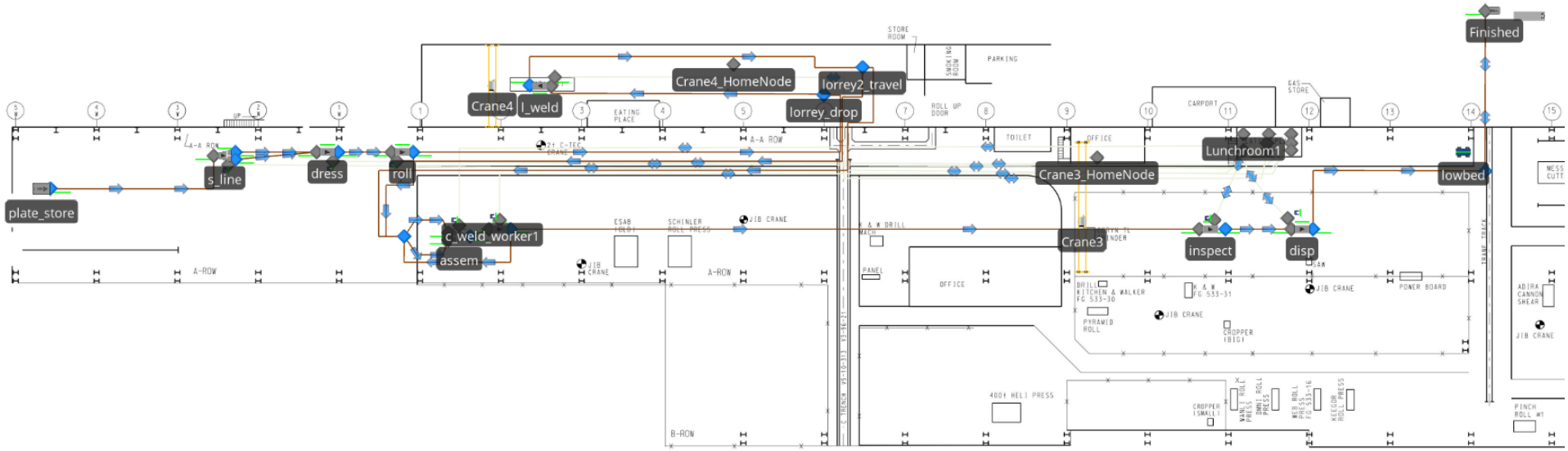


Figure L.1: Developed Model-1.0

Appendix M

Plagiarism Declaration

I, Laetitia Ackerman Haasbroek, a student at the North-West University of Potchefstroom in the Department of Industrial Engineering, with student number 30252180, declare:

1. I declare that this is my original work.
2. I understand what plagiarism is and I am aware of the University's policy in this regard.
3. Where other people's work has been used (either from the internet, a printed source, or any other source) this has been carefully acknowledged and referenced per departmental requirements.
4. I have not used another student's past work to hand in as my own.
5. I have not allowed, and will not allow, anyone to copy my work to hand it in as his/her own work.

Signed on 22 of November, 2023:

L.A. Haasbroek