CRANFIELD UNIVERSITY

NEHA VENILAL PRAJAPAT

ENHANCING THE DECISION MAKING CAPABILITIES OF DISCRETE EVENT SIMULATION USING OPTIMISATION AND VISUALISATION

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

PhD
Academic Year: 2016 - 2017

Supervisor: Prof Ashutosh Tiwari
May 2017

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

PhD

Academic Year 2016 - 2017

NEHA VENILAL PRAJAPAT

ENHANCING THE DECISION MAKING CAPABILITIES OF DISCRETE EVENT SIMULATION USING OPTIMISATION AND VISUALISATION

Supervisor: Prof Ashutosh Tiwari
May 2017

This thesis is submitted in partial fulfilment of the requirements for the degree of PhD

© Cranfield University 2017. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

ABSTRACT

Technological innovations for data collection and an increasingly competitive global market have led to an increase in the application of Discrete Event Simulation by manufacturing companies in recent years. Scenario analysis and optimisation methods are often applied to these simulation models to improve objectives such as cost, profit and throughput. New visualisation technologies such as Virtual Reality have gained significant interest in a variety of industries and can be linked with simulation to provide significant benefits.

The literature review has identified four main research gaps: the development of optimisation-ready simulation models, the application of multi-objective optimisation methods to simulation models, the development of an interactive immersive visualisation framework and the development of a framework for visualising the relationship between inputs and outputs of simulation models.

The aim of this research is to develop a framework to simulate, optimise and visualise manufacturing processes to assist stakeholders with decision making. A framework is presented to generate ready-to-optimise simulation models including an interface for input and output variables. Following the development of simulation models, a framework is proposed to optimise multiple objectives simultaneously using design of experiments and meta-models to create a Pareto front of solutions. The results show the resource allocation meta-model provides acceptable prediction accuracy whilst the lead time meta-model was not able to provide accurate prediction. A closed-loop immersive interactive interface for simulation is also presented. Regression trees have been proposed to assist stakeholders with understanding the relationships between input and output variables. The framework uses regression and classification trees with overlaid values for multiple objectives and random forests to improve prediction accuracy for new points. The research is validated using a set of real-life test cases.

This thesis was funded by General Electric and partly funded by Innovate UK through the Towards Zero Prototyping of Factory Layouts and Operations Using Novel Gaming Technologies Project (Proj. no. 101888).

ACKNOWLEDGEMENTS

Firstly, I would like to thank Dr. Nadir Ince and Dr. Xiao-Peng Gan who believed in me more than I believed in myself. Thank you for encouraging me to pursue a PhD, for sponsoring me and supporting me throughout the journey. Thank you especially to Prof. Ashutosh Tiwari for guiding me throughout the process and for encouraging me to be resilient. Special thanks to Windo Hutabarat, John Oyekan and Tony Waller for their advice and guidance, it has been a pleasure working with you and learning from you.

I would also like to thank all of the past and present members of TM4T and GE colleagues for making the journey so enjoyable; I have really enjoyed working with you, getting to know you and the constant supply of cakes.

Thank you so much to my parents for encouraging me to pursue higher education and supporting me throughout the process, I owe all of my achievements to you both. Thank you especially to my sisters and brother for listening to my endless PhD talk and helping me through the difficult times. I am also grateful to all of my friends who have been supportive and encouraged me throughout my PhD. To Jayson, thanks for being so patient with me throughout this time and for motivating me to complete. Thanks for always listening to me when I'm stressed and for making me laugh.

And finally, thank you to Becky for giving me the confidence to be who I am today. You have taught me so much about friendship, compassion and selflessness. I dedicate this thesis to you, may you rest in peace.

LIST OF PUBLICATIONS

1. Prajapat, N., Hutabarat, W., Tiwari, A., Pattacini, M. (2015) “Investigating Scheduling models for Power Plant Preventive Maintenance using Genetic Algorithm”, Multidisciplinary International Scheduling Conference (MISTA 2015), 25 - 28 Aug 2015, Prague, Czech Republic
2. Prajapat, N., Vejju, K., Tiwari, A. (2015) “Comparison of multi objective algorithms for discrete event simulation based turbine assembly model optimisation”, The 11th edition of the International Conference on Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems, EUROGEN-2015, 14 – 16 September 2015, Glasgow, UK
3. Prajapat, N., Waller, T., Young, J., Tiwari, A. (2016) “Layout Optimisation of a Repair Facility Using Discrete Event Simulation”, 9th International Conference on Digital Enterprise Technology- DET2016 – “Intelligent Manufacturing in the Knowledge Economy Era”, 29-31 March, Nanjing, China
4. Prajapat, N., Tiwari, A., Gan, X., Ince, N., Hutabarat, W. (2017) “Preventive Maintenance Scheduling Optimisation: A Review of Applications for Power Plants”, Advances in Through-Life Engineering Services, Chapter 24, Accepted for publication
5. Prajapat, N., Tiwari, A. (2016) “A review of assembly optimisation applications using discrete event simulation”, International Journal of Computer Integrated Manufacturing, pp. 215-228, Vol. 30

TABLE OF CONTENTS

ABSTRACT i
ACKNOWLEDGEMENTS iii
LIST OF PUBLICATIONS iv
LIST OF FIGURES ix
LIST OF TABLES xiii
LIST OF ABBREVIATIONS xv
CHAPTER 1 - INTRODUCTION 1
1.1 Discrete Event Simulation 2
1.2 Manufacturing Process Optimisation 3
1.3 Multi-Objective Optimisation 5
1.4 Immersive Visualisation 6
1.5 Towards Zero Prototyping of Factory Layouts and Operations Using Novel Gaming and Immersive Technologies 7
1.6 Problem Statement and Motivation 8
1.7 Thesis Scope 9
1.8 Aim 9
1.9 Objectives 9
1.10 Methodology 10
1.10.1 Research Problem Definition 10
1.10.2 Literature Review 10
1.10.3 Identification of Research Gap 11
1.10.4 Definition of Aim, Objectives and Methodology 11
1.10.5 Framework to Create Ready-to-Optimise DES models 11
1.10.6 Generation of DES models 13
1.10.7 Framework for Interactive Immersive Visualisation 13
1.10.8 Framework for Multi-Objective Optimisation 13
1.10.9 Framework to Visualise Relationships Between DES Variables 13
1.10.10 Validation of Frameworks using Test Cases 14
1.10.11 Conclusions 15
1.11 Research Contributions 15
1.12 Thesis Structure 16
CHAPTER 2 - LITERATURE REVIEW 19
2.1 Introduction to DES 19
2.2 Optimisation of Assembly Systems Using DES 23
2.2.1 Introduction 23
2.2.2 Methodology 24
2.2.3 Problem Domain 25
2.2.4 Objective Functions 30
2.2.5 Constraints 35
2.2.6 Model Formulation 37

2.2.7 Optimisation Methodology 42
2.2.8 Discussion on Optimisation of Assembly Systems Using DES 51
2.3 Visualisation of Manufacturing Processes 54
2.3.1 Introduction 54
2.3.2 Visualisation of Assembly Processes 54
2.4 Discussion on Literature Review 56
2.5 Research Gaps 58
2.6 Industry 4.0 and the Internet of Things 59
2.6.1 Cyber Physical Systems 60
2.6.2 Big Data 60
2.6.3 Smart Factories 61
2.7 Methods and Software 62
2.7.1 Software used 62
2.7.2 Test Cases 63
2.7.3 Modelling Methods 64
2.8 Chapter Summary 66
CHAPTER 3 – DEVELOPMENT OF READY-TO-OPTIMISE DES MODELS 67
3.1 Background 67
3.2 Framework to enable the development of ready-to-optimise DES models 69
3.3 Test Case 1 - Assembly 72
3.3.1 Problem description and process modelling 73
3.3.2 Application of framework to create ready-to-optimise DES model 74
Simulation run parameter selection 80
3.3.3 Test case discussion 85
3.4 Test Case 2 – Repair Facility 87
3.4.1 Problem description 87
3.4.2 Application of framework to create ready-to-optimise DES model 89
Generate interface for input variables 92
Adjust simulation model after testing 100
3.4.3 Test case discussion 103
3.5 Discussion 104
3.6 Chapter Summary 105
CHAPTER 4 – MULTI-OBJECTIVE OPTIMISATION OF DES MODELS 107
4.1 Background 107
4.2 Framework for multi-objective optimisation of DES models 109
4.3 Test Case- optimisation of a turbine assembly 110
4.3.1 Experiment 1 112
4.3.2 Experiment 2 115
4.3.3 Experiment 3 118
4.4 Results and Discussion 122

4.5 Chapter Summary 126
CHAPTER 5 -INTERACTIVE IMMERSIVE VISUALISATION OF DES MODELS 129
5.1 Background 130
5.2 Requirements from an immersive interface 132
5.3 Framework for interactive immersive visualisation of DES models 134
5.4 Test cases- layout reconfiguration and labour adjustment 137
5.4.1 Layout reconfiguration experiment 137
5.4.2 Labour utilisation experiment 139
5.4.3 Results 139
5.4.4 Discussion of Results 142
5.5 Chapter Summary 144
CHAPTER 6 -VISUALISATION TO SUPPORT DECISION MAKING WITH DES MODELS 145
6.1 Introduction 145
6.2 Framework for visualisation to support decision making with DES models 147
6.3 Test Case- A Small Manufacturing System 150
6.3.1 Description of the manufacturing process 150
6.3.2 Application of Framework for Test Case 1 152
6.3.3 Summary of results from Test Case 1 164
6.4 Test Case 2 – Medium value production line 165
6.4.1 Description of the Manufacturing Process 165
6.4.2 Application of the framework for Test Case 2 for two conflicting objectives 168
6.4.3 Application of Random Forests to Test Case 2 175
6.4.4 Summary of Results from Test Case 2 184
6.5 Test Case 3 – Assembly System 186
6.5.1 Regression Tree for ‘Average Assembly Time’ Objective Function 186
6.5.2 Regression Tree for ‘Average Fitters’ Objective Function 192
6.5.3 Random Forest for ‘Average Assembly Time’ Objective Function .. 195
6.5.4 Random Forest for ‘Average Fitters’ Objective Function 198
6.5.5 Summary of Results from Test Case 3 201
6.6 Discussion of Results 203
6.7 Chapter Summary 206
CHAPTER 7 - DISCUSSION AND CONCLUSIONS 207
7.1 Literature Review 207
7.1.1 Observations 207
7.1.2 Research Trends and Research Gaps 208
7.2 Development of ready-to-optimise DES models 210
7.3 Framework for Multi-Objective Optimisation of DES Models 211
7.4 Framework for Interactive Immersive Visualisation of DES Models 213

7.5 Framework for Visualisation and Decision Making using DES Models..214
7.6 Research Limitations 216
7.7 Future Research 217
7.8 Conclusions 218
REFERENCES 221

LIST OF FIGURES

Figure 1-1 - A taxonomy of engineering optimisation problem types (Tiwari et al., 2015) 5

Figure 1-2 - A flowchart to show the research methodology 12

Figure 2-1 - A graph to show the number of papers produced per year group 25

Figure 2-2 - A pie chart to show application domains of papers 26

Figure 2-3 - A bar chart to show the number of papers using each objective function type 30

Figure 2-4 - A pie chart to show the number of papers appyling each optimisation method 43

Figure 3-1 – Framework to enable development of ready-to-optimise DES models 69

Figure 3-2 IDEF3 process model of the turbine assembly 73

Figure 3-3 Decomposition of Task 4 from IDEF3 process model in Figure 3-2 74

Figure 3-4 – A graph to show the average variance of each output over replicates for varying run lengths 81

Figure 3-5 - Graphs to show KPI values (averaged over replicates) for a simulation run of 185 turbine assemblies 83

Figure 3-6 - Graphs to show the results for 2 Design of Experiments with the original NCR formulation (left) and new formulation (right) 85

Figure 3-7 – A diagram of the repair process 88

Figure 3-8 - A diagram to show the location of the 4 quadrants within the factory 95

Figure 3-9- A look up matrix to show the different possible factory movements. Values relate to the list above 96

Figure 3-10- A graph to show the percentage time spent waiting for one task 99

Figure 3-11- Utilisation graphs of the first scenario (above) and second scenario with cycle times reduced by 10% (below) 102

Figure 4-1 – A multi-objective optimisation framework for DES models 109

Figure 4-2 - A diagram to show the optimisation methodology followed for each experiment for the turbine assembly test case 111

Figure 4-3 - Residual plots and histograms for the average assembly time and average fitters for experiment 1 113

Figure 4-4 - A graph to show the meta-model Pareto front and the same points re-run using the actual simulation model 115

Figure 4-5 - Residual plots and histograms for the average assembly time and average fitters for experiment 2 116

Figure 4-6 - A graph to show the comparison between the regression model Pareto fronts and actual results for experiment 1 and experiment 2 117

Figure 4-7 - Residual plots and histograms for the average assembly time and average fitters for experiment 3 119

Figure 4-8 - A graph to show the comparison between the regression model Pareto fronts and actual results for experiment 2 and experiment 3 120

Figure 4-9 - A graph to show the comparison between the regression model Pareto fronts and actual results for experiment 2 and experiment 3 with the results of single objective optimisation for each objective function 121

Figure 4-14 - A graph to highlight the pairs of points which have identical decision variables and are closest to each other with respect to each objective function 125

Figure 5-1 - Proposed framework with Matlab GUI for virtual reality visualisation of DES models 135

Figure 5-2 - Process steps required to carry out the factory reconfiguration .. 138

Figure 5-3 - Simulation run data for the original layout (left) and for the new proposed layout (right) showing the new distance travelled 138

Figure 5-4 - Screenshots of the immersive environment from two locations. Location 1 (a and c) and location 2 (c and d). Screenshots a and b show the original layout; screenshots c and d show the layout after reconfiguration. 140

Figure 5-5 - Process steps required to carry out the labour changes 140

Figure 5-6 - Screenshots of the Matlab UI with the original 20 workers (a) and after a reduction in the number of workers (b) with the labour utilisation for each scenario shown at the bottom of the 'machine data' section 141

Figure 5-7 - A screenshot to show the buffer for a part in the incorrect location (part lies on the red floor area posing a potential collision risk) 141

Figure 5-8 - A screenshot to show the workstation buffer too close to the workstation 142

Figure 6-1 – A diagram to show the process for creating regression trees 148

Figure 6-2 A flowchart to show the regression tree growing algorithm 150

Figure 6-3 - A screenshot of the case study DES model 151

Figure 6-4 – Regression trees created for 6 different outputs 155

Figure 6-5 - Regression trees for ‘LabourIdle’ and ‘LabourBusy’ variables with correlation coefficient -1 157

Figure 6-6 - Regression trees for ‘Machine2WaitLab’ and ‘Machine2Busy’ variables with correlation coefficient 0.93 157

Figure 6-7 - Regression trees for ‘Throughput’ and ‘Machine1Blocked’ variables with correlation coefficient 0.90 158

Figure 6-8 - Regression trees for ‘Machine2Busy’ and ‘Throughput’ variables with correlation coefficient -0.81 158

Figure 6-9 - Regression trees for ‘Machine2WaitLab’ and ‘Machine3Idle’ variables with correlation coefficient 0.79 159

Figure 6-10 – A regression tree for case study 1- for ‘mean time for Part 1’ output variable. 161

Figure 6-11 – A regression tree for Mean Time for part 1 with average values for throughput 163

Figure 6-12 – A screenshot of the WITNESS DES model for case study 2 166

Figure 6-13 – A graph to show the weighted average of WIP values for each part type, reading were taken in 5 simulated minute intervals 169

Figure 6-14 – Regression trees for throughput and man hours per day, correlation coefficient of 0.50 170

Figure 6-15 – A regression tree to show the throughput and ‘man hours per day’ results overlaid 171

Figure 6-16- Regression tree for manhours per day with throughput values overlaid 173

Figure 6-17 – A graph to show the prediction error against the number of regression trees 177

Figure 6-18 – Graphs to show the %incMSE and incNodePurity for each input variable for the throughput objective function 179

Figure 6-19 – A graph to show the observed values against predicted values for test data for the throughput objective function 180

Figure 6-20 – A graph to show the prediction error against the number of classification trees within the random forest for the man hours per day objective function 182

Figure 6-21 - Graphs to show the mean decrease in accuracy and the mean decrease in Gini index for each input variable for the man hours objective function 183

Figure 6-22 - A graph to show the observed values against predicted values for test data for the ‘man hours per day’ objective function 184

Figure 6-23 – Initial regression tree created for ‘average assembly time’ objective function 187

Figure 6-24 – A regression tree for the ‘average assembly time’ with ‘average fitters’ values overlaid and numbered nodes 190

Figure 6-25 - Initial regression tree created for ‘average fitters’ objective function 192

Figure 6-26 - A regression tree for average fitters with ‘average assembly time’ values overlaid and numbered nodes 194

Figure 6-27 – A graph to show the residual error as the number of trees increases 196

Figure 6-28 – Graphs to show the variable importance using % increase in MSE and increase in node purity 197

Figure 6-29 – A graph to show the observed against predicted values for the test data for ‘average assembly time’ objective function 197

Figure 6-30 – A histogram of the residuals for predictions using the random forest on the ‘average assembly time’ objective function 198

Figure 6-31 – A graph to show the convergence of residual error as the number of trees increases 199

Figure 6-32 – Graphs to show the % increase in MSE and increase in node purity for the average fitters objective function 199

Figure 6-33 – A graph to show the observed against predicted values for the ‘average fitters’ objective function 201

Figure 6-34 - A histogram of the residuals for predictions using the random forest on the ‘average fitters’ objective function 201

LIST OF TABLES

Table 3-1 – An example table to show the process of creating cycle times for each turbine assembly task 76

Table 3-2 - A table to show the average variance between replicates for varying run lengths for 3 output variables 81

Table 3-3 - A table to show the input and output data within the Excel spreadsheet for the repair facility test case 90

Table 3-4 - A table to show the cycle times, number of stations and workers for each process. * indicates the worker is shared amongst workstations. 91

Table 3-5- A table to show the set up for experiments- all experiments were run for 25 weeks 101

Table 4-1 – A table to show the run parameters for the three experiments 112

Table 4-2 - A table to show the average variance between replicates for varying run lengths for 3 output variables with probabilistic cycle times 118

Table 4-3 - A table to show the statistical results for regression models for each experiment and each objective 122

Table 4-4 - A table to show the points where the regression model and actual simulation model overlap in objective space 123

Table 4-5 - Tables to show the difference between decision variable values for the regression model and the actual model for point 1 (above) and point 2 (below). 123

Table 5-1 – A table to show the requirements from an immersive interface framework 134

Table 6-1 – A table to show the nomenclature for the decision tree growing algorithm 149

Table 6-2 - A table to show the fixed and variable inputs for case study 1 153

Table 6-3 – A table to show the correlations between output variables for case study 1 156

Table 6-4 – A table to show the pivot table created for case study 1 for average throughput 162

Table 6-5 - A table to show the pivot table created for case study 1 for average throughput 162

Table 6-6 – A table to show the validation of the generated regression tree for 4 different test conditions 164

Table 6-7 - A table to show the daily shift pattern of workers for case study 2 167

Table 6-8 – A table to show the fixed and variable inputs for case study 2 168

Table 6-9 – A table to show the pivot table created for case study 2 for average man hours per day 170

Table 6-10- A table to show the validation of the generated regression tree for 4 different test conditions 172

Table 6-11 - A table to show the pivot table created for case study 2 for average throughput 173

Table 6-12 - A table to show the validation of the generated regression tree for 4 different test conditions 174

Table 6-13 – A table to show the results of variable importance from the random forest for the 'throughput' objective 179

Table 6-14 - A table to show the results of variable importance from the random forest for 'man hours per day' objective function 182

Table 6-15 – A pivot table to show calculation of values for node 7 188

Table 6-16 -–A pivot table to show calculation of values for node 5 188

Table 6-17 - A table to show the validation of the generated regression tree for 4 different test conditions 191

Table 6-18 - A table to show the validation of the generated regression tree for 4 different test conditions 195

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
AOQL	Average Outgoing Quality Limit
API	Application Programming Interface
APOW	Automation and Process Optimisation Workbench
CAD	Computer Aided Design
DE	Differential Evolution
DES	Discrete Event Simulation
DOE	Design of Experiments
DSS	Decision Support System
GA	Genetic Algorithm
GDE3	Generalised Differential Evolution 3 algorithm
HMD	Head Mounted Display
IC	Integrated Circuit
IDEF3	Integrated DEFinition 3 process capture method
ICT	Immersive Computational Technology
IPA	Infinitesimal Perturbation Analysis
jMetal	Java based MEtaheuristic ALgorithms
KPI	Key Performance Indicator
LAPS	Labour Available Per Shift
MS	Microsoft
NCR	Non Conformance Report
NSGAII	Non-dominated Sorting Genetic Algorithm 2
OLH	Optimal Latin Hypercube
OMOPSO	Multi Objective Particle Swarm Optimisation
OTD	On Time Delivery
PAES	Pareto Archived Evolutionary Strategy algorithm
PUDDE	Pareto Utility Discrete Differential Evolution algorithm
PLM	Product Lifecycle Management
QUEST	QUantum Electron Simulation Toolbox
SA	Simulated Annealing
SCM	Standard Clock Method
TCP/IP	Transmission Control Protocol / Internet Protocol
UI	User Interface
UML	Unified Modelling Language
VBA	Visual Basic for Applications
VR	Virtual Reality

WIP	Work In Progress

CHAPTER 1 - INTRODUCTION

In an increasingly competitive global market, companies are progressively seeking to optimise their production processes, with the aim of improving profitability and reducing production costs. This is of particular significance as production costs can often account for a large percentage of the product cost.

The use of Discrete Event Simulation (DES) for simulation of manufacturing processes has risen significantly within recent years, driven by the development of new technologies and the rise of continual process improvement methodologies in manufacturing. DES is a modelling method which enables users to build in complex system logic such as entities, probabilistic variables and interactions. Users can run various scenarios in their model to obtain insights into system behaviour. Significant benefits for manufacturers can be achieved through DES modelling and optimisation including: increased utilisation, increased throughput, reduced production costs and reduced lead times amongst others.

Despite the increasing use of DES to simulate manufacturing processes, there is still a significant challenge associated with engaging stakeholders and communicating DES insights with relevant decision makers. Jahangirian et al., (2010) conclude from their review that DES does not possess high levels of stakeholder engagement as demonstrated in other methods, such as simulation gaming or traffic simulation. Similarly, Negahban & Smith, (2014) determine that more emphasis should be placed on integration of simulation with upper levels of management and control systems to increase stakeholder engagement.

This research details the development of methods to visualise and optimise DES models of manufacturing processes with the view to improve stakeholder engagement and decision making capabilities with DES. This chapter introduces the main concepts of DES, manufacturing process optimisation, multi objective optimisation and immersive visualisation; this is followed by the

research problem, motivation, aim of the research, objectives, research methodology, contributions and finally the thesis structure.

1.1 Discrete Event Simulation

Manufacturing facilities are often difficult to model with analytical models due to their complexity (Koo et al., 2011), hence these models must be created through simulation. Simulation modelling is capable of representing complex systems and constraints from the real world (Korytkowski, Wisniewski, & Rymaszewski, 2013). Among a number of simulation techniques, Jahangirian et al., (2010) find that DES is the most widely used modelling method in manufacturing and business as over 40% of papers applying simulation use DES. The key advantage of DES is that it is capable of dealing with stochastic variables, which can be modelled with probability distributions. This feature renders DES particularly well-suited for modelling of manufacturing facilities.

DES is commonly used for modelling, analysis and improvement of various systems. DES can be used to model a system as it dynamically evolves by a representation in which the state variables change instantaneously at distinct points in time (Law., 2014). Entities within the system are modelled in a simple 2D environment and the system logic, variables and interaction between entities are all added to the model. Typical entities include humans, machines, vehicles and paths; typical variables include cycle times, number of workers and inventory levels.

Another benefit of simulation modelling is the ability to provide multiple scenarios to aid in the decision making process (Lu & Sundaram, 2002). Upon completion and validation of a DES model, experiments can be carried out on the model to analyse system performance. Users can change input variables and simulate each scenario to produce Key Performance Indicators (KPIs). Results can then be analysed to identify the best input variables and to ultimately ascertain the best configuration of the system.

More recently DES has been linked with optimisation algorithms and engines to automate scenario analysis and identify the best solutions. Nowadays most DES packages include an optimisation tool; this indicates the rise in importance of simulation based optimisation. Many factors can be optimised through DES including: throughput, cost, lead time, profit, efficiency and utilisation among others.

Essentially DES is a computationally expensive tool, hence a number of meta modelling techniques have been increasingly applied to provide fast robust insights to assist with decision making processes (Negahban & Smith, 2014). This computational cost is amplified when DES models are linked with multi-objective optimisation algorithms such as GAs. As a result, meta-modelling is a vital area of research as simulation is increasingly used for operational decision making downstream in manufacturing processes.

Other research into improving decision making capabilities of DES models has been proposed in recent times. DES is a decision support tool (Dehghanimohammadabadi and Keyser, 2017) although there are some decision making processes which cannot be carried out within simulation programs such as decision making analysis and computational support tools. Hence, hybrid approaches must be considered. This is a growing area of research.

1.2 Manufacturing Process Optimisation

Traditionally, within industry, product optimisation has been used extensively to produce the most efficient and cost effective products. This is of particular significance for production of high-value, low-volume products. With increasingly competitive markets there has been more emphasis on cost reduction and continuous improvement; therefore, increasing interest in the optimisation of manufacturing processes.

Manufacturing process optimisation can be applied at different levels for various problems. Process optimisation within engineering can be decomposed into four main areas as indicated in Figure 1.1.

- Manufacturing system covers the optimisation of factory configurations, resource allocation and higher level scheduling within the overarching manufacturing system.
- Assembly optimisation covers assembly sequence planning and assembly line balancing which are critical to reduce the time to market for products.
- Multi-machine optimisation refers to the optimal set up and configuration of a sequence of machines which often have a complex network of dependencies.
- Machine optimisation refers to choosing the best settings and configurations for a single machine to optimise production with respect to chosen objective functions.

These various types of process optimisation problems, which occur frequently within manufacturing facilities, can be optimised through modelling, simulation, application of optimisation algorithms and visualisation methods. The results can be implemented within manufacturing facilities to create measurable business benefits.

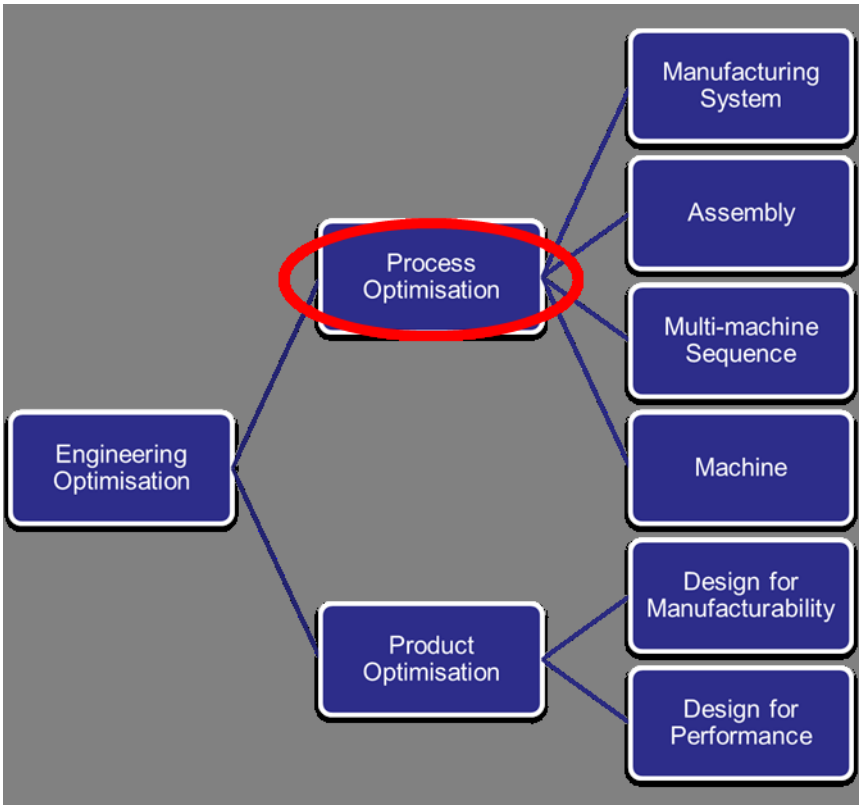


Figure 1-1 - A taxonomy of engineering optimisation problem types (Tiwari et al., 2015)

1.3 Multi-Objective Optimisation

Manufacturing process optimisation problems can be complex with multiple constraints and objectives. Traditionally multi-objective optimisation problems have been transformed to form single objective problems through application of user-defined parameters (Deb, 2001); when optimised this leads to a single optimal solution. Generally expert knowledge is used to create the weighting factors; however, this method does not enable analysis of a variety of other potential solutions.

For manufacturing facilities, there is often a range of production targets including: cost, throughput, utilisation, order fulfilment etc. Hence the optimisation of a manufacturing facility requires optimisation of a set of

objectives within a multidimensional space where traditional methods are not suitable (Neira Dueñas et al., 2007).

Multi-objective optimisation has become increasingly popular in recent times as it used to explicitly search for a range of optimal solutions known as the Pareto front. The Pareto front is a set of solutions which are all as good as each other in the objective function space. The Pareto front is a key feature of multi-objective optimisation, as it enables the user to choose the best solution after taking into account other factors. The front enables decision makers to access more information about various options and their outcomes. This feature could be highly valuable for complex manufacturing process optimisation problems where there are often multiple solutions. Providing manufacturing decision makers with a number of solutions and their respective decision variables will enable deeper understanding of the underlying process and solution space.

Traditional single objective optimisation algorithms are often deterministic and use a single solution which is iteratively updated. Due to the nature of multi-objective optimisation, traditional single objective algorithms cannot be applied. Hence evolutionary computation methods have been developed in recent years to facilitate multi-objective optimisation.

1.4 Immersive Visualisation

In recent years, technological advancements in hardware and software have enabled users to develop 3D immersive environments which provide users with a sense of presence (Oyekan et al., 2015). Immersive visualisation allows users to be placed within a computer generated scene or Virtual Reality (VR) which offers a sense of realism. The immersive visualisation requires the input of 3D models which are used to build the scene. The VR system has a stereoscopic output (Choi and Cheung, 2008) which enables the user to experience immersion. The stereoscopic output can be experienced through a VR headset or a VR cave environment.

Users are can also pair immersive visualisation with haptic devices to enable interaction with various entities within the virtual environment. Immersive visualisation and VR have been applied in various sectors including surgical planning, military training and manufacturing simulation amongst others (Choi and Cheung, 2008).

Within the field of manufacturing, immersive visualisation is often used to produce virtual prototypes of components or environments to assist with planning activities. Immersive visualisation can assist with designing new-build factories and managing operations for existing factories (Oyekan et al., 2015). For instance, immersive visualisation can be used to visualise virtual factories with the aim of clearly seeing the impact of a change in the system (Jahangirian et al., 2010). This immersive visualisation can enable users to detect layout errors and also reduce costly downtime or rework. VR and simulation reduces the requirement for expensive prototyping, minimising waste and ensuring 'right-first-time' is achieved (Choi and Cheung, 2008).

1.5 Towards Zero Prototyping of Factory Layouts and Operations Using Novel Gaming and Immersive Technologies

Within this thesis, the repair facility test case has been developed as part of the 'Towards Zero Prototyping of Factory Layouts and Operations Using Novel Gaming and Immersive Technologies' project which was funded by Innovate UK (project no: 101888).

This project aims to enhance the capabilities of Discrete Event Simulation by enabling immersive visualisation of DES models and a real-time data link with systems modelled using DES. The project aims to use novel gaming technologies, such as the Oculus Rift and the Microsoft Kinect, to enable this enhanced DES experience.

The developed methods will enhance the capabilities of DES by enabling:

- Virtual prototyping of new processes and layouts in a 3D immersive environment
- Collaboration of remote users within an immersive DES model
- Comparison between model and live shop floor data
- Performing scenario analysis from within the immersive DES model
- Visualising the impact of implemented changes in the system

The repair facility test case introduced in Chapter 3 is developed as part of this project. Chapter 5 presents work carried out as part of this project to develop an interface to enable interaction with an immersive DES model to visualise DES models and to analyse and optimise DES models through scenario analysis.

1.6 Problem Statement and Motivation

Within engineering firms, such as General Electric, optimisation has been widely used for design engineering, however much less emphasis has been placed on applying optimisation methodologies to streamline and optimise manufacturing processes. Potential manufacturing process optimisation problems include: assembly optimisation, factory layout optimisation and scheduling optimisation.

The application of optimisation methodologies to manufacturing processes can provide significant cost benefits and other benefits such as improved utilisation of high value assets, improved throughput, reduced manufacturing bottlenecks, and increased labour utilisation etc. These factors are vital as the cost of labour for assembly processes remains high (Lee & Stecke, 1996).

Manufacturing process optimisation could be applied at several different levels, considering a large variety of problems and optimisation objective functions. However, performance objectives for assembly lines for instance, are always competing with each other and cannot be met simultaneously (Zeng, Wong, & Leung, 2012). This leads to the requirement for multi-objective optimisation algorithms to be applied on manufacturing problems to simultaneously optimise multiple objectives.

In addition, there is a requirement to integrate DES with upper levels of management to develop improved stakeholder engagement. In particular, the time consuming and difficult data gathering process often repels stakeholders (Jahangirian et al., 2010). New technologies have been developed in recent years such as virtual simulation which enables users to model systems in a 3D immersive environment (Jahangirian et al., 2010). These methods can be used to improve stakeholder engagement and to clarify the impact of changes on manufacturing systems.

The problem statement is given as follows:

Need for a framework to simulate, optimise and visualise manufacturing processes to assist manufacturing stakeholders with decision making.

1.7 Thesis Scope

This thesis focuses on the simulation, optimisation and visualisation of assembly systems. Assembly systems can be considered a sub-set of manufacturing systems.

The phrases ‘manufacturing systems’ and ‘assembly systems’ are often used interchangeably. However, this thesis is limited to the field of assembly systems and the application of methods to these systems. Maintenance systems and generic manufacturing systems are not within the scope of this thesis.

1.8 Aim

The aim of this research is to develop a framework to simulate, optimise and visualise manufacturing processes to assist stakeholders with decision making.

1.9 Objectives

To facilitate this aim, the research will have the following objectives:

1. Identify typical constraints, objectives and optimisation methods used with manufacturing simulation models.
2. Develop a framework to enable the creation of ready-to-optimise DES models.
3. Develop a framework for multi-objective optimisation of DES models.
4. Develop a framework to enable interactive immersive visualisation of simulation models of manufacturing processes.
5. Develop a framework to enable visualisation of the relationship between input and output variables for manufacturing simulation models.
6. Validate visualisation and optimisation methods using a set of real life and generated test cases.

1.10 Methodology

This section details the methodology which will be followed to achieve the objectives outlined in the previous subsection. Figure 1-2 is a flowchart demonstrating the research process plan which links the objectives from the previous section. Each of the components within the flowchart is summarised in the following sections.

1.10.1 Research Problem Definition

The research within this thesis covers the application of visualisation and optimisation methods to DES models of assembly systems, with the view to improve the decision making capabilities of DES models for stakeholders. The scope for this thesis has been defined in section 1.7. The focus of the thesis is on the simulation, optimisation and visualisation of assembly systems.

1.10.2 Literature Review

This literature review fulfils the first research objective: to identify typical constraints, objectives and optimisation methods used with manufacturing simulation models.

The literature review has been carried out in Chapter 2 and includes two parts: a review covering DES models of assembly systems and a review covering the visualisation methods applied to manufacturing processes. The literature review in the area of DES for assembly systems has been carried out to identify the key trends within research in the area of DES modelling for assembly systems. Searches have been made within Scopus to identify relevant papers. Data from the papers has been collated into an Excel spreadsheet and used to identify key research trends and any research gaps. The second part of the literature review has been used to identify research trends and gaps in the area of visualisation of manufacturing processes.

1.10.3 Identification of Research Gap

Following the literature review, the research gap has been identified and summarised in Chapter 2.

1.10.4 Definition of Aim, Objectives and Methodology

The research aim has been developed using the research problem. The research gap and research trends have been used to develop the research objectives, in order to ensure that the research undertaken covers novel areas. The research objectives include the development of four frameworks to enable users to link DES with visualisation and optimisation methods. The research methodology and the flowchart in Figure 1-2 have been generated using the research objectives.

1.10.5 Framework to Create Ready-to-Optimise DES models

A framework is presented in Chapter 3 which enables the development of ready-to-optimise DES models. The framework defines the processes which should be carried out to build and integrate optimisation features into DES models whilst they are constructed. This framework is subsequently applied to test cases. The generation of this framework fulfils the second research objective: to develop a framework to enable the creation of ready-to-optimise DES models.

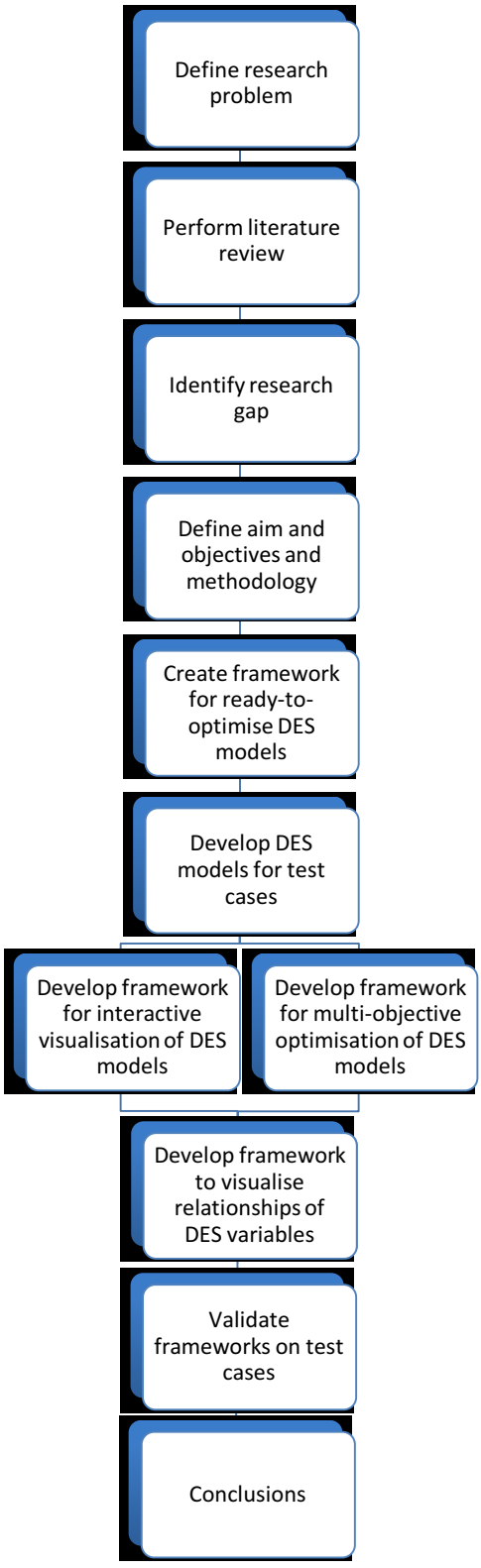


Figure 1-2 - A flowchart to show the research methodology

1.10.6 Generation of DES models

The framework presented in Chapter 3 is used to generate ready-to-optimise DES models of real life assembly systems. Two different test cases are identified and modelled using DES software WITNESS, with the view to optimise the models with respect to chosen objective functions. The test cases used within this thesis are detailed within 1.10.10.

1.10.7 Framework for Interactive Immersive Visualisation

A framework for enabling interactive immersive visualisation of DES models is presented in Chapter 5. An interactive interface is developed to improve stakeholder engagement with DES models and assist users with decision making. A test case is used to develop the requirements for the interactive interface. The framework is then developed and tested on a DES model for two different test cases. The generated framework fulfils objective 3: to develop a framework to enable interactive immersive visualisation of simulation models of manufacturing processes.

1.10.8 Framework for Multi-Objective Optimisation

A framework has been developed for enabling multi-objective optimisation of DES models in Chapter 4. The framework is a meta-modelling approach, which has been developed to relieve the computational cost of multi-objective optimisation directly on the simulation model. The framework includes the generation of a Design of Experiments (DOE) with replications, the generation of meta-models of each objective function and the simultaneous optimisation of multiple objectives. The framework is applied on three different experiments and the results are demonstrated and discussed. The framework fulfils objective 4: to generate a framework to enable multi-objective optimisation of DES models.

1.10.9 Framework to Visualise Relationships Between DES Variables

A methodology to enable visualisation of the relationship between input and output decision variables for manufacturing simulation models will be developed in Chapter 6. The methodology will include the generation of decision trees for

visualising the impact of decisions on multiple objectives. In addition, random forests will also be used to improve prediction accuracy for decision makers. This framework fulfils objective 4: to develop a framework to enable visualisation of the relationship between the input and output variables for manufacturing simulation models.

1.10.10 Validation of Frameworks using Test Cases

In order to demonstrate the four different frameworks, a number of real life and fictional test cases have been developed. The test cases represent a variety of different manufacturing processes which require optimisation such as factory layout optimisation, assembly process optimisation and resource assignment optimisation. Each of the test cases has been modelled using DES software WITNESS. In order to carry out experimentation and optimisation on the test cases, the WITNESS models have been integrated with a GE in-house optimisation tool – APOW. The test cases are summarised as follows:

The turbine assembly process optimisation test case is a real life test case for a high-value low-volume production line. The problem aims to efficiently allocate resources to the turbine assembly in order to reduce the lead time, without incurring excessive labour cost. This test case is introduced in Chapter 3.

The repair facility test case is a factory layout optimisation problem. The existing facility is to be rearranged for a future maintenance regime. The repair facility has a number of workstations, cranes and resources which need to be considered when carrying out the rearranging of the factory configuration. The configuration should aim to maximise utilisation and ensure the required demand is met. This test case is introduced in Chapter 3.

Case study 1 is a small manufacturing system which has been created to represent the key features of a manufacturing system on a small scale. The model includes: manpower, transportation resources, variable arrival times and buffers. The model has been developed to test the decision tree creation framework introduced in Chapter 6.

Case study 2 has been developed as a medium size production line which has conflicting objectives. This test case has been developed as an alternative medium volume manufacturing production line. This production line has 8 machines and 8 buffers with no materials handling equipment required for production. The labour is shared amongst workstations. The model has been developed to test the decision tree creation framework introduced in Chapter 6.

1.10.11 Conclusions

Following the generation of frameworks in Chapters 3-6, the conclusions will be presented in Chapter 7. This will include the contributions to knowledge, research limitations and the attainment of the research aims and objectives. Future research areas are also discussed.

1.11 Research Contributions

A number of different contributions to research have been made within this thesis in the area of DES, optimisation and visualisation for improving decision making. The main research contributions are as follows:

- A number of observations have been noted regarding requirements for enabling optimisation of DES models. These observations have been used to generate a framework to develop ready-to-optimise DES models. The framework includes: generation of an interface for input and output variables, simulation run parameter selection and model modifications. Two different test cases have been identified and the framework has been applied to each test case.
- A framework has been proposed to enable multi-objective optimisation of DES models. In particular, the framework involves the generation of multiple meta-models to link with a multi-objective optimisation algorithm. This framework reduces the time required to carry out direct multi-objective optimisation. The results demonstrate that the method can be used to drive a search into the Pareto front region of the search space.

Results demonstrate the cost function meta-model provides better prediction than the assembly time meta-model.

- A lack of research in scenario analysis of DES models from an within an immersive environment has been noted. A framework has been proposed with a closed feedback loop to enable users to carry out specific tasks within an immersive environment. This framework is validated using two different experiments: the layout reconfiguration case and the labour utilisation case. The framework can be used to help increase stakeholder collaboration and engagement and to assist with decision making. The system has already been used to identify potential flaws with new layouts of the repair facility test case.
- A lack of research in the area of data mining techniques used to visualise the impact of decisions on DES models of assembly systems was noted. A framework has been proposed to enable visualisation of the impact of different decisions on output variables using regression trees. The framework enables multiple objectives to be visualised on the same regression tree. The regression trees are found to be poor for predicting new data points, hence random forests have been proposed for variable importance visualisation and prediction of new points. The framework is validated on two generated test cases and a real life turbine assembly test case with accurate prediction of variable importance and acceptable residual errors.

1.12 Thesis Structure

This thesis is structured in eight chapters.

Chapter 1 provides an overview of manufacturing process optimisation, DES, multi objective optimisation, the research motivation, problem statement, aims, objectives, methodology and research contributions.

Chapter 2 provides a comprehensive literature review in the areas of optimisation of manufacturing processes and visualisation of manufacturing

processes. Research gaps and trends are identified for these areas and summarised. An overview of Industry 4.0 and the Internet of Things (IoT) is provided. Methods and software used within this thesis are also summarised.

Chapter 3 presents a framework to develop ready-to-optimise DES models. The framework is subsequently applied to two different test cases. Limitations of the framework are discussed.

Chapter 4 presents an optimisation framework to enable multi-objective optimisation of DES models. The framework is demonstrated on a test case for 3 different examples.

Chapter 5 outlines the work carried out combining DES and visualisation methods; this work was carried out as part of an Innovate UK funded project titled ’Towards Zero Prototyping of Factory Layout and Operations Using Novel Gaming Technologies’ in collaboration with (General Electric (formerly Alstom Power)), Cranfield University and Lanner group. This chapter proposes a framework for enabling immersive interaction with 3D DES models. The framework is demonstrated on a DES model for 2 different cases.

Chapter 6 presents a methodology for visualising the relationships between input and output results from DES experiments using regression trees and random forests. The methodology is demonstrated on 3 different examples, two generated test cases and a real life test case, and subsequently limitations are discussed.

Chapter 7 presents the discussion and conclusions from the research. In particular, the research gaps and trends, limitations and areas for future development are outlined.

CHAPTER 2 - LITERATURE REVIEW

This chapter presents a comprehensive literature review covering research in DES, optimisation and visualisation. The aim of this chapter is to provide an overview of modelling, optimisation and visualisation of assembly systems. A variety of key features are extracted from each paper, including: application domains, modelling method, optimisation objectives and optimisation constraints. Other manufacturing systems are not covered in this literature review as they are not within the scope of this thesis.

The chapter is presented in various sections. The first section provides an introduction to DES and includes benefits of DES and application areas. The second section covers literature on application of DES modelling and optimisation of assembly systems. The third section covers the use of visualisation methods for manufacturing processes. Discussion on the literature review is presented. The subsequent sections outline the trends and research gaps identified by the literature review in these main areas. A summary of Industry 4.0 and the Internet of Things (IoT) is presented. Subsequently software and methods used within this thesis are presented.

2.1 Introduction to DES

Modelling methods are widely used across a range of industries to imitate real world systems and processes. Models can then be used to gain insights into how the process or system performs under various situations. Users can then reconfigure and optimise their systems according to their chosen objectives. A system comprises of entities and variables. The entities in a system can be people, machines or objects. The variables within a system are the values which relate to the system entities and which determine the state of the system at a given point in time.

Various modelling methods exist which are suited to different types of real world systems and processes; for instance, simple systems can be modelled with analytical solutions such as calculus or algebra. Many real world systems however are complex and dynamic, as the state of a system changes over time. Systems in which the variables change at distinct points in time (events) are discrete systems, whereas systems in which variables change continuously over time are continuous systems. In addition, the states of discrete and continuous systems may depend on probabilistic variables. For complex discrete systems with probabilistic parameters, simulation modelling is an ideal solution as simulation modelling is capable of representing complex systems and constraints from the real world (Korytkowski, Wisniewski, & Rymaszewski, 2013).

In particular, Discrete Event Simulation is a modelling method that can be applied to complex discrete systems with probabilistic parameters. Discrete event and state-based simulations tend to be convenient for detailed process analyses and perspectives, resource utilization, queueing and relatively shorter-term analyses (Kellner et al., 1999). DES can be used to create entities, variables and logic which can describe the behaviour of the complex system. DES can be applied to a range of different problems from different industries, these include:

- Designing and analysing manufacturing systems
- Evaluating hardware and software requirements for a computer system
- Evaluating a new military weapons system or tactic
- Determining ordering policies for an inventory system
- Designing communications systems and message protocols for them
- Designing and operating transportation facilities such as freeways, airports, subways, or ports
- Evaluating designs for service organizations such as hospitals, post offices, or fast food restaurants
- Analysing financial or economic systems amongst others (Law et al., 2014).

In order to create a DES model, a number of assumptions are often made about the real life system being modelled. These assumptions are made to simplify the modelling process and reduce model complexity. Model assumptions must be validated to ensure that the DES model is also a valid representation of the real life system before any conclusions can be drawn from the simulation model.

The main components of a DES include: simulation clock, entities, variables, probabilistic parameters and system logic. The simulation clock is a key aspect of DES as it tracks the time within the simulation and is used to trigger events which occur within the simulation. The entities within the simulation model can represent a variety of physical and conceptual objects which are used within the system; typical entities include: people, machines, parts, cranes, transportation equipment, buffers, orders and paths amongst many others. Variables within a DES include input and outputs; typical variables can include: inventory levels, time taken for tasks, distances between areas, throughput, production cost, parts completed, number of workers and utilisation amongst others. Probabilistic parameters can also be used within DES models; probability distributions can be used to model these parameters such as triangular distribution, uniform distribution and Weibull distribution amongst others. System logic is also input into the simulation model to describe the system behaviour; this information includes how entities go through the system and how they interact with each other. For instance, a part or order may go through a series of processes within a system in a specific order; this logic must be input into a DES model.

DES models can be built programmatically, however over recent years a number of commercial software have become available which make the task of creating a DES model much simpler (Law et al., 2014). Various DES software has been used including: QUEST (Lee et al., 2009), WITNESS (Ghani et al., 2012), ProModel (Marsh et al., 2010) and Arena (Ferreira et al., 2012) amongst others. Commercial DES software often prepopulates a range of icons for typical entities which may be used in a DES model to quickly generate models.

In addition, commercial DES software may include icons for people, vehicles, buffers, shift patterns and other entities. Commercial DES software may also provide functionality such as: linking with other software, optimisation capabilities and linking to 3D visualisation.

Manufacturing systems are well suited to modelling using DES as they are complex and often have probabilistic variables. Koo et al., (2011) note that manufacturing facilities are often difficult to model with analytical models due to their complexity. Among a number of simulation techniques, Jahangirian et al., (2010) find that DES is the most widely used modelling method in manufacturing and business as over 40% of papers in the review which apply simulation use DES. They find that DES is used for a range of different application areas including: scheduling, process engineering manufacture, inventory management, assembly line balancing, capacity planning, transportation management, process engineering service, production planning and control, supply chain management and project management amongst other areas. Many of these applications are within a variety of different types of manufacturing and assembly systems. These application areas can be further broken down into specific problem types. The various problem types found within literature include:

- Design and balance of assembly lines,
- Uncertainty associated with capacity planning,
- Sequencing the expansion of current resources/ capacity,
- Development of equipment and production schedules,
- Matching labour to demand,
- Order release scheduling,
- Process design and improvement,
- Design of a new facility and
- Instability in supply chain amongst others.

2.2 Optimisation of Assembly Systems Using DES

This subsection reviews literature on the application of DES and optimisation methods to assembly systems. A number of relevant papers are collated and relevant data has been extracted from the papers. Data from papers is classified based on application domain, optimisation objectives, constraints, model formulations and optimisation methods. This classification enables the identification of key trends within the research.

2.2.1 Introduction

In recent times manufacturing environments have become data rich, providing researchers with an opportunity to model and simulate the performance of manufacturing systems (Yang, Bukkapatnam, & Barajas, 2013). Automotive, electronics and other general production industries have identified simulation modelling as a method to analyse and improve their manufacturing facilities.

DES in particular has been widely applied to model and optimise complex manufacturing systems and assembly lines. DES is particularly well suited for modelling manufacturing systems as it can explicitly model the variation within manufacturing systems using probability distributions. DES is thus capable of answering key operational questions relating to throughput, resource allocation, utilisation and supply and demand. This information has enabled manufacturers to implement significant and beneficial modifications to their facilities to improve operations in order to meet production targets.

Various literature reviews already exist in the area of DES modelling for manufacturing systems; a few of the most relevant review papers were analysed as follows. Nikoukaran & Paul, (1999) present a review of software selection methods for manufacturing. The authors cite 42 papers and present software evaluation techniques, criteria for software selection and guidelines. Smith, (2003) presents a comprehensive review of 212 papers in the area of simulation for manufacturing design and operation. Smith (2003) then classifies the papers based on manufacturing system design, system operation and

language or package development. Jahangirian et al., (2010) present a systematic review of a wide range of literature on simulation applications from between 1997 to 2006. The paper provides a much broader picture of the range of applied simulation methods used in manufacturing and business. They conclude that hybrid modelling is an area with a rising profile. More recently Negahban & Smith, (2014) performed a review of 290 papers following the classification strategy of Smith, (2003). They identify a new category, simulation optimisation and meta-modelling. They note this provides an area for potential future research. Alrabghi & Tiwari, (2015) provide an up to date review of simulation optimisation for maintenance systems. Sahu and Pradhan (2016) review assembly line balancing methods in order to find the best solutions for optimising various parameters.

This work however investigates the more specific area of DES with optimisation methods applied to assembly systems.

2.2.2 Methodology

The database used to select papers for this review was Scopus. A search was carried out using the keywords '(discrete event simulation) AND (assembly or manufacturing) AND (optimisation or optimisation)'. The search returned 368 results; papers which were not assembly related were filtered out of these results. This search method was used due to 'manufacturing' and 'assembly' often being used interchangeably within the literature.

Another further search was made in Scopus using the keywords "discrete event simulation manufacturing assembly" returning 149 results. This was carried out to find papers which didn't explicitly specify optimisation as a method in the paper. Papers which were not related to assembly systems and optimisation were filtered out. Following both searches 52 papers were chosen for this review.

Information from the papers has been collated into a MS Excel spreadsheet to identify trends in literature and research gaps. Columns within the spreadsheet include: year of paper, paper type (conference or journal paper), application

area, simulation software used, model formulation and objective functions among others. From the column 'year' it was seen that the application of DES to assembly systems has received much more interest since 2001; only 6 of the 52 papers studied were from before 1996. The bar chart in Figure 2-1 illustrates this growing interest.

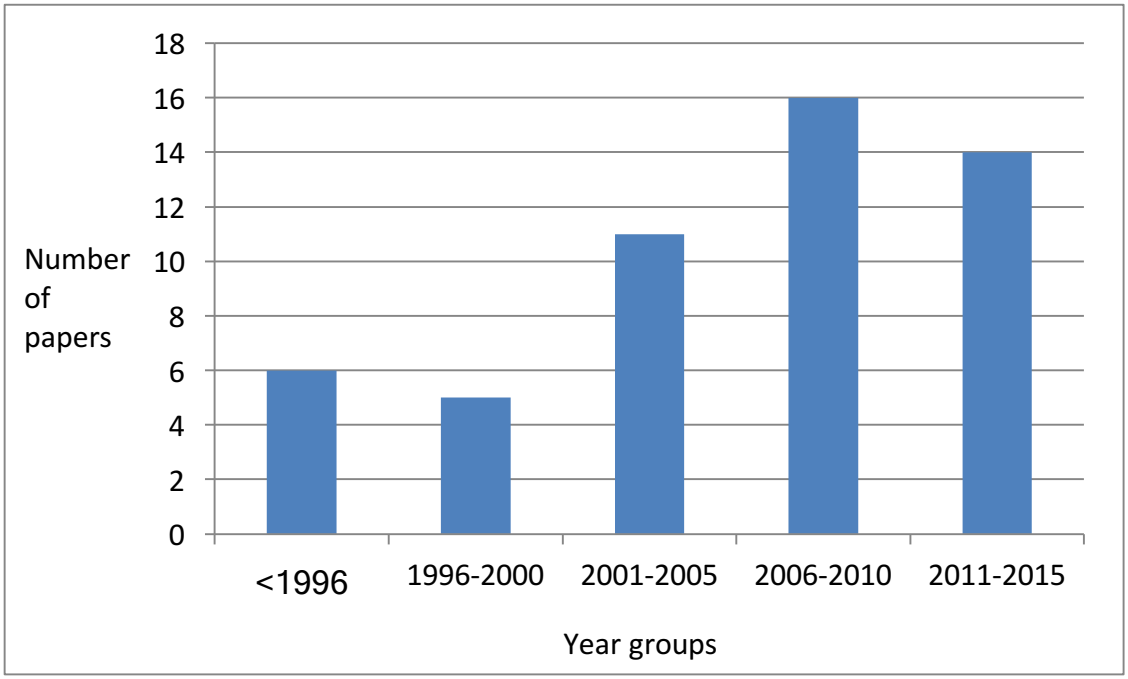


Figure 2-1 - A graph to show the number of papers produced per year group

2.2.3 Problem Domain

The problem domains of papers studied cover a range of industrial application areas: from small electronics assemblies (Li, Nagarur, & Srihari, 2011) to large assemblies such as for the Boeing 747 final assembly (Lu & Sundaram, 2002). The application domains for the papers in this review have been categorised as follows: automotive industry, electronics industry, other production systems and other applications. The most common application domain was other production systems; over a third of all papers have been classified as other production

systems as shown in Figure 2-2. The following subsections detail the different problem domains of the papers within this review.

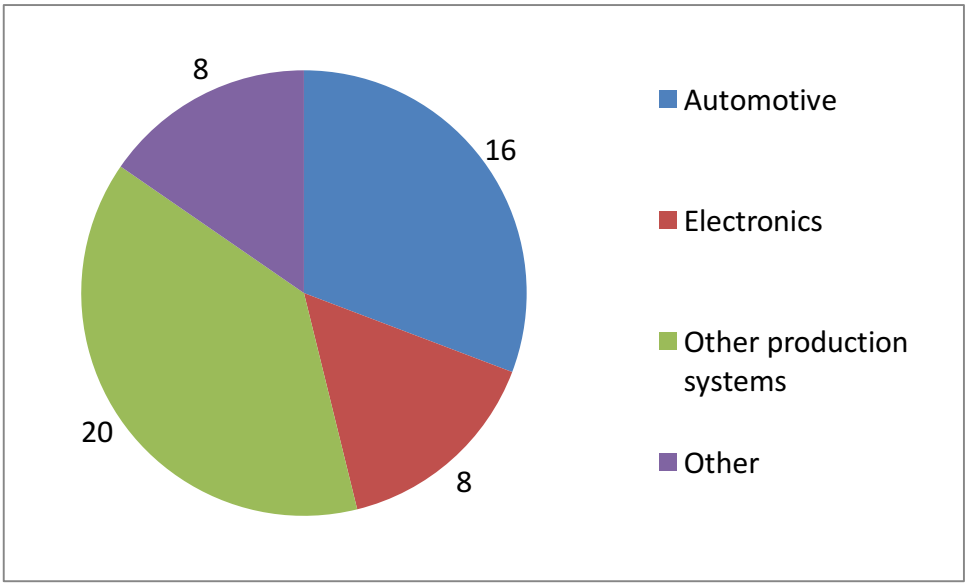


Figure 2-2 - A pie chart to show application domains of papers

2.2.3.1 Automotive Industry

Yang et al., (2013) note that the automotive manufacturing environment produces a large amount of data; this provides an ideal opportunity for modelling of these systems using simulation methods. Authors of research papers within the automotive industry have leveraged the ability of DES to deal with variation, which is inherent within automotive production systems.

Ghani, Monfared, & Harrison, (2012) use DES to model machining processes in order to monitor energy consumption and they test the method on an assembly production line for an automobile engine block. McDonald, Van Aken, & Ellib, (2012) study motor manufacturing for varying demand levels. Baines, Hadfield, Mason, & Ladbrook, (2003) perform analysis on worker variation in an assembly line in an automotive plant. Manns & Elmaraghy, (2009) consider an automotive manufacturing system with main and side loops and they analyse the different inter arrival times of components.

Studies have been carried out in various parts of the automotive assembly process. Yu et al., (2006) compare sun roof fitting for automobiles using manual

and lift assisted methods. Similarly Chandra, Huang, & Kumar, (2003) analyse the roof and chassis assembly process in an automated car assembly line using LEGO blocks. Martinelli & Valigi, (1999) analyse the car dashboard assembly process, which is controlled by a Kanban system. Aguirre et al., (2008) analyse the problem of part manufacturing for the internal combustion engine and Yang et al., (2013) focus on the powertrain manufacturing process.

The control aspects of automobile assembly have been widely considered for optimisation. Ferreira et al., (2012) develop a Decision Support System (DSS) for a four closed loop automobile assembly and preassembly lines; this is noted to be a typical network configuration within the automotive sector. Steinemann et al., (2013) model the downstream assembly processes for automotive manufacturing to assist the line manager with decision making. Otamendi, (2011) analyses different cell designs to incorporate new tasks into assembly of automobiles. McHaney & Douglas, (1997) and Zhao et al., (2010) both consider vehicle assembly and the associated material handling processes.

2.2.3.2 Electronics Industry

Electronics manufacturing is another area in which a considerable number of papers were found. In particular electronics manufacturing often comprises of mixed model assembly lines incorporating high product variability, this variability renders DES an ideal modelling method for electronics industry applications.

Khalil & Stockton, (2010) model a low volume mixed model electronics production environment. DeJong, (2001) considers semiconductor manufacturing with different product mix. Mendes et al., (2005) balance a mixed model PC camera assembly line in order to smooth the workload balance between and within workstations. Li et al., (2011) analyse the printed circuit board electronics assembly process and they build a model incorporating the different cycle times for different products. Pradhan & Damodaran, (2009) look at Integrated Circuit (IC) manufacturing for optoelectronics assembly with potential failures during testing stages followed by rerouting.

Other features of electronics assembly have been modelled. Stratman, Roth, & Gilland, (2004) model the workforce skill level for an electronics manufacturer for the manual assembly of electronic components, whereas Brennan & Rogers, (1995) use DES to design a partially automated assembly line. Li, Nikolaidis, & Mourelatos, (2011) apply their methodology to assist with understanding the effect of changes on the electronics parts assembly system without using Monte Carlo Simulation methods.

2.2.3.3 Other Production Systems

The most common application area was general production or manufacturing systems. In particular authors often model flexible production systems and then analyse the system for different production scenarios.

Various authors have modelled their flexible assembly systems using DES. Iwata & Oba, (1984) build a simulation engine for a flexible manufacturing system and similarly Cave, Nahavandi, & Creighton, (2006) consider a stochastic mixed model assembly system. Zeng et al., (2012) model the labour intensive and complicated apparel sewing manufacturing assembly line; this line is required to be very flexible. They solve the operator allocation problem for this line. Bulgak & Sanders, (1991) also consider flexible assembly systems.

The control aspects of production assembly have been modelled by a number of authors. Koulouriotis, Xanthopoulos, & Tourassis, (2010) model a four stage manufacturing line and examine the production control, they note this is vital for maintaining competitiveness. Ismail et al., (2011) consider a high volume assembly line for stationary manufacturing and they model the control and physical aspects of the system. Kibira & McLean, (2002) also visually model the assembly tasks for workers for a tool production environment. Korytkowski et al., (2013) optimise the dispatching rules for a printing production system.

The resource management and material handling have been considered by a number of authors. Hao & Shen, (2008) model the production and material handling process for an assembly line in a production environment and Mehrsai

et al., (2013a) model the logistics for production and then optimise parameters for autonomous objects. Quite differently Proth et al., (1997) consider the supply chain for an assembly system with inventory control. De Kok & Visschers, (1999) analyse the different allocation policies for common components. Neira Dueñas et al., (2007) analyse and optimise control variables for the refrigeration compressor production line. Desmet, Aghezzaf, & Vanmaele, (2010) analyse inventory for an assembly system. Kouikoglou, (2000) considers resource management for a production network with machines for assembly and disassembly operations.

The design of production systems has also been analysed within studies. Wischnewski & Freund, (2004) present a solution for modelling of transportation systems for production environments. Bulgak & Sanders, (1991), Liu & Sanders, (1989) and Liu & Sanders, (1988) analyse different designs for asynchronous flexible assembly systems. In later work Asil Bulgak, (1992) analyses and designs automatic assembly systems and subsequently analyses the trade-off between quality and productivity.

2.2.3.4 Other Application Areas

A range of other application domains have been considered in a number of research papers, particularly in transportation areas. Lee, Shin, & Ryu, (2009) consider a panel block assembly shop for a shipyard, while Lu, Petersen, & Storch, (2007) model the boiler assembly process for ships. Marsh et al., (2010) consider a bicycle assembly problem. Lu & Sundaram, (2002) make the airplane assembly process a moving assembly line. Cochran & Kaylani, (2008) optimise the design of a tube shop of an aerospace manufacturer. Hugan, (2001) uses DES to model traffic at an automobile plant. Other application domains include configurations for a computer keyboard assembly cell (Spedding et al., 1998) and large scale modular construction of an industrial plant (Taghaddos et al., 2010).

2.2.4 Objective Functions

Among the papers considered in this review there were a range of optimisation objective functions specified by authors. These objective functions have been subcategorised as follows: time based, cost based, bottleneck reduction, throughput, resource management, utilisation and other objectives.

Figure 2-3 shows the number of papers which consider each type of objective function; in addition twenty papers considered more than one objective function. Some overlap has been noticed between various optimisation objectives; for instance bottleneck reduction will automatically increase the throughput, and the increase in throughput will lead to an increase in sales. Fundamentally many of the objective functions can be interpreted as cost objectives, as they essentially aim to increase the production rate in order to produce more products; this drives sales and increases profit. The following subsections provide more details on each objective function category.

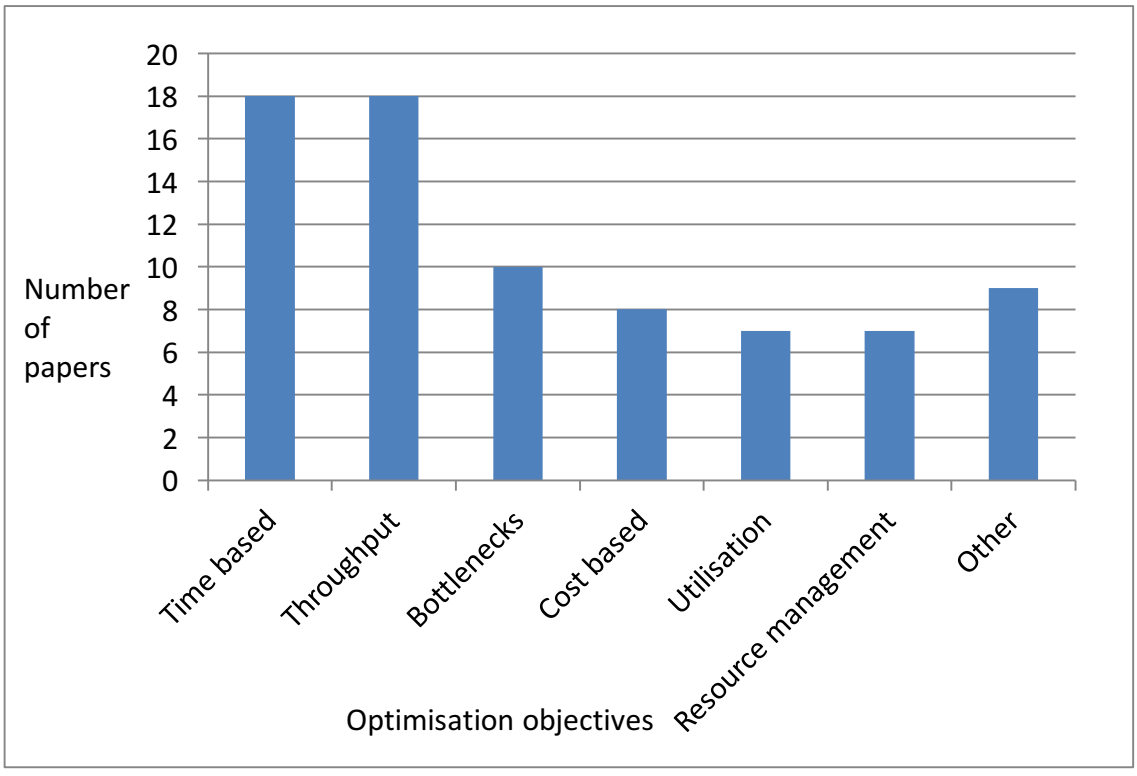


Figure 2-3 - A bar chart to show the number of papers using each objective function type

2.2.4.1 Time Based

Time based objectives were the most common amongst all papers studied along with throughput objective functions. This is expected as within competitive manufacturing environments focus is being increasingly placed on implementation of lean initiatives (Steinemann et al., 2013), shortening lead times (Kibira & McLean, 2002) and meeting delivery to customers (Li et al., 2011; Proth et al., 1997).

Flow time is defined as the time a job spends in the system; this objective function has been used in several studies. Mehrsai et al., (2011) and Mehrsai, Thoben, & Karimi, (2013) consider flow time and make span (the time difference between start and finish of sequence of tasks) as objectives for minimisation. Korytkowski et al., (2013) also consider the mean and maximum flow time, but they also introduce mean and maximum tardiness with the aim to minimise these four objectives. Zeng et al., (2012) consider minimising the make span and maximum flow time but also add minimising the maximum idle time (time when machines have no job to process) as another time based objective function. Lu & Sundaram, (2002), Mendes et al., (2005) and Pradhan & Damodaran, (2009) also consider flow time as an objective function.

Li et al., (2011) consider the cycle time variation as the product design changes and they note that most of the machines in the assembly line are controlled with a standard operation time. Kibira & McLean, (2002) also consider cycle time and they aim to predict machine cycle time using their modelling method. Desmet et al., (2010), Lee et al., (2009), Lu et al., (2007) and Marsh et al., (2010) consider lead time reduction as an objective and Cochran & Kaylani, (2008) implement lead time reduction through a cost function. DeJong, (2001) considers on-time delivery performance and Li et al., (2011) aim to minimise the waiting time of parts.

2.2.4.2 Throughput

An equally popular optimisation objective function considered among papers is throughput. The average throughput can be defined as the number of final products delivered per time unit (Koulouriotis et al., 2010). This objective can be linked to profit and cost, as an increase in throughput increases the number of sales; it can also be classified as a time based objective by definition.

Ferreira et al., (2012), Han et al., (2003), Kouikoglou, (2000), Koulouriotis et al., (2010), Manns & Elmaraghy, (2009), Martinelli & Valigi, (1999), McHaney & Douglas, (1997), Mendes et al., (2005) and Spedding et al., (1998) all identify the throughput to be an objective function, with many authors considering it to be the primary objective. Yu et al., (2006) and Neira Dueñas et al., (2007) identify productivity as an objective, which is essentially the machine throughput. Ferreira et al., (2012) consider the number of cars produced per hour. Aguirre et al., (2008) consider production volume and likewise Asil Bulgak, (1992), Bulgak & Sanders, (1991) and Liu & Sanders, (1988) consider the expected production rate to be the optimisation objective. Yang et al., (2013) calculate throughput for both their simulation model and their System Dynamics model and they use this performance indicator to perform model validation.

2.2.4.3 Bottleneck Reduction

Bottleneck reduction is another objective function which is common amongst a number of papers studied. Following the modelling of a manufacturing facility or assembly line this is a clear choice of an objective for simulation modellers. This objective function will enable improvements in throughput and utilisation.

Work In Progress (WIP) is the number of partially finished goods or orders within a given manufacturing system. Aguirre et al., (2008), Koulouriotis et al., (2010), Li et al., (2011), Otamendi, (2011), Pradhan & Damodaran, (2009), Yang et al., (2013) and Zeng et al., (2012) all aim to minimise WIP. Intuitively

this number increases when there are system bottlenecks; hence reduction in WIP has been classified as system bottleneck reduction in this work. This is agreed by McDonald et al., (2012) who identify potential bottlenecks in their system and note this can reduce performance indicators such as WIP. Hugan, (2001) considers alleviating congestion for a truck plant; this is a similar form of bottleneck reduction. Khalil & Stockton, (2010) and Neira Dueñas et al., (2007) consider reducing blockages; this is essentially a form of bottleneck reduction. Steinemann et al., (2013) consider overcoming bottlenecks through available measures such as re-sequencing or adding additional labourers.

2.2.4.4 Cost Based

Cost based objectives are also popular among the papers considered, as these objectives will enable manufacturers to maintain a competitive advantage through cost reduction (Proth et al., 1997). Cost based functions can be based on minimising cost or maximising profit.

A number of authors consider reducing the costs of production. Marsh et al., (2010) consider the costs of different assembly sequences. Proth et al., (1997) aim to minimise the sum of backlog and inventory costs while dealing with stochastic demand and yield times. Stratman et al., (2004) consider production and quality costs, assessing labour, inspection, testing and defect costs. Lu & Sundaram, (2002) also consider resource costs, but they also add minimising materials, which is not considered by Stratman et al., (2004). Cochran & Kaylani, (2008) optimise designs with respect to a cost function; this function also incorporates lead time reduction.

On the other hand optimisation with respect to profit has been carried out by Koulouriotis et al., (2010); they consider profit which is formulated based on operation per time unit and also profit from product sale.

2.2.4.5 Utilisation

The utilisation of workers and machinery has also been deemed to be an important objective by a number of authors. Cave et al., (2006) and Mehrsai et al., (2013b) use the maximisation of machine utilisation as an objective function. Aguirre et al., (2008) consider utilisation of workers and workstations, whereas Yu et al., (2006) only maximise utilisation of workers. Hugan, (2001) aims to maximise the utilisation of an area for a truck plant and Taghaddos et al., (2010) maximise utilisation of yard resources. Utilisation factors have also been considered as objectives by Neira Dueñas et al., (2007).

2.2.4.6 Resource Management

Another objective used in a number of studies is the management of resources. This objective is linked with cost minimisation as bringing in more labour and materials will incur a cost for the manufacturer.

Zeng et al., (2012) consider minimising the maximum useless productivity of labourers. Mendes et al., (2005) also consider resource utilisation as an objective with the aim that the decision maker can use this information to fine tune the configurations of the line. Lu et al., (2007) minimise materials and resource costs and Steinemann et al., (2013) consider resource optimisation. Aguirre et al., (2008) consider average in process inventory to be a Key Performance Indicator. Ismail et al., (2011) reduce the manpower requirement through system automation and similarly Yu et al., (2006) consider the productivity of manual against assisted fitting operations and conclude that manual operation increases productivity.

2.2.4.7 Other Objective Functions

A number of alternative objectives have been proposed in papers; these include quality, efficiency and availability among others.

Zeng et al., (2012) consider minimising the allocation scheme complexity along with their time based objectives. Ferreira et al., (2012a) and Ferreira et al., (2012b) consider maximising the performance and system availability and similarly Lu & Sundaram, (2002) considers maximising the efficiency. Ghani et al., (2012) aim to optimise for energy consumption and they note that reducing the cycle time has an exponential impact on the energy consumption. Asil Bulgak, (1992) also considers product quality as an objective and he considers the trade-off between quality and production rate. Taghaddos et al., (2010) use the welfare of society as an objective. Wischnewski & Freund, (2004) reduce risk through modelling and visualisation.

2.2.5 Constraints

Authors have applied various constraints to their model formulations within this literature review. These constraints include: volume constraints, routing constraints, configuration constraints and resource constraints, amongst others. The following subsections summarise the types of constraints which have been applied by authors.

2.2.5.1 Volume Constraints

Volume constraints are considered to be highly important in a number of studies and include: production demand, WIP, batch size constraints and capacity constraints. Production demand is considered a highly important constraint Aguirre et al.,(2008). Lee and Stecke (1996) and Li et al., (2011), apply constraints to ensure production demand for different product types is met. Lee and Stecke (1996) also introduce a flexibility capacity constraint to ensure that tasks assigned to a machine do not exceed the constraint, they apply a constraint to ensure each task is only assigned to one workstation. Hao and Shen (2008) introduce a different volume constraint; they constrain the system to have no extra WIP. This is considered to be a volume constraint as it reduces

the volume of parts in the system. Ghani et al.,(2012) apply fixed production volumes to their simulation model. Aguirre et al.,(2008) also place a constraint on maximum batch sizes to their model alongside a demand constraint. Iwata et al., (1984) apply a constraint to the overall warehouse capacity.

2.2.5.2 Routing Constraints

Routing constraints are key within modelling of production lines and are applied within a number of the papers. Lee and Stecke (1996) place a unidirectional flow constraint on their model formulation to ensure that the parts do not revisit any workstations. Cochran and Kaylani (2008) use production Kanbans to control the flow of parts, no part can be released until a Kanban has been received. Hao and Shen (2008) also apply Kanban routing constraints. Li et al (2011) apply defined production sequences and routing for parts. Dejong (2001) also applies routing logic to their test case. Hugan (2001) applies sequencing logic for some of the trucks within his model; however, some trucks are not modelled with set sequences for parts.

2.2.5.3 Configuration Constraints

Configuration constraints are highly popular amongst papers within this literature review. The most common configuration constraint considered is buffer size constraints considered by a number of authors. Aguirre et al., (2008) place a constraint on the number of stages which are allowed in work centres. Ghani et al., (2012) apply physical constraints to model the machining processes within the production line. They also apply limits to the processing speeds at workstations. Baines et al., (2003) apply minimum and maximum times for tasks to be completed. Similarly, Brennan and Rogers (1995) apply time constraints of the system for minimum and maximum repair times. Hugan (2001) applies a constant velocity to trucks in his simulation model. Iwata et al., (1984) apply a range of configuration constraints including limiting the number of workstations, tools, jigs, fixtures and modules. Han et al., (2003) apply

constraints to the total number of conveyor points and offline buffers, this limits the number of paint changes within the production line

2.2.5.4 Resource Constraints

Resource constraints are considered by a number of authors within this literature review. Ghani et al.,(2012) apply fixed shift patterns to their simulation model. Hao and Shen (2008) also apply fixed work schedules for their vehicle drivers. Aguirre et al., (2008) apply manpower constraints to their model. Hugan (2001) creates a shipping schedule which limits the availability of a rail crossing at certain times of the day. He also applies shifts for workers and accounts for employee traffic in his model.

2.2.5.5 Other Constraints

Various other constraints have been considered amongst papers in this literature review including cost, quality and safety constraints. Kouikoglou (1999) implements a cost constraint to limit the total expenditure of the system. He also considers fixed failure probabilities. Bulgak (1992) applies a quality based constraint: Average Outgoing Quality Limit. He requires 100% quality inspection until the AOQL rises to 100%. Chandra et al., (2003) list a number of different safety based constraints for their production line.

2.2.6 Model Formulation

Within literature, authors have formulated models of their manufacturing systems and assembly lines in a number of different ways. All of the studies analysed have modelled their systems using DES, although it has been noted that simulation is labour intensive and time consuming (Kibira & McLean, 2002).

In some applications alternative models are formulated and the results are compared to the simulation model; this is often carried out as a validation

exercise, as the models are created separately and validated against one another.

In other application papers the use of DES models has been coupled with other modelling methods to enable detailed system analysis and optimisation. Among the 52 papers considered within this review, 21 of the papers were found to apply only DES, 20 papers coupled DES with other methods such as heuristic rules and analytical approaches and 11 papers were found to compare DES and other modelling methods. This shows significant progress in the field of application of DES modelling through growing integration with other modelling methods. The following subsections outline the papers within each category and the modelling methods used.

2.2.6.1 Discrete Event Simulation

DES models have typically been used to answer specific operations related questions (Yang et al., 2013). The majority of authors have applied commercial DES software to model these systems such as QUEST (Lee et al., 2009; Lu & Sundaram, 2002), WITNESS (Ghani et al., 2012), ProModel (Marsh et al., 2010), Arena (Ferreira et al., 2012a) etc.

Arena commercial software suite has been particularly popular amongst studies. It has been used in various assembly applications including: to model a four closed loop network configuration (Ferreira et al., 2012; Ferreira et al., 2012), to evaluate different control policies (Koulouriotis et al., 2010) to validate an aggregated event scheduling method (Zhao et al., 2010), to validate an analytic model (Pradhan & Damodaran, 2009), to create a mixed model assembly line (Mendes et al., 2005) and to model a keyboard assembly cell (Spedding et al., 1998).

ProModel software has been used in a number of studies for various modelling tasks including: for modelling of varying assembly sequences (Marsh et al.,

2010), for the evaluation of the effect of worker skill dynamics on the cost performance (Stratman et al., 2004) and for productivity evaluation of manual and automated fitting tasks (Yu et al., 2006).

Another simulation suite QUEST has also been popular among papers studied. Lee et al., (2009) use QUEST to assist shop floor decision making. Lu et al., (2007) use QUEST to model potential moving line concepts for aircraft assembly and Kibira & McLean, (2002) use QUEST to build a DES model, but they also model the manual assembly operations using IGRIP.

Other commercial DES packages include Automod, WITNESS and Siemens PLM. Hugan, (2001) uses Automod to model an automobile truck plant in 3D, Steinemann et al., (2013) use Siemens PLM with a support system called SAMSON to support operational decisions downstream in the production cycle. Ismail et al., (2011) use WITNESS to create a DES model based on a UML representation. Hao & Shen, (2007) use AnyLogic with a hybrid approach combining Discrete Event and agent based methods for modelling the material handling process.

Various other modelling approaches have been used by authors including: a DES model in C++ to solve an operator allocation problem for hybrid assembly lines (Zeng et al., 2012), development of a simulation software program for flexible manufacturing systems (Iwata & Oba, 1984) and a DES program written in Pascal (Asil Bulgak, 1992). A number of authors use undisclosed simulation software for instance for the modelling of high mix low volume systems (Khalil & Stockton, 2010 and Li et al., 2011).

2.2.6.2 Hybrid approaches

Over a third of the papers selected have used simulation modelling in combination with some other modelling technique. These techniques include heuristic methods, mathematical programming formulations, and resource

planning methods amongst others.

Heuristic rules have been applied within the model formulation by a number of authors. Korytkowski et al., (2013) use heuristic dispatching rules within an Arena simulation model. Similarly Lee et al., (2009) use heuristic dispatch rules to make initial guesses and they integrate this with a simulation model in QUEST. Mendes et al., (2005) also use heuristics to build initial solutions and the heuristics are then fed into an Arena simulation model. DeJong, (2001) applies heuristic methods to assign test problems and then uses Microsoft Visual Basic 6.0 to simulate the different scenarios. Liu & Sanders, (1988) apply heuristic modifications to the stochastic quasigradient method to find optimal buffer sizes.

Analytical approaches have been applied in combination with simulation models; these include decomposition approaches, mathematical programming and artificial neural network models amongst others. Bulgak & Sanders, (1991) use an analytical method to determine the number of pallets for their system and they feed this into their simulation model. Mehrsai et al., (2013) use Artificial Neural Networks, in particular a radial basis function network, to model autonomous objects. Mehrsai et al., (2013) use a linear programing approach with crisp and fuzzy parameters for assisting with operational decisions for autonomous objects. Proth et al., (1997) use a non-linear programming formulation to model inventory control for a cyclic manufacturing system. Martinelli & Valigi, (1999) use a decomposition approach with the simulation model to split the problem into sub allocation problems. De Kok & Visschers, (1999) also use decomposition; their system is decomposed into a number of divergent inventory systems.

Methods have been applied to combine resource planning and material handling with simulation modelling software to enable full system optimisation. Cochran & Kaylani, (2008) use manufacturing resource planning and simulation in unison, Neira Dueñas et al., (2007) use Computer Aided Production Planning

and simulation simultaneously. Taghaddos et al., (2010) use simulation with Multi Agent Resource Allocation for a modular construction example. Hao & Shen, (2007) use AnyLogic with a hybrid approach combining Discrete Event and agent based methods for modelling the material handling process.

Visualisation methods have also been applied with simulation modelling. Wischnewski & Freund, (2004) use visualisation in 3D with simulation modelling to assist decision planners. Kibira & McLean, (2002) use QUEST to build a DES model but in addition they model the manual assembly operations using IGRIP.

2.2.6.3 Comparison Approaches

Pradhan & Damodaran, (2009) note that developing, validating and verifying simulation models is time consuming and computationally costly; analytical models can be a good alternative and easier to solve, particularly if the error between the models is insignificant. It is for these reasons that a significant proportion of papers have created alternative models and compared them to simulation models.

Analytic models have mainly been used for comparison to DES models. Manns & Elmaraghy, (2009) present a deterministic analytical model for inter-arrival times at a bottleneck station. Han et al., (2003) note that their analytical model is unable to handle time related constraints, which can be achieved by a DES model. Yang et al., (2013) formulate the multistage assembly line problem using both System Dynamics and a DES model. Pradhan & Damodaran, (2009) derive equations based on queuing networks with non-Poisson arrival process and non–exponential service time distribution for an analytical model of an assembly line. They use the DES model to validate the analytical derived model. Zhao et al., (2010) use an Arena simulation model to validate an aggregated event scheduling method, similarly Pradhan & Damodaran, (2009) use a model in Arena to validate an analytic model.

Other modelling methods have been used for comparison with DES models; these include normal approximation methods and regression models. Li et al., (2011) propose a new method based on Probabilistic Reanalysis to replace time consuming Monte Carlo Simulation runs. Similarly Liu & Sanders, (1989) create a system of equations as an approximation of the Monte Carlo Simulation method; they show that the method compares well to Monte Carlo Simulation. Desmet et al., (2010) compare simulation models with a new proposed normal approximation model method. McHaney & Douglas, (1997) create a meta-model of a simulation model using a full factorial Design of Experiments and they use the results to create a linear regression model. This regression model is shown to accurately reflect the simulation model with less than 5% error.

2.2.7 Optimisation Methodology

This review focusses in particular on the use of optimisation methods applied to the simulation of assembly systems. Studies analysed in this review were split into five different categories based on the optimisation methods applied by authors. The number of papers in each category is shown in Figure 2-4. The most common optimisation strategy is the use of what-if scenario analysis to compare different combinations of input variables and to assess their influence on the chosen output parameters. Other optimisation strategies include manual optimisation, artificial intelligence algorithms and other papers which do not perform optimisation and focus only on modelling. Details of each of these optimisation strategies are given in the following sub-sections.

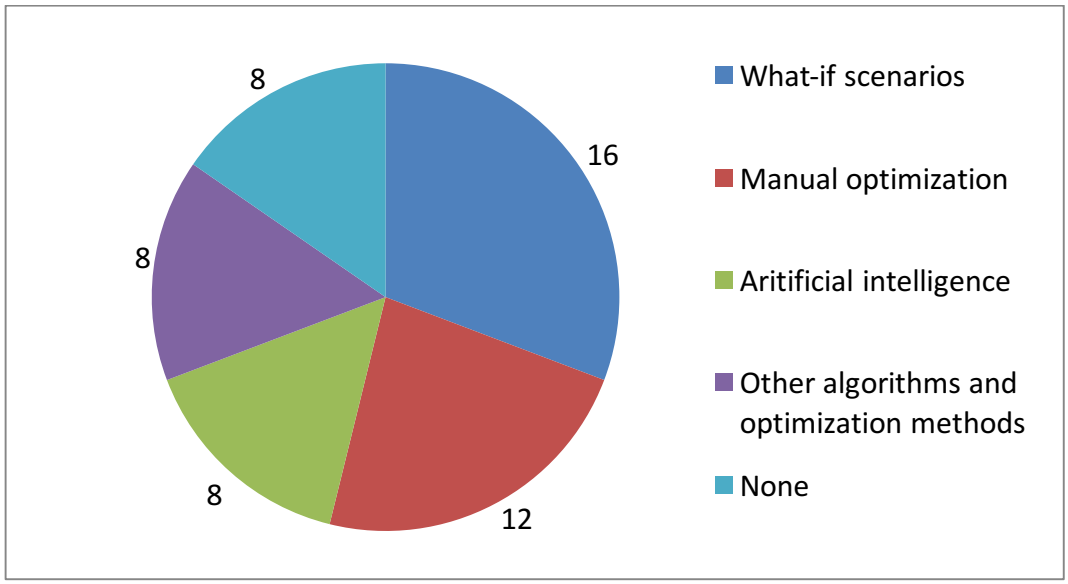


Figure 2-4 - A pie chart to show the number of papers appyling each optimisation method

2.2.7.1 What-if Scenario Analysis

The most common method of optimisation among papers studied was using what-if scenario analysis. Figure 2-4 indicates that over a quarter of the papers studied applied what-if scenario analysis as the method of optimisation. A number of authors propose analytic and simulation models and they run these models with different parameter scenarios or introduce changes to the models in order to optimise the system.

Studies have largely used scenario analysis to analyse and determine the effect of different parameter configurations on production and demand scenarios. Ghani et al., (2012) run their model with two different production scenarios, the second production scenario being a what-if scenario for demand increase. Koulouriotis et al., (2010) examine two different demand scenarios and six different control policies. McDonald et al., (2012) also use their DES model with various what-if demand scenarios, in order to identify bottlenecks and to ascertain if the system is ready to deal with the expected increase in demand.

Mendes et al., (2005) test five different demand scenarios; the results indicate that, for two demand scenarios, the required production levels could be met. Lu et al., (2007) simulate three different scenarios with different selection percentages at each decision point representing major design changes. Results are compared for different demand patterns. Ferreira et al., (2012) compare the results of the simulation model output with different values of an external input parameter. The impact of the number of pallets in the first closed-loop on the production rate is also considered. Aguirre et al., (2008) first analyse the possible bottlenecks of the system and then identify the critical processes. Alternative scenarios are then created based on increased capacity, resources and machines; the modification of operational rules is used to identify significant factors. De Kok & Visschers, (1999) analyse different allocation policies and find that pre-allocation policies perform well with respect to cost and service target levels.

Design of Experiments has been used as a more systematic method for performing parameter studies by a number of authors. Authors can use different experimental designs to perform analysis on a variety of different input parameter combinations to assess the effect on output parameters. Khalil & Stockton, (2010) design a series of experiments with varying cycle times between machines; they formulate the relationship between these times and blocking and waiting. They test these relationships with estimations and show less than 5% error in estimations. McHaney & Douglas, (1997) also use Design of Experiments to create a linear regression model of a simulation model to speed up the evaluation process. Stratman et al., (2004) create a Full Factorial experimental design with sixteen different simulation experiments. It is found that 60% of variable manufacturing costs for the temporary worker pool are due to learning and forgetting. Spedding et al., (1998) use a Full Factorial Design of Experiments to identify the best buffer and pallet sizes for maximisation of the throughput.

Two authors have performed comparative studies to understand the effect of the assembly sequence on the performance indicators. Li et al., (2011) make a comparative study of two different assembly scenarios concluding that the product design has a dramatic effect on the cycle time. Marsh et al., (2010) use their DES model and linked cost model to analyse the effects of different assembly sequences on the cost. Simulation results indicate the different assembly sequences have little effect on the average cost per product and cost per operation, however they have a large effect on the cost of inventory.

Labour scenarios have also been analysed using what-if analysis. Lu et al., (2007) produced various labour scenarios. Authors analyse the logistics of the stand movement and combine jobs which would require similar stands; they were able to identify an optimum solution from these scenarios. Yu et al., (2006) also compare manual against assisted operations, also finding that the assisted method yields are preferable as they lead to better productivity.

Authors have also used simulation to model and compare various other scenarios. Mehrsai et al., (2013) present two different scenarios for autonomous objects: a decentralised system and a system with a universal network for all stations; the latter is shown to perform better under test runs. Pradhan & Damodaran, (2009) use a simulation model in Arena and 25 different instances to measure the accuracy of the analytical estimates of the performance indicators. Results show that error depends on a number of factors including traffic intensity, product yields, number of nodes and number of shared nodes. Asil Bulgak, (1992) analyses the impact of including product quality considerations in the analysis and design of Automated Assembly Systems; this author concludes that the quality and inspection procedures should be specified as design parameters to enable full optimisation. Otamendi, (2011) screens three different cell layout designs for an assembly cell; these designs are compared with respect to the Work In Progress objective function, synchronisation of cycle times and minimisation of investment costs.

2.2.7.2 Manual Optimisation

Many authors use simulation and analytical models simultaneously to perform manual optimisation. These methods often include the analysis of simulation results and quantification of relationships between variables to allow the manual identification of optimal solutions.

A number of authors have used their simulation results to optimise a number of their simulation outputs using manual methods. Ghani et al., (2012) analyse their simulation results and show that there is an exponential relationship between cycle time and energy consumption and they suggest an optimal result from amongst the solutions. Marsh et al., (2010) run their simulation model a number of times to generate output data and these are used to feed into the cost model. An optimum solution can be seen from the results of the different simulation runs and through manual optimisation. Stratman et al., (2004) use results from the simulation to deduce that the largest reductions in manufacturing costs are obtained when workers with the highest levels of competence are put upstream in the production process. DeJong, (2001) uses a Design of Experiments to compare weekly and daily customer orders. Results indicate that using the daily customer orders results in better On Time Delivery (OTD) performance and that assignment of the test program after the assembly increases the OTD performance indicator. In a different application Wischnewski & Freund, (2004) use their developed tool for the simulation and 3D visualisation of the assembly system; this enables collision detection in the virtual commissioning phase to ensure smooth setup in real life.

Analytic methods such as equations have been used to assist in manual optimisation. Han et al., (2003) use an analytic method to assign workstations for painting to improve system throughput and reduce setup times. The system throughput is maximised by a reduction in the number of colour changes. Desmet et al., (2010) use their normal approximation models to optimise the required safety stock level for maintaining target service levels. Liu & Sanders,

(1989) develop a system of equations as an approximation method for allocating buffer sizes in order to maximise the expected production rate.

Authors have used simulation results to manually optimise specific parameters. Ferreira et al., (2012) implement a Decision Support System (DSS) with the view to optimise the production process. They vary the lengths of the buffers and choose values which maximise the production of the line and reduce blockage and starvation. Yu et al., (2006) suggest that a good workstation designer should know the trade-off between effect of lifting devices on productivity and lower back problems. They use their simulation model to compare manual sunroof installation with use of a lifting assist device, showing that use of the assisted device can increase productivity. Manns & Elmaraghy, (2009) synchronise design parameters to reduce the number of different inter-arrival times and increase the output of their simulation model.

2.2.7.3 Artificial Intelligence Methods

In recent years a growing interest has been seen in the use of Artificial Intelligence (AI) methods for solving optimisation problems. This trend can also be seen in the use of AI methods to optimise manufacturing and assembly systems. AI methods use concepts from nature to guide the optimisation algorithm in the search for optimal solutions. In particular, Genetic Algorithms (GAs) are a popular algorithm amongst studies. Other methods applied include Simulated Annealing (SA), Differential Evolution (DE) and Artificial Neural Networks.

GAs have been used to optimise system parameters for assembly systems. Korytkowski et al., (2013) use a GA coded in VBA with a simulation model in Arena to optimise dispatch rules. The conclusion is that there is no best dispatching rule, but some rules perform better than others for specific requirements. Koulouriotis et al., (2010) utilise simulation optimisation to choose optimum design parameters using a GA interface with the serial lines and

assembly system. They use resampling as a method to deal with the stochastic objective functions. Neira Dueñas et al., (2007) compare a Macroevolutionary method and a Genetic Algorithm. The authors demonstrate the benefits of the Macroevolutionary algorithm including visibility of convergence and a reduction in the number of parameters to adjust. Cochran & Kaylani, (2008) also use a GA to optimise their model, which includes stochastic DES parameters.

Simulated Annealing (SA) has also been applied to optimise output parameters. SA has been developed for application to combinatorial optimisation problems; hence it is an appropriate optimisation algorithm for the problem. Lee et al., (2009) use dispatching heuristics as a starting point to create solution schedules which are then optimised using the SA algorithm. Similarly Mendes et al., (2005) use a heuristic initially to create a solution; then SA is used to attempt to improve the number of workstations. Bulgak & Sanders, (1991) use a modified SA method and compare the results to the Stochastic Quasigradient method. Statistical test results suggest no difference in the results using both methods, however it is noted that the SA algorithm cannot be used for the optimisation of large systems because of the slow convergence to the optimal solution.

Other methods such as Artificial Neural Networks and Differential Evolution have also been applied. Zeng et al., (2012) propose a new PUDDE differential evolution algorithm. They apply this developed methodology to three test cases; the PUDDE algorithm performs better than GDE in all tests and cases. Mehrsai et al., (2013) use Artificial Neural Networks (Radial Basis Function networks) with a simulation model. They only train weights and suggest that by intelligently training all parameters of ANN, better results can be obtained.

2.2.7.4 Other Algorithms and Optimisation Methods

A variety of other optimisation algorithms and methods have been proposed within papers. These include exhaustive search algorithms, gradient descent

algorithms and response surface methodology.

Gradient descent methods have been particularly popular among studies. Proth et al., (1997) use a powerful stochastic gradient descent algorithm, which minimises a deterministic function between successive updates of a precision index. Results show rapid progression to the optimal solution in early stages of the algorithm. Infinitestimal Perturbation Analysis (IPA) has been used within studies to enable gradient estimation. Brennan & Rogers, (1995) use IPA to calculate gradient estimates for optimisation of an electronics manufacturing line. Kouikoglou, (2000) uses simulation with both IPA and the Standard Clock Method (SCM) to enable gradient estimation for the throughput. SCM is used as an alternative for when IPA fails to produce consistent gradient estimates. The steepest descent algorithm is applied with these methods.

Exhaustive search algorithms have also been applied to optimise assembly systems. Li et al., (2011) apply Solver within Microsoft Excel to solve their electronics assembly optimisation problem. The algorithm applied is an exhaustive search algorithm. Their methodology is used to optimise the system performance with budget constraints.

Other algorithms and methods have also been used to perform optimisation. Mehrsai et al., (2013) use a fuzzy optimisation model to solve their non-linear mathematical modelling formulation. Results show that autonomous objects can be incorporated to perform global optimisation linking decisions at the operational and strategic level. Taghaddos et al., (2010) use the greedy and ascending auction algorithms to optimise yard resource utilisation. The greedy algorithm is shown to produce sub-optimal solutions; this suggests that the ascending auction algorithm is the preferred option. Martinelli & Valigi, (1999) use a local search algorithm to optimise decomposed sub problems.

2.2.7.5 No optimisation

A number of authors only model their chosen manufacturing and assembly systems and validate these analytical and simulation models and do not perform any optimisation on their models. Often analytic models have been proposed as an alternative to simulation models as they require less computation time. Authors who have not explicitly considered optimisation have suggested their results can lead to optimisation in future research.

Language and control based modelling methods have been used to model systems, but these methods often do not consider optimisation. Ismail et al., (2011) create an explicit control structure using UML to automate a high volume assembly line. The UML structure is used to build a simulation model of the new automated system. Although all efforts are focussed on modelling, the number of operators is reduced by two thirds for the implemented automated system. Iwata & Oba, (1984) develop a simulation engine for a flexible manufacturing system. They also do not consider optimisation.

Many of the authors focus on modelling; they suggest how their modelling work can be further utilised to perform optimisation in future research. Li et al., (2011) integrate design information into their simulation model and they suggest that this will enable Electronics Manufacturing Services to optimise resource deployment for specific products. Steinemann et al., (2013) develop a Decision Support System to help downstream operational decisions; initial trials for the operational focus showed promising results where planning engineers and line managers could see the benefit of the proposed tool. Hao & Shen, (2007) concentrate on combining agent based and DES methods to model a flexible pull production line; they suggest that the material handling and arrangement of buffers can be optimised through comparative studies. Baines et al., (2003) perform a data collection exercise to analyse and quantify the variation in human workers skills in an automotive assembly line. This is used to produce probability distributions for the task time of different workers and tasks. It is

suggested that the resulting probability distributions can be used as an input into a DES model in further work.

2.2.8 Discussion on Optimisation of Assembly Systems Using DES

The detailed analysis of the papers considered in this study has led to the identification of a number of key trends and links within the field of assembly system simulation and optimisation. The key trends and potential areas for exploitation are outlined in this section.

Analysis of the publication year of papers within this study shows that the number of papers in manufacturing and assembly simulation area of research has grown considerably in recent years; this is shown visually in Figure 2-1. The number of papers has increased since 2001 showing growing interest in this subject area.

The various application domains of papers within this review were examined to assess trends. The most common application domain was for general production systems. The production line is a highly complex system often with stochastic variables, they are required to be flexible and often incorporate a product mix. This makes the job of production planners particularly complicated; hence production systems are well suited to modelling using DES. The automotive industry was the second most popular application domain amongst papers considered. This is a particularly common application area as the car assembly line often has many options (Steinemann et al., 2013), which leads to many different potential scenarios. The electronics manufacturing industry was the third most popular domain; test cases are often mixed product assembly lines which are required to be very flexible (DeJong, 2001; Mendes et al., 2005).

A number of common objective functions of simulation modelling and optimisation of assembly systems have been noted. The objective functions have been categorised as cost based, time based, utilisation, resource

management, bottlenecks, throughput and other objectives. The most popular objective functions among papers studied were the time based functions and throughput. This suggests manufacturers are keen to streamline their processes, reduce lead time and increase output of their assembly facilities.

Across the studies in this literature review various constraints have been applied, these constraints have been categorised as: volume constraints, routing constraints, configuration constraints on the system, resource constraints or others. Popular constraints include: buffer size, production demand and schedule constraints.

The majority of papers within this review were found to use commercial software packages for DES modelling such as WITNESS (Ghani et al., 2012), QUEST (Lee et al., 2009) and ProModel (Marsh et al., 2010) amongst others. With the wide range of software available, it is unlikely that there is a requirement for further DES software. Other approaches such as mathematical programming approaches and heuristic methods for representation have been applied in unison with simulation modelling. It has been noted that simulation models require expert knowledge to build and are computationally expensive to run. It is for this reason that a number of authors attempt to build analytical models to replace these simulation models. Simulation models were compared to analytical models by 11 of the papers studied. An increasing use of DES in combination with other methods has also been observed. These methods include decomposition approaches (de Kok & Visschers, 1999), and analytic methods (Bulgak & Sanders, 1991).

Optimisation approaches of the papers within this review have been categorised and analysed. The most widespread approach to optimising simulation models is the use of what-if scenario analysis. Authors have widely used scenario analysis to analyse the effect of different production and demand scenarios, labour scenarios and parameter choices. This scenario analysis enables production planners to reconfigure their facilities efficiently; however, this

method doesn't have the benefits of application of optimisation algorithms. Manual optimisation has been applied in a number of papers. This involves the analysis of results to deduce relationships and trends between variables and to identify optimum values based on specified objective functions. Authors have also used optimisation algorithms to perform optimisation with respect to their KPIs and objectives. Artificial Intelligence methods such as Genetic Algorithms and Simulated Annealing in particular have been applied in various studies (Mehrsai et al., 2013; Mendes et al., 2005; Zeng et al., 2012). Many of these authors have optimised multiple objectives simultaneously using either aggregated approaches to perform single objective optimisation (Cochran & Kaylani, 2008) or using explicit multi objective optimisation (Neira Dueñas et al., 2007). Authors who have concentrated on modelling Discrete Event systems, and not performed optimisation, have noted that the next step is to optimise their developed models.

In summary, this section has analysed studies within the area of DES for assembly optimisation. DES in particular has been used due to the variability inherent in the production lines. A number of trends have been identified within the literature such as the growing number of research papers in this field. Time based and throughput objective functions were the most widely used among papers studied; these objectives enable reduction in lead time and thus an increase in production and sales. Over a third of the papers considered have used hybrid modelling methods combining simulation and other modelling strategies such as heuristic methods, resource planning and analytical models. The most predominant optimisation strategy was the use of what-if scenario analysis; this enables simulation modellers to perform parameter studies to gauge the effect of input parameter changes on the output variables. The increasing use of a variety of Artificial Intelligence methods and multi-objective optimisation of simulation models has also been shown.

2.3 Visualisation of Manufacturing Processes

This section outlines research on the use of immersive visualisation and Virtual Reality (VR) methods to visualise manufacturing processes. This is a growing area of interest as VR technology is becoming more affordable for widespread adoption. A variety of technologies have been released in the last few years including various Head Mounted Displays (HMDs) and software programs for visualisation of immersive environments. These technologies can be applied to manufacturing process simulation to create a variety of benefits. The following sub-sections detail a number of chosen use cases of virtual manufacturing and engineering to visualise these processes.

2.3.1 Introduction

Virtual commissioning is becoming a growing area of increasing interest in industry, as these methods can be used to validate designs for entire systems before the system is constructed (Ghani et al., 2012). In particular 3D models are produced and used for providing physical description of the facilities and processes; these models can be used for experimentation. The models can be used to save time at the real start-up phase and to help identify and correct sub optimal behaviour (Wischnewski & Freund, 2004). The 3D models can also be visualised using immersive visualisation to give users a sense of presence and to enable error detection in layouts. The following section covers assembly visualisation examples within literature. This field of research is still recent; hence the papers identified in this section have mostly been from the last decade.

2.3.2 Visualisation of Assembly Processes

Ghani et al., (2012) use Virtual Engineering methods to visualise a case study of an assembly production line for an automobile engine block. The detailed design of the station's components has been developed in a Virtual Engineering

environment where their OP900 virtual station can be visualised. Device functionality has been imported into the WITNESS based DES model and the model has been used to determine the optimised production parameters for a given production rate, with the aim to minimise energy consumption.

Kibira & McLean, (2002) incorporate visual representation of workstation operations and DES assembly process flow into their simulation methodology. Authors aim to identify the data requirements and issues associated with simulation study of a small assembly line. They use QUEST for the simulation modelling and IGRIP to perform the graphical ergonomic modelling of the workstation operations. An assembly tree is constructed to show assembly stages; the precedence relations are then used to aggregate the tasks which can be performed at single stations. QUEST was populated with pre-built sub models and DELMIA language was used to program the detailed manual operations in the display of the virtual reality environment. The DES modelling was found to be a straightforward process, however the animation construction was found to be very time consuming and tedious, particularly for worker kinematics. The authors note that the model would not be able to deal with the realistic scenario where operators adapt to different situations.

Wischnewski & Freund, (2004) develop a software program to model and emulate transport systems. They simulate a mechanical assembly process using time discrete simulation and 3D visualisation to enable collision detection for construction. The simulation results can be visualised by 3D views; in addition, interaction is possible to allow experimentation whilst the simulation is running. The software is also able to detect unrealistic states.

In a more recent study, Behdad et al., (2014) use Immersive Computing Technology (ICT) with an analytical mixed integer nonlinear program model to determine optimum disassembly sequences for the Burr puzzle problem. This problem includes some uncertainty which is also modelled. The results from the use of ICT have indicated that there are potential disassembly sequences which were not clear from the mathematical model.

Zankoul and Khoury (2014) develop a simulation model with 3D animation capabilities for an on-shore wind farm construction process. The simulation model is generated in Anylogic. The geographical location of the proposed wind farm is loaded into AutoCAD and used to generate the visualisation. The authors use the 3D animation to allow users to visualise the model in 3D and to assist with validation of the simulation model.

In summary there have been various studies which incorporate the use of visualisation methods for manufacturing and assembly system optimisation. These include the development of systems to perform simulation and visualisation (Wischnewski & Freund, 2004), the linking of DES with detailed visualisation methods (Kibira & McLean, 2002) and the use of ICT to assist in optimising disassembly sequences (Behdad et al., 2014). In particular, the area of visualisation for optimising system performance is highly interesting. Wischnewski & Freund, (2004) utilise visualisation with simulation, however they do not propose functionality for interacting with a model from within an immersive environment. This could be a highly interesting area of research.

Research in the area of visualisation of manufacturing simulation models is still growing with the increase in new technologies such as Virtual Reality. This is an area which will receive significant amount of research interest in the coming years as new technologies continue to be released enabling better process visualisation.

2.4 Discussion on Literature Review

The literature review which has been carried out has generated a range of interesting insights regarding DES, visualisation and optimisation. This section outlines the key trends observed.

A number of research trends relating to DES modelling and optimisation of assembly systems have been made. Trends are as follows:

- Application domains applied within studies are domains with high variability; these include: automotive, electronics assembly and other flexible manufacturing systems. This suggests that DES and optimisation is particularly applicable to assembly problems with high variability.
- Predominantly time-based objective functions have been used for optimisation within applications. However, it has been noted that many of the proposed objectives are interlinked, these include: bottleneck reduction, utilisation, cost, throughput and resource management. For instance, bottleneck reduction will result in higher utilisation and higher throughput.
- Predominantly commercial software packages have been used for DES. Arena was found to be the most popular package.
- Over a third of papers have used hybrid modelling approaches, combining heuristic rules, analytical approaches and visualisation with DES.
- Many papers have compared analytical models with DES; this is due to the high computational cost of DES.
- Most of the papers applied what-if scenario analysis to optimise their assembly systems.
- Artificial Intelligence has been identified as an area of growing interest for optimising DES models of assembly systems
- A number of authors have considered multiple objectives in their optimisation; however these papers have predominantly aggregated their objectives into a single function for optimisation.

Various trends have been observed relating to visualisation of manufacturing processes, these include:

- Visualisation and DES is a relatively new area in literature with few papers in the field.
- Creation of animations for worker kinematics is time consuming and tedious

- Visualisation and DES can be used to assist with collision detection for construction of assemblies
- Visualisation of models of manufacturing processes (virtual commissioning) can be used to help reduce time in the real life start-up phase

2.5 Research Gaps

The papers analysed within this review have led to a number of research gaps being identified.

The research gaps are summarised as follows:

- Although many authors have linked DES models with multi objective optimisation algorithms for a range of specific test cases, the literature does not contain discussion and explanation on the process of creating models of assembly systems which are ready-to-optimise. There are a few challenges associated with creating a generic process to create ready-to-optimise DES models which could be explored.

- The literature shows a growing interest in hybrid optimisation and AI methods. There is also growing interest in the use of multi-objective optimisation; many authors aggregate their multiple objectives into a single objective. A few authors explicitly apply multi objective optimisation. However, there is no literature to relieve the computational cost of running multi objective optimisation on DES models of assembly systems.

- Within the area of immersive visualisation, there is a growing interest in the use of DES with immersive visualisation. Although there is research in the area of using immersive visualisation and DES for layout planning; there is a lack of research in scenario analysis from within the 3D environment. Creating a closed feedback loop to visualise and modify

DES models within an immersive environment can assist decision makers.

- The literature search has revealed papers which perform optimisation on DES models; however, after the DES results are obtained there is a lack of research into use of data mining techniques to assist users with understanding insights from the results. In particular decision trees can be used to link input and output variables from DES models visually and assist decision making.

These are highly interesting areas and could help to engage manufacturing stakeholders and improve decision making with DES models.

2.6 Industry 4.0 and the Internet of Things

In recent years, technological advancements have enabled greater connectivity, increasing data volumes and a rise in computational power; these advancements have led to the creation of Industry 4.0 and the industrial Internet of Things (IoT). This section summarises Industry 4.0 and IoT.

Industry 4.0 is also known as the fourth industrial revolution, where an industrial revolution is a disruptive leap in industrial processes resulting in significantly higher productivity (Blanchet et al., 2014). Industry 4.0 collectively refers to a wide range of current concepts and technologies which can be used to enhance the productivity and efficiency of manufacturing systems and the competitiveness of manufacturing companies (Lasi and Kemper, 2014). Industry 4.0 concepts include: mass product customisation, short batch sizes, flexible automated production systems, part tracking in production lines, self-awareness of components, part machine communication, improved human machine interactions and the optimisation of production amongst others (Posada et al., 2015). Industry 4.0 will also enable closer cooperation between business partners and employees providing new opportunities for mutual benefit (N.N).

Internet of Things refers to a network of resources, information, objects and people which can incorporate the entire manufacturing process and create ‘smart factories’ or environments (N.N).

The following subsections summarise the key concepts from Industry 4.0: Cyber-Physical Systems, Big Data and Smart Factories.

2.6.1 Cyber Physical Systems

Cyber-Physical Systems (CPS) are systems of collaborating computational entities which are in intensive connection with the surrounding physical world and its on-going processes (Hellinger and Seeger, 2012). When CPS are applied to production systems they are known as Cyber Physical Production Systems (CPPS). CPPS comprise smart machines, storage systems and production facilities capable of autonomously exchanging information, triggering actions and controlling each other independently (N.N). Lee et al., (2014) note there is a challenge of rapid decision making to improve productivity. CPPS will be able to be controlled in real time (Blanchet et al., 2014) thereby increasing the agility of production environments and increasing productivity. CPPS will enable flexible changing of machines and production processes to effectively reduce downtime within production facilities and increase efficiency (Blanchet et al., 2014). CPPS are also able to interact with humans using interfaces (Monostori, 2014).

2.6.2 Big Data

Current manufacturing systems require deeper analysis of various data from machines and processes to improve productivity and efficiency. With the advent of smart sensors such as RFID technologies, collecting data has become a simple exercise (Lee et al., 2013). When this data is aggregated, this information is called ‘Big Data’. Big Data technologies include storage (pre-processing), data processing, data visualization (post-processing), Big Data

analytics, as well as models and algorithms for decision-making (Zhong et al., 2013). A sensor-packed manufacturing system in which each process or piece of equipment provides live event and status information, coupled with market research provide a wealth of information (Babinceau and Seker., 2016). Following the data collection, Big Data technologies such as algorithms can then be applied to the data to generate and action meaningful insights in real time; visualisations can be created to demonstrate such insights to production management.

2.6.3 Smart Factories

A Smart Factory is a manufacturing solution that provides such flexible and adaptive production processes that will solve problems arising in a production facility with dynamic and rapidly changing boundary conditions in a world of increasing complexity (Radizwon et al., 2014). A smart factory will combine concepts such as CPPS, Big Data, automation and other associated technologies to create efficient and configurable production facilities. Within a smart factory, different manufacturing resources like workers, machines, materials, tools, and logistics vehicles are turned into smart manufacturing objects (SMOs) that are able to interact and interoperate with each other to ultimately achieve smart operations and decision-making (Zhong et al., 2013). These SMOs will hold data relating to themselves and the means to communicate with one another in order to optimise factory operations. Manufacturing performance also heavily relies on the logistics efficiency and effectiveness (Lu et al., 2017), hence automated logistics management such as Automated Guided Vehicles (AGVs) will be a crucial component of smart factories. Radizwon et al., (2014) also note that a combination of software, hardware and/or mechanics within a smart factory should lead to optimization of manufacturing resulting in reduction of unnecessary labour and waste of resources.

2.7 Methods and Software

This section details the methods and software which have been applied within this thesis. The software and methods outlined in this section include: WITNESS DES software, APOW tool integration and optimisation platform, Visionary Render, test cases used and modelling methods.

2.7.1 Software used

WITNESS is commercial DES software which has been used by a number of authors in the literature review. WITNESS enables users to model their systems in a 2D environment and 3D environment to visualise the processes. Users can model entities, resources, parts, shift patterns, variables, probability distributions and graphs within the environment. Within WITNESS, users can input system logic such as push and pull rules, part routing and other complex dynamic logic to represent the system. Users can modify the entity representations to resemble the real life system as much as possible. Once the system logic is input into the WITNESS model, users can run the model and visualise the processes running to assist with model validation. Graphs and variables can be visualised whilst the model is running to monitor the simulation progress. WITNESS can also be linked to Microsoft Excel to import spreadsheets, variables and values into the model; this is particularly useful for systems where data is already stored in Excel spreadsheets. The model can also be built into a 3D environment; users can add 3D models for each of the entities and then create a 3D visualisation. The 3D model can also be run. WITNESS 3D can also be linked with Visionary Render to enable further functionality.

APOW is an in-house tool integration and optimisation platform at General Electric. APOW can be used to link together various tools, scripts and optimisation algorithms to optimise different models. APOW has also been integrated with a library of multi objective algorithms jMetal (Durillo & Nebro, 2011) which can be used to optimise different models. APOW can be linked

with WITNESS through using either WITNESS Command Language (WCL) functionality or through calling WITNESS as a batch file. Once a link has been established between APOW and WITNESS, APOW can run multi objective optimisation on WITNESS DES models.

Visionary Render is a type of Virtual Reality software, provided by Virtalis, which enables users to visualise their 3D models in an immersive environment. Users can connect Visionary Render with VR caves, VR power walls or Head Mounted Displays (HMDs) to visualise their models in VR. Visionary Render for WITNESS is a module which allows users to visualise their 3D WITNESS DES models within Visionary Render and ultimately visualise their DES models in various VR formats. Users can run their models within the VR environment and visualise the dynamic processes to assist with model validation and design reviews.

2.7.2 Test Cases

Various real life and example test cases have been proposed to test and validate the frameworks proposed within this thesis. An assembly process optimisation test case has been proposed which models a turbine assembly stand within a factory. This test case has two main objectives: minimising the lead time and minimising the cost of production. This test case is a resource allocation test case as the allocation of more resources leads to a reduction in lead time. However, the allocation of extra resources also leads to increased production costs; hence this problem has conflicting objective functions and is a multi-objective optimisation problem. A layout optimisation test case has also been proposed for a repair facility. The facility carries out periodic maintenance on components. Different maintenance intervals are specified for the components which lead to different maintenance requirements. In between maintenance periods the factory is required to be rearranged to align with the new maintenance requirements. The rearrangement of the factory layout should

ensure that the distance travelled by parts is minimised to reduce the overall repair time. This rearrangement should ensure that sufficient space is allowed for each workstation and no safety problems arise from the new layout. 3D visualisation for this test case can be used to validate the new layout. Two further example test cases have been proposed in Chapter 6 to represent small production lines. The first example test case is a small system which produces two different parts. The system includes a crane, labour, 3 workstations and a buffer. The main objective for optimisation is maximising the throughput. The second example test case is for a single piece flow medium volume assembly line. The assembly line has 8 workstations with buffers between each workstation and 1 input buffer. All parts much be processed by all workstations, there are no cranes or materials handling processes required between workstations. The main objective functions for this test case are throughput maximisation and reduction in the total 'man hours' for the assembly. These are considered to be conflicting objective functions.

2.7.3 Modelling Methods

A number of different modelling methods have been applied within this thesis; these include: DES models, linear regression models, decision trees and random forests. DES models have been generated for all of the test cases within this thesis. DES models, as introduced in Chapter 1, provide a method to generate dynamic representation of systems while accounting for uncertainty in variables. However, large DES models are computationally expensive to run particularly when linked with optimisation algorithms. Hence alternative modelling methods have been proposed to deal with the large computational cost of DES. Linear regression models can be used as meta-models of DES models, as applied by McHaney and Douglas (1997). Linear regression models can be used to fit a predictive model to a set of data to predict the output variable given the input variables. Linear regression has been chosen as the interactions being modelled within the test cases are expected to mostly follow linear relationships. A least squares method is used to create the linear

regression models, where the model is chosen to reduce the value of the sum of all squared residuals. In the test cases, linear regression models are created for multiple objective functions. Decision trees are another modelling method applied within this thesis. Decision trees are a modelling method used to assist users with decision making. A decision tree is a tree like graph structure which shows the impact of different decisions on the chosen output variable. At each node the expected value of the output variable is given and a decision is stated. The two leaves which come after the node show the two options for the input variables to take and the subsequent expected values of the output variable. This ability to show the impact of decisions is highly useful to decision makers. Random forests are also used within this thesis as an alternative method to linear regression for predicting new data points. Random forests are a method to improve the prediction accuracy of decision trees. Random forests use sampling with replacement and aggregation of decision trees to improve prediction accuracy.

2.8 Chapter Summary

This chapter is split into various sections. The first part introduces DES, the second part presents a literature review covering the use of DES and optimisation for assembly systems. The third part presents a literature review of applications on the visualisation of manufacturing processes.

Within the literature review, key information such as application domain, objective functions, constraints and optimisation methodology have been categorised for the analysed papers. The results have been used to identify research trends and research gaps. Trends such as the increasing use of Artificial Intelligence, hybrid optimisation methods and growing number of papers in these areas have been highlighted.

The main research gaps identified are:

- the literature does not contain discussion and explanation on the process of creating models of assembly systems which are ready-to-optimise.
- there is no literature to relieve the computational cost of running multi objective optimisation on DES models of assembly systems.
- there is a lack of research in scenario analysis from within the 3D immersive DES model.
- there is a lack of research into use of data mining techniques to assist users with understanding insights from DES model results

The last two sections summarise Industry 4.0 and the methods and software which will be used within this thesis.

CHAPTER 3 – DEVELOPMENT OF READY-TO-OPTIMISE DES MODELS

The literature review carried out in Chapter 2 demonstrates the increasing application of optimisation methods to manufacturing simulation models; however, the papers reviewed still prominently applied what-if scenario analysis to improve performance.

In order to create a DES model which is ready-to-optimise, often a number of changes need to be made to the original DES model. This chapter presents a framework to enable the development of ready-to-optimise DES models. The framework is subsequently applied to two different manufacturing process test cases.

This chapter aims to achieve the following:

- Present a framework to enable the development of ready-to-optimise DES models
- Apply the developed framework to two different test cases: turbine assembly test case and repair facility test case
- Detail any challenges faced whilst making each model ready-to-optimise

3.1 Background

The application of optimisation methods to DES models is becoming an area of growing interest, as has been identified through the literature review. The most common method of optimisation used is ‘what-if’ scenario analysis, however the use of optimisation algorithms is growing in this area.

The application of what-if scenario analysis or optimisation algorithms to DES models often leads to a new set of requirements to be considered during the simulation modelling. Within literature, authors have generally applied optimisation to DES models for specific test cases. Often bespoke modifications are made to enable optimisation of DES models. However, authors have not attempted to generalise the types of modifications required to enable optimisation for generic simulation models of manufacturing processes.

This chapter presents a framework to enable the creation of ready-to-optimise DES models. This framework is novel as literature in this area has focussed on generating DES models for specific applications. Many of the studies have used some of the methods within the proposed framework, however this work creates a structured formal processes required for creating optimisation ready DES models. The framework provides a process which users can follow to develop a generic DES model which is ready-to-optimise. The model can subsequently be optimised through what-if scenario analysis or the application of optimisation algorithms to the model. The framework has been applied to two test cases of different manufacturing processes to demonstrate the framework and identify any shortcomings.

The first test case is an assembly test case for turbines. In particular, the simulation model is aimed at optimal resource planning to minimise lead time and cost. This resource optimisation is similar to work by McHaney and Douglas (1997). For this test case, optimisation is carried out through application of optimisation algorithms. The second test case is a repair facility layout optimisation test case; this simulation model is aimed reconfiguring the plant to reduce the distance travelled by parts whilst ensuring high utilisation of workstations. For this test case, optimisation is carried out through what-if scenario analysis. This layout optimisation test case is similar to the test case in Otamendi (2011).

3.2 Framework to enable the development of ready-to-optimise DES models

The literature review identified the application of optimisation methods to DES models of assembly systems as a growing area of interest. However, there is no research to generate a framework to enable the generation of DES models which are ready-to-optimise. This section introduces a framework which can be used to build a simulation model which is ready-to-optimise. This framework can either be used to build a model or can be applied to an existing simulation model. The framework is applied to two case studies in the subsequent sections. These test cases have been chosen as they represent different types of problems which can be solved through DES.

The framework involves a set of steps which should be carried out to generate an interface for the input and output variables of a DES model. The framework is shown in Figure 3-1. Each stage of the framework is summarised as follows.

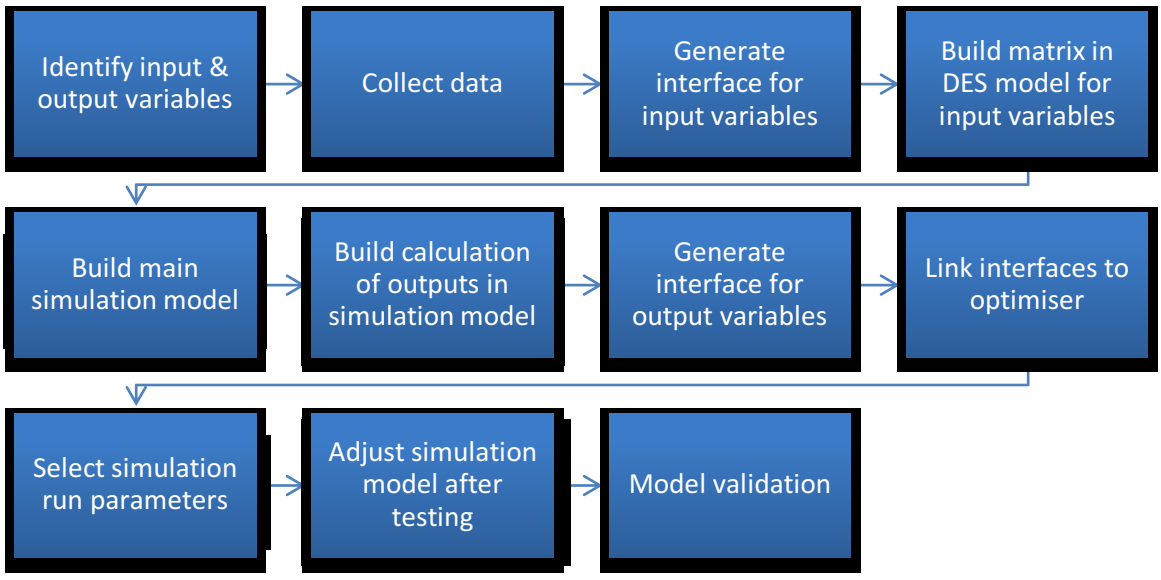


Figure 3-1 – Framework to enable development of ready-to-optimise DES models

Identify input and output variables

To optimise any DES model, the input and output variables must first be selected. The input variables constitute the variables which users will be able to change within the simulation model. The selection of input variables should include the definition of the range of values which the input variables can take and the type of values which they can take (continuous or discrete values). The range of values for each input variable should be selected to be as small as possible, as this will minimise the computational effort required to optimise the model. The total number of input variables should also be minimised to reduce the computational effort required.

The output variables constitute the optimisation objectives which are to be minimised or maximised. The output variables should be dependent on the input variables selected. Typical output variables include: cost, profit, throughput and utilisation.

Collect data

Data should be collected relating to the variables within the simulation model which are predefined. Collection of data for probabilistic variables is highly important as this can affect the model validity. Data for probabilistic variables should be collected for multiple instances and under varying conditions to enable a fair representation of the variation to be created. Baines et al., (2003) for instance collect data on variation in human workers' skills and the impact on their assembly line.

Generate interface for input variables

This process involves creating an interface which users can utilise to input their decision variable choices. Within this research, an Excel based interface is recommended as it is user friendly and Excel is widely used within manufacturing. This interface will be linked directly with the optimiser.

The interface should be set up to show the input variables and their respective values. The interface should be directly linked with the DES model.

Build matrix in DES model for input variables

At this stage, if the DES model does not already exist, the first part of the model is built. The first step is to generate a matrix within the model to contain the input variables. The matrix will enable users to visualise the input variable values for the simulation model.

Build main simulation model

At this stage the main simulation model is built. The entities and processes within the simulation model are built up and the interactions are generated. Any probabilistic parameters are also added to the simulation model. Any assumptions should be documented for future reference.

Build calculation of outputs in simulation model

The output variables are added to the simulation model. The output variables include optimisation objectives such as: cost, profit, throughput and lead time. The outputs are calculated within the model and stored within variables.

Generate interface for output variables

An interface should be created to store the output variables. An Excel based interface should be created for users to visualise simulation output parameters. The spreadsheet should contain columns for variables and rows for each simulation time period. Users are then able to see how the simulation output variables change during the simulation run. The interface should also contain summary statistics from the whole simulation run such as throughput, cost and lead time. These statistics provide a high level overview of the simulation run performance.

Link interfaces to optimiser

The optimisation algorithm or method should be linked to the interface for the input and output variables. The optimisation will change values in the input interface and the resulting values will be printed into the output interface. A number of authors have generated links between DES models and optimisers (Korytkowski et al., (2013), Zeng et al., (2012)), however in some instances the algorithm may be within the DES model tool used.

Select simulation run parameters

The simulation model should be tested to generate the appropriate simulation run parameters to be used for optimisation. The simulation run parameters to be selected are: simulation run length, simulation run window and number of replications (Law, 2014).

Adjust simulation model after testing

After the simulation model and optimisation link are set up, initial testing can be carried out to verify that the model performs as expected with different input variable combinations. Following the model testing, adjustments should be made to the simulation model as required.

Model validation

The model should be validated to ensure it represents the manufacturing system with an acceptable level of accuracy Law (2014). Historical data from the system can be used to perform model validation, expert based validation can also be applied.

3.3 Test Case 1 - Assembly

DES has traditionally been used for production and factory planning, however in recent years DES has been increasingly applied to manufacturing system operations applications (Negahban & Smith, 2014). The assembly process of high-value, low-volume components such as turbines is complex and multidimensional with stochastic parameters; this renders DES ideal as the modelling methodology for this problem. DES can be applied to model the assembly process to enable optimisation of operational aspects such as labour assignment to assembly tasks.

This subsection details the turbine assembly test case. The test case has been modelled in WITNESS using the framework introduced in section 3.2.

3.3.1 Problem description and process modelling

The assembly of turbines is a largely manual process which is carried out on an assembly stand. The main turbine components are brought to the assembly stand from the nearby storage areas using cranes. Due to the manual nature of turbine assembly, the labour assignment is highly important. The assignment of extra workers to the assembly stand generally results in reduced overall lead time. However, this reduction in lead time comes with an increased production cost. If too many workers are assigned to the stand, the idle time of workers increases. This conflicting nature of production objectives leads to the requirement for the process to be optimised with respect to multiple objectives.

A process model of the turbine assembly has been created to ensure accurate understanding of the overall process and alignment with stakeholders. IDEF3 was chosen as the process modelling method. The IDEF3 process model was created in MS Visio to analyse the component constraints and inter-task dependencies and to fully understand the logic behind the assembly process. The turbine assembly schedule documentation from the factory was analysed to create the process model. Initially the assembly process was split into 13 major assembly tasks; two of the tasks were identified as preassemblies. Figure 3-2 demonstrates this initial process model.

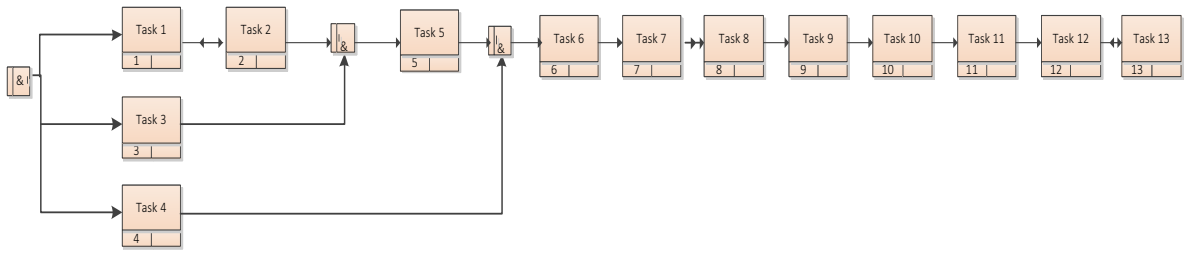


Figure 3-2 IDEF3 process model of the turbine assembly

The major tasks from Figure 3-2 have been subsequently decomposed to create a detailed process model of tasks. Figure 3-3 demonstrates the decomposition of Task 4 from Figure 3-2. From the decomposed IDEF3 process model it was observed that the turbine assembly process is largely linear with minimal opportunity for rescheduling or parallel assembly tasks. However, a

significant number of processes in the decomposition led to the possibility of rework or emergent work. For a number of emergent work tasks, the task could be escalated to create a Non Conformance Report (NCR). The NCR would then take a variable amount of time to be resolved based on the nature of the report. Figure 3-3 shows three tasks which could lead to requirement of emergent work or NCRs; subsequently the process will loop back to a previous step.

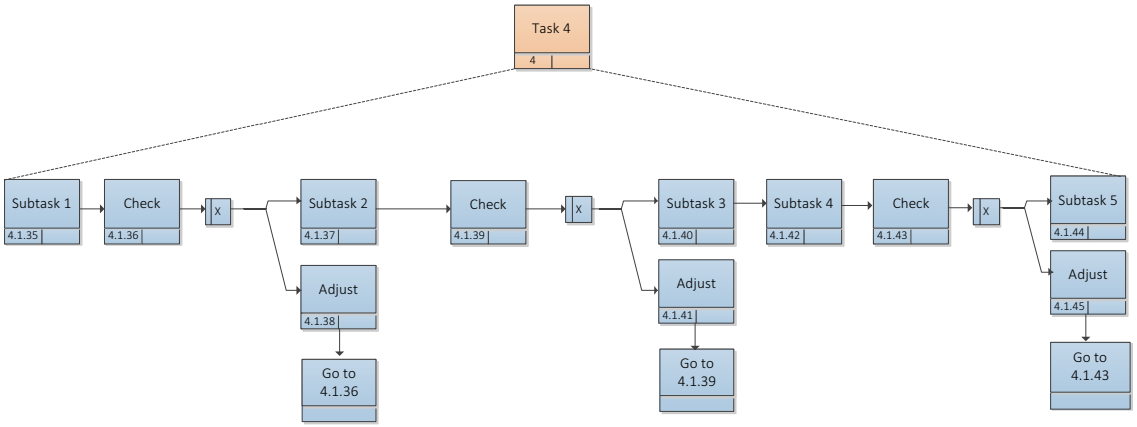


Figure 3-3 Decomposition of Task 4 from IDEF3 process model in Figure 3-2

Upon analysis of the system using the IDEF3 process model, the turbine assembly process was found to be stochastic, dynamic and discrete in nature; this leads to the confirmation and selection of DES as the most appropriate modelling method.

3.3.2 Application of framework to create ready-to-optimise DES model

The framework presented in section 3.2 has been applied to create the DES model. The following sub-sections detail the application of the framework to this test case.

Identify input and output variables

The main inputs for the model are the number of workers (fitters) per shift for each of the tasks, as the assembly is primarily driven by the work carried out by fitters. The fitters work on three shifts in the day each lasting eight hours. The

input variables are the number of fitters assigned per shift to each of the 24 assembly tasks. Upon assignment of extra fitters to each assembly task, the workload is split evenly between the workers and the task time is updated to reflect the new completion time. By definition, the assignment of more workers will reduce the overall time taken to complete each turbine assembly.

The main outputs for the simulation model are the running average lead time and running average number of fitters. The running average lead time is a clear objective for optimisation. Intuitively any optimisation algorithm applied to the model to minimise lead time would lead to solutions which schedule as many workers as possible to minimise the lead time. Hence another objective function is required to reduce the number of workers on the assembly stand. A cost function has also been introduced into the simulation model as the second objective function also to be minimised. The cost function is calculated as the weighted arithmetic mean of the number of workers for each assembly task; the weights are the length of time the workers spend on the stand for each task. It is assumed that these objective functions are conflicting based on the model formulation. No other constraints or objective functions have been applied within the simulation model.

Data collection

A factory visit has been carried out to enable accurate data collection for the assembly process. Accurate data collection was a significant challenge as the assembly process is a manual process. Data was collected from eight different previous turbine assemblies, this data was used to model the main assembly process and to analyse variation between different assemblies. The data collection was initially carried out through analysis of time stamps for task completion; however, it was found that time stamps could correspond to either completion of the task or completion of the quality control checks. Further data was collected based on worker timesheets.

Generate interface for input variables

The collected data has been used to generate a spreadsheet for the assembly tasks for each turbine assembly. An example spreadsheet is shown in Table 3-1. In the spreadsheet, each task is assigned a row. Two columns are created for each turbine assembled: the total number of planned man-hours and the total recorded man-hours. Another column has been used to indicate the number of workers which were assigned to each task. The final column of the sheet has been used to create cycle times for each task by dividing the actual man hours by the number of workers. Once this information was collated for the 8 turbine assemblies it was clear that 2 of the assemblies were outliers and had significantly longer cycle times compared to the other assemblies. These 2 outliers were excluded from the model building process.

Emergent work tasks have also been added to the spreadsheet as demonstrated in Table 3-1. These activity times were not included in the cycle times for each of the assembly tasks as the NCRs occur stochastically and have stochastic durations. Factory visits and historical data have been used to create an estimate of the average number of NCRs per turbine assembly and the minimum and maximum time required to fix an NCR. Details of modelling of NCRs are provided in the next subsection.

Table 3-1 – An example table to show the process of creating cycle times for each turbine assembly task

	Turbine assembly 1			
	Man hours planned	**Actual man-hours**	**Workers assigned**	**Cycle times in hrs**
Task 1	5	5	2	2.5
Task 2	15	14	2	7
Task 3	16	13	1	13
Task 3- Emergent work*	-	2	2	1
Task 4	7	9	3	3

Build matrix in DES model for input variables

The main aim of the optimisation of this simulation model is to optimise the lead time with respect to the resource allocation. The addition of resources to the assembly stand results in an additional cost to the assembly, however it contributes significantly to the reduction in lead time. An excessive number of workers on the stand will not result in the efficient completion of the assembly tasks as there will be too many workers; this will result in idle behaviour. In order to prepare the model for optimisation, it was chosen to limit the upper boundaries to reasonable values. The upper boundaries for the number of workers on the assembly stand have been set to low values to ensure that the model realistically represents labourers carrying out work. For the assembly tasks the minimum number of workers is always 1 and the maximum could be 2, 3 or 4 (depending on the nature of the task).

In order to allow the optimisation algorithm to vary the resource allocation, a matrix has been created to represent the Labour Available Per Shift (LAPS) for each assembly task. The matrix rows represent the 24 main assembly tasks; the 3 columns represent the 3 shifts which the labourers work. The matrix contains 72 values which are the optimisation inputs. The 72 variables are given names *LSx_y,* where *x* is the shift number and *y* is the task number.

This matrix enables easy visibility and modification of the input variables. A variable has also been created called ‘CurrentLabourAvailable’; this is used as the labour requirement for each assembly task. For each assembly task, all of the current labour available would be automatically assigned to that task. This functionality enables the adjustment of the resource for each assembly task and each shift according to the input variables within the ‘LAPS’ matrix.

Build simulation model

The assembly stand has been modelled in WITNESS using a multi cycle machine construct; the assembly tasks were assigned as different cycles on the machine. A total of 24 main assembly tasks were modelled on the assembly stand; three preassembly tasks were carried out elsewhere. The cycle times for the 24 tasks were created by averaging the cycle times from the six previous turbine assemblies.

A number of key assumptions have been made to model the assembly process:

- The time taken for movement of parts in the simulation model is accounted for in cycle times
- The crane is available 100% of the time for the assembly stand
- Parts and preassemblies are always ready when required
- As more workers are assigned to the assembly stand, the cycle time will reduce proportionally
- All available labour will be assigned to the assembly stand

A crane has been modelled for moving components to and from the assembly stand; it is assumed that the crane is assigned only to the assembly process. A path has been built into the simulation model for the crane to use to transport parts to and from the assembly stand. Travel times are assumed to be included within cycle times.

The turbine assembly process has been modelled with each turbine being assembled distinctly from all other turbine assemblies; parts are not interchangeable between turbines so each turbine assembly is disjoint from all other assemblies. All parts and pre-assemblies are assumed to be available immediately when required.

It is assumed that if more workers are assigned to an assembly task, the cycle time will reduce proportionally. This assumption will not prove to be valid for large numbers of workers. Hence the upper bound of workers assigned to assembly tasks will be set to a realistic value for the optimisation.

A major challenge in the modelling of the turbine assembly process is the modelling of the variability within the assembly process. In particular, during the assembly process a NCR could arise at any stage within the assembly. It has also been noted that there were various stages within the assembly where a NCR could occur which were not represented within the decomposition of the assembly process. This creates difficulty for the modelling of the assembly

process; in particular, it does not enable the model to be built with explicit decision points for NCRs. In order to overcome this difficulty, NCRS have been modelled as breakdowns within the DES model. When an NCR occurs, the entire assembly stand will break down and no progress is made until the NCR is resolved. A uniform distribution has been used to model the frequency of breakdowns; another uniform distribution has been used to model the time taken to repair the breakdown. This method of modelling the NCRS resolves a significant amount of difficulty in the modelling process as it removes the requirement for collection of detailed data on exact NCR timing. Such detailed information would be difficult to obtain and would ultimately lead to difficulty validating the model.

The chosen method for modelling the NCRs assumes that the whole of the assembly process is put on hold whilst the NCR is resolved. This is a realistic assumption in most cases. However, the production manager will always restructure work accordingly, in order to perform as many tasks as possible whilst the NCR is being resolved.

Build calculation of outputs in simulation model

The outputs of interest were built into the simulation model. The outputs include: average assembly time of a turbine, average number of fitters working, average number of NCRs, utilisation of the assembly stand and utilisation of the fitters. The main objective functions are: minimisation of average assembly time and the minimisation of average number of fitters.

Generate interface for output variables

An Excel spreadsheet has been used to create the interface for output variables. The spreadsheet contains output variables in columns and each row represents a different turbine assembly. This representation allows users to visualise the simulation of each individual turbine assembly to assist with model validation.

Link interfaces to optimiser

A workflow has been created within an in-house optimisation platform (APOW) to enable the optimisation of the simulation model. The built in WITNESS Command Language functionality has been used to enable this. Three batch file command processes were built into a workflow in APOW to open and run the model and to save the results. A WITNESS Command Language file was created to enable the generation of replications for the simulation runs. The WITNESS Command Language file was called within the APOW optimisation loop.

Simulation run parameter selection

Before the optimisation can be carried out, a variety of simulation parameters need to be selected and set. These simulation parameters include: the number of replications (n), the length of each simulation run (m), and the window for data collection (w). Initial testing is required to determine these parameters; the testing carried out to select these values is detailed as follows.

To determine the selection of the simulation run parameters, it is vital to take into account the variation within the chosen simulation model. Testing is required to ensure that the selected parameters produce an accurate representation of steady state behaviour of the simulation model. The computation time required for different parameter settings should also be used to make an appropriate selection and to create an appropriate compromise between convergence and computation time. The next sub-sections detail the selection of the three parameters.

Run length selection

The run length parameter setting is governed by the variability within the simulation model and the other simulation parameters such as run window and number of replications. For models with a large amount of variation in output parameters, it is required to select a large run length to ensure that the model has had significant amount of time to create steady state results for analysis.

In order to select an appropriate run length for the optimisation, various tests have been carried out to assess the effect of the run length on the variation

within the output results. For each test a 100-point Optimal Latin Hypercube design of experiments has been created with 3 replications for each design point. Four different run lengths were chosen for testing: 432000 minutes, 1,000,000 minutes, 1,500,000 minutes and 2,000,000 minutes; each of the lengths represents the simulated minutes in the model.

Table 3-2 - A table to show the average variance between replicates for varying run lengths for 3 output variables

	Average assembly time	Average NCRs	Average Fitters
Average variance for run length 432000	0.481286	0.01704	0.001972
Average variance for run length 1,000,000	0.198291996	0.006453	0.001330683
Average variance for run length 1,500,000	0.15493902	0.005233547	0.001822487
Average variance for run length 2,000,000	0.097134918	0.003548472	0.001474924

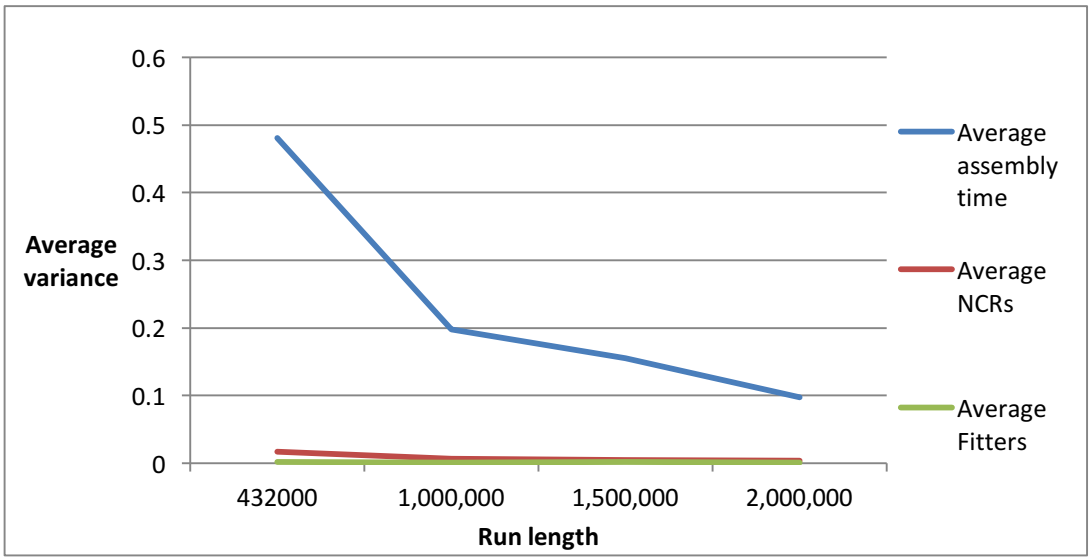


Figure 3-4 – A graph to show the average variance of each output over replicates for varying run lengths

The results of the experiments have been used to create ‘average variance’ for each output statistic and for each run length. This value is generated by calculating the variance for each design point over replicates and then the average variance of the 100 design points is calculated. Results in Table 3-2 show the average variance between replicates for all run lengths for each output variable. This information is also demonstrated in Figure 3-4.

For the ‘average assembly time’ Key Performance Indicator (KPI), the average variance reduced considerably as the run length increased. This higher run length however took considerably more computational time for each run.

To further demonstrate the effect of the run length on the variance of the results, testing was carried out to show the results of each KPI as each simulation run progresses. This testing averages results from 3 replications of an identical scenario run for 185 turbine assemblies. The time taken to run a simulation to complete 185 turbine assemblies is around 10 minutes. Figure 3-5 shows 3 graphs for the results of the 3 KPIs. It is shown that the average value for each KPI tends to converge after around 100 turbine assemblies.

To provide a compromise, it was decided to set the simulation run length to 1,000,000 simulated minutes for optimising the model. This compromise has been chosen as it requires half the computation time of the 2,000,000 run length and still has a reasonably low variance. This run length can be used for initial optimisation; further testing with a longer run length can be used if it is required.

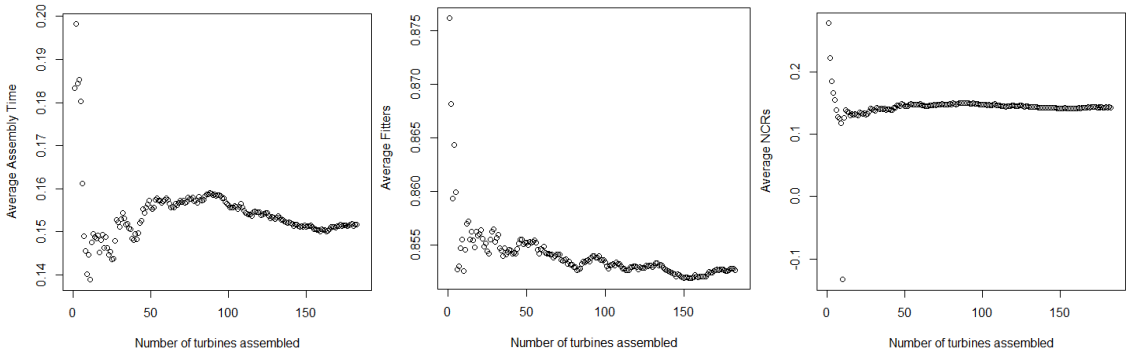


Figure 3-5 - Graphs to show KPI values (averaged over replicates) for a simulation run of 185 turbine assemblies

Number of replications

Three replications (n=3), were chosen as sufficient replications for demonstrating the variance; this is due to initial testing which was carried out to assess the effect of run length on average variance of output parameters in Figure 3-5. Although the average variance for the average assembly time for the run length of 1,000,000 was reasonably high (0.19), the variation in other KPIs was found to be less than 5%.

Run window

The window for the selection of output results was chosen to be the entire simulation run length for initial calculations. It was decided to fix this value and vary the number of replications and run length as all three values are interconnected.

Adjust simulation model after testing

After the initial simulation model was built, a pre-study was carried out, as outlined in Prajapat, Vejju, & Tiwari, (2015), to choose appropriate optimisation algorithms for this particular simulation model. A number of important observations have been made based on the results of the pre-study. This section details adjustments that have been made to the model.

It is vital to consider the effect of the optimisation on the NCRS. Initially the simulation model had been created with NCRs modelled as breakdowns. The breakdown would occur with a frequency based on 'available time', meaning time that the assembly stand is available. The 'time between failures' had been modelled using a uniform distribution; the 'time to repair' of the NCR had also been modelled with a uniform distribution. The values for 'time to repair' had been chosen based on observation and discussion with experts. The 'time between failures' was estimated based on average number of NCRs per assembly and tuned using historical data. Upon optimisation of the model, it became apparent that the number of NCRs reduced when the resource is optimally allocated; this is due to the reduction in amount of time that the assembly stand is available ('available time') when the lead time is reduced. This leads to skewed results for the lead time which are overly optimistic with respect to lead time reduction. Hence another formulation is required for the accurate calculation of the NCRs during the turbine assembly.

A rough estimation of the average number of NCRS for each turbine assembly has been used to fix the total number of NCRs for each assembly, hence only the 'time to repair' for the NCRs would vary. A function has been created in the WITNESS model to calculate the rate of NCRS. The NCR rate is calculated to be a function of the output 'AverageAssemblyTime'.

This creates complications as upon model initialisation the average assembly time is zero. Hence the function was chosen to return a fixed value if the average assembly time is 0 and to return the average assembly time divided by a fixed parameter. The fixed parameter was chosen after carrying out tuning to ensure that, on average, a set number of NCRs will occur during each turbine assembly. Figure 3-6 shows the assembly time and NCR results of two different 100 point Design of Experiments with the original formulation (left) and the new formulation (right). The axes have been non-dimensionalised. It can be seen that the new formulation ensures the average NCRs remains within a smaller range of values. The acceptable range of values for NCRs was deemed to be the average number of NCRs $\pm$20%.

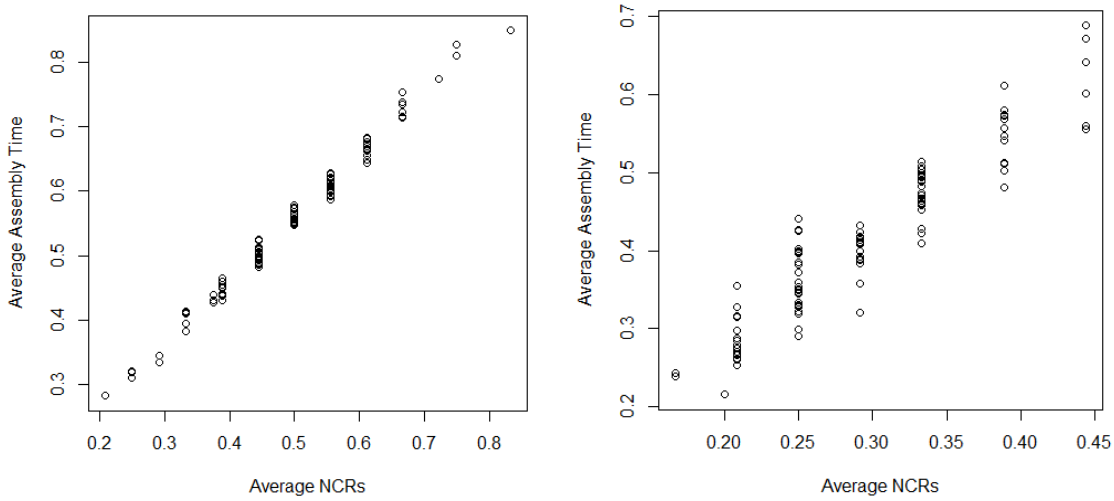


Figure 3-6 - Graphs to show the results for 2 Design of Experiments with the original NCR formulation (left) and new formulation (right)

Model validation

The validation of model assumptions has been carried out through meetings with domain experts. The model results have been demonstrated to domain experts and production mangers in order to gain model confidence. The correlated inspection approach has been used to validate the simulation model (Law, 2014). The simulation model has been run with historical system input data for the six turbine assemblies, the results were compared between the simulation model outputs and the actual system outputs. The average difference between the simulation model and true system results was found to be 0.03502%. As the results of the simulation model differed from the actual system by less than 5%, the model was deemed to be valid by domain experts.

3.3.3 Test case discussion

A number of key observations were made through the application of the framework to create a ready-to-optimise DES model.

The initial simulation model was found to lead to skewed results for KPIs such as ‘average assembly time’ when the model was linked to an optimisation algorithm. This occurred as there was interdependency between a probabilistic parameter and the ‘average assembly time’ output. This formulation led to a wide range of NCRs amongst results when the model was linked to an optimiser. For configurations which led to a smaller lead time, the number of NCRs reduced due to the reduction in available time of the assembly stand. This leads to even smaller ‘average assembly time’. Similarly, for configurations which lead to a large lead time, the available time of the assembly stand increases, directly leading to more NCRs. In order to alleviate this issue, it was necessary to fix the number of NCRs to a small range and only apply a probabilistic value to the NCR duration.

In addition, for the labour availability input variables, it was necessary to reduce the range of the input variables to the smallest range of values which covers all feasible realistic scenarios. This enables the optimiser to focus only on feasible regions of the search space to reduce wasted computation time. A matrix was created to set up the input variables for the optimisation; this matrix contains labour available per shift for each assembly task.

In order to run an optimiser on the DES model, appropriate simulation run parameters also need to be selected. In order to select these parameters a number of experiments were carried out. The main simulation parameters to be set were: simulation run length, number of replications and simulation run window. For this test case initial testing at varying run lengths with 3 replications was carried out to assess appropriate simulation run parameters. A simulation run length of 1,000,000 simulated minutes was chosen as a compromise between computation time and convergence of the KPIs. It has been noted that this run length does not provide converged KPI results, it does provide a reduced computation time which is required to enable reasonable experimentation for the optimisation. This chosen setting can be modified after assessment of the results.

3.4 Test Case 2 – Repair Facility

This section introduces another real-life test case based on a repair facility which requires reconfiguration for a new maintenance regime. This particular test case has been chosen as it demonstrates the framework on a different problem type which can be solved and optimised through DES. This test case is a layout optimisation problem, the workstations within the factory need to be rearranged to align with the new maintenance regime. The reconfiguration of factory layouts can be time consuming and expensive; hence DES is used to model and assess various scenarios to assist production managers with layout planning. The optimisation will be carried out through what-if scenario analysis, a method which was applied in a number of studies in Chapter 2 (Ghani et al., (2012), McDonald et al., (2012)); although this method may not find global optimum solutions, it will enable users to detect local optima amongst a set of pre-selected scenarios. The framework outlined in Figure 3-1 is applied to this test case to produce a ready-to-optimise DES model. The following subsections present the application of the framework to the test case.

3.4.1 Problem description

A repair facility carries out the periodic repair of parts. The repair facility requires reconfiguration to align with process changes for every major repair cycle.

Factory managers would benefit from the ability to test various layout configurations and to assess them with respect to KPIs. In particular the distance travelled by the parts can be used as an indicator to identify optimal configurations, as materials handling processes can account for 12-50% of manufacturing costs (Kulwiec, 1985). Additionally, the utilisation of each workstation and of the workers is of interest to management, as this information will enable reconfiguration to reduce bottlenecks and the elimination of waste from the repair process.

Process description

The flowchart in Figure 3-7 shows the repair process for each part for a specific repair program. The part is disassembled into three main sub-assemblies during the repair process. The sub-assemblies are then re-assembled, tested and shipped in batches.

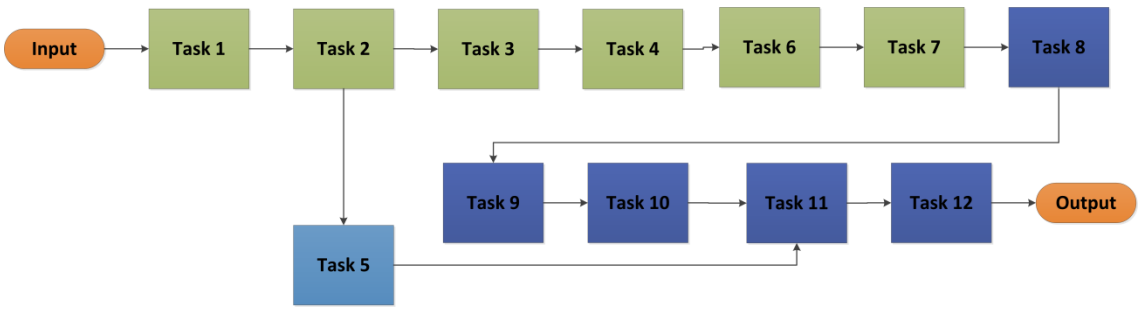


Figure 3-7 – A diagram of the repair process

Each part has an associated type. The parts must arrive and leave the facility in two different set sequences based on their part types. Hence, during the repair process, the sequence of part types is changed. This change of sequence has been indicated in Figure 3-7 by the colours of the repair tasks. The green tasks represent the parts moving along in the input sequence, the blue tasks show the parts moving through the process in the output sequence. The task in light blue is a sub-assembly repair task.

Although the sequence is set, the factory managers and staff often use their own knowledge of the priorities of the parts to override the sequence. Managers and workers use their initiative to put through the parts which have a higher priority due to being required early in the output sequence.

The facility is rectangular in shape and has a wall down the centre. There are four cranes within the factory, two on either side of the wall. Effectively each crane services one quadrant of the factory, although having two on each side also provides more flexibility and cover during breakdowns. There are four gaps

in the wall, which are used to move parts between the two sides of the facility. The movement of parts between the two sides of the factory is more complex as this may require cranes from both sides and other equipment such as trolleys.

There are a number of paths within the factory which are used by the workers. The movement of parts from one side of the wall to the other must also be carried out on one of four paths. Other crane based movements can occur in any direction and they do not require a path.

The movement between different locations uses a variety of materials handling resources. For instance, parts which move from one side of the wall to the other often require two cranes, one from each quadrant, and a trolley to move the part. With the complexity of different materials handling equipment the time taken for the movement is often complex to calculate and can be lengthy. Typically, the time taken for movements will vary from 0-8 minutes, more complex movements may take longer.

3.4.2 Application of framework to create ready-to-optimise DES model

The framework introduced in Figure 3-1 has been applied to this test case to create a ready-to-optimise DES model. The following sub-sections detail the application of the framework to the repair facility test case.

Identify input and output variables

Based on the process description, discussions with stakeholders and the literature review, the input and output variables have been defined. Variables such as workstation cycle times, utilisation and WIP have been chosen based on the literature review (Li et al., (2011), Cave et al., (2006), Hugan (2001)). Variables such as movement times, shift patterns, input/output sequences, workstation positions and distance travelled by parts have been chosen based on discussion with production managers. Input variables include: workstation

cycle times, movement times, shift patterns and part sequences. Key output variables include: distance travelled by parts, WIP, parts completed and utilisation of workers and workstations. Table 3-3 lists all of the input and output data.

Table 3-3 - A table to show the input and output data within the Excel spreadsheet for the repair facility test case

Input data	**Output data**
Cycle times of workstations	Distance travelled by parts
Movement times	Weeks and parts completed
Shift patterns and overtime	Utilisation graphs of workstations/ labour
Input sequence of types	New location of workstations
Output sequence of types	Parts in final buffer
Position of each workstation	WIP

Data collection

Data has been collected for all of the input variables by factory workers. There are 12 main workstations within the facility. Average cycle times for workstations were collected. Table 3-4 shows the workstations, cycle times (non dimensionalised) and number of workers at each workstation. Task 5 is split into two identical workstations carrying out the same task. Task 7 can be carried out at one of 4 identical stations; the one worker is shared amongst all 4 workstations. Task 7 has 4 different cycles, two of which can be automated. Tasks 3 and 10 share one worker; this is due to the smaller cycle times of these tasks. Labourers were found to work an eight-hour shift pattern for five days a

week with no overtime. Input and output sequences for the part types were provided, although it was noted that these may be adjusted ad hoc based on experience of production managers. Average movement times were provided. A 2D CAD drawing of the factory was provided to show the position of each workstation. This CAD drawing has been used as the backdrop for the simulation model to ensure the locations and distances are modelled to scale.

Table 3-4 - A table to show the cycle times, number of stations and workers for each process. * indicates the worker is shared amongst workstations.

Tasks	Number of stations	Cycle times (scaled)	Number of workers
Task 1	1	12	1
Task 2	1	12	1
Task 3	1	12	1*
Task 4	1	38	3
Task 5	2	90	6
Task 6	1	39	2
Task 7 cycle 1 cycle 2 cycle 3 cycle 4	4	0 100 0 71	1
Task 8	1	38	1
Task 9	1	38	3
Task 10	1	12	1*
Task 11	1	38	2
Task 12	1	46	1

Generate interface for input variables

An Excel spreadsheet has been created to enable users to input key information into the simulation model, and to visualise the outputs in order to assess the performance of the simulation model. This interface has been created to enable production managers to use and modify DES models easily, particularly as Excel is a tool which production managers are familiar with. Data which is input into the spreadsheet will be brought into the simulation model when the model initializes.

The spreadsheet also acts as a direct interface for optimisation algorithms. The input variables can be changed within the Excel spreadsheet by an optimiser.

Within the factory, the managers and workers use their knowledge of the overhaul process and the input and output sequences to change and adjust the sequence as required. Often the workers will prioritise certain parts and ensure the chosen parts complete the repair process before others; this is required in particular for parts which appear early in the output sequence. The input and output sequences have been built into the Excel spreadsheet as input variables. The generated arrays for input and output sequence can be used as optimisation inputs to identify optimal sequences to maximise the KPIs.

As the model has been created to allow all of the workstations to change location, the equipment requirements will also change whenever the layout is changed. This leads to new movement times for any new layout configurations. Due to timing complexities it was decided not to simply base timings on direct distances between locations. Instead a fully flexible array has been created in Excel to allow users to input estimated movement times for each movement. This input array will enable users to configure the model to create realistic representations of the factory, rather than relying on a distance factor which does not account for movement complexity.

An Excel sheet has been used to generate a table with the same information as in Table 3-4. This has been used as the interface for input of number of

workstations, cycle times and number of workers. Shift pattern information has also been included in the sheet.

A ‘SetPosition’ worksheet has also been created to allow users to input the planned layout of the model based on workstation coordinates. This is key functionality which must be enabled in order to optimise the facility layout.

Build matrix in DES model for input variables

The parts arrive and leave the factory in different sequences depending on the part types. Within the simulation model, two arrays have been created to represent the input and output sequences. Similarly, matrices were created to contain: number of each workstation, cycle times, number of workers, movement times and coordinate locations for workstations.

Build main simulation model

A 2D DES model has been built using WITNESS simulation software to enable factory managers to assess various configurations of the factory. The simulation model creation includes modelling machines, buffers, sub-assemblies, sequences, movements and materials handling equipment requirements.

Within the simulation model 12 machines have been created to represent all of the workstations within the repair process. The workstations can be selected and moved easily to change the layout of the facility. The workstations have been modelled as single cycle machines with the exception of Task 7. The four workstations for task 7 are modelled as multi cycle machines, each has 4 cycles for the manual and automated processes. Task 5 has been modelled as two workstations which carry out identical tasks. All of the machines are modelled to require workers to carry out the tasks.

Various buffers have been created within the simulation model to represent areas within the factory which are used to store parts. The buffers which have been created are as follows:

- Before workstation 1

- After workstation 7
- Before workstation 9
- After workstation 9
- Before workstation 5
- After workstation 5
- After workstation 11
- Before shipping

Three main sub-assemblies are used within the simulation model. The parts arrive in the model in batches. They are put into the first buffer on arrival. During the main tasks, the parts are disassembled, maintained, reassembled, tested and shipped.

The sequence change has been built into the simulation model using a built in ‘match’ rule within WITNESS, Task 8 will only pull in parts in the correct output sequence as defined by the user.

The built in ‘SetPosn’ function within WITNESS has been used to select the X and Y coordinates within the Excel worksheet and to use them to set the factory layout to the new configuration. A button has been added in the simulation model toolbar named ‘SetPosition’ which will take the values from Excel and change the layout according to those values. This functionality is particularly useful as users can test many configurations and then revert back to a baseline.

A pool of workers has been modelled to represent the main labourers within the factory. Task 7 work areas are attended by one specific worker; so a separate worker has been modelled for this. One worker is shared between task 3 and 10; so a separate worker has been modelled just for these two workstations. All workers are modelled to work on a set eight-hour shift pattern.

A key requirement of the simulation model is full flexibility to move any of the workstations to any location in order to test various configurations. The modelling of the movements between workstations is therefore modelled flexibly

using a construct in WITNESS called a pseudo path. These show movement ‘as the crow flies’ between locations and the distance can be incorporated into journey timings. When good or optimal solutions are found, fixed paths for movement can be detailed as needed, with full path constructs replacing the pseudo paths.

The material handling equipment has been modelled as variables within the WITNESS model as a simple approach that allows for a great deal of flexibility to attach and detach equipment at any stage in the process. Figure 3-8 shows the location of the four quadrants within the facility, each of which is served by one crane. Quadrants 1 and 2 are on one side of the wall and quadrants 3 and 4 are on the other side.

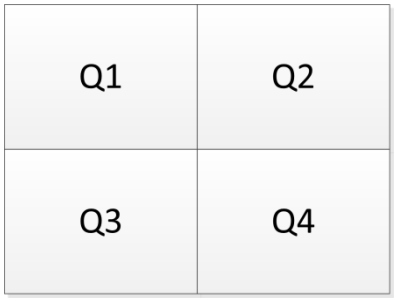


Figure 3-8 - A diagram to show the location of the 4 quadrants within the factory

Each of the four cranes has been modelled as a separate indicator variable (taking the value of 1 or 0). The trolleys have been modelled as a pool of four trolleys which can be accessed. In order to calculate which material handling equipment is required, the model must first check in which quadrant a particular movement begins and in which quadrant the movement ends; this calculation is carried out in the Excel spreadsheet. The coordinates of the workstations within Excel have been leveraged to calculate quadrants. There are 15 major movements within the factory.

The start and end quadrant of the movement will govern which materials handling equipment is required. There are 10 different combinations of requirements which could occur based on the starting and finishing locations of parts. The numbers 1-10 are used to index which equipment is required. The index of materials handling equipment requirements is as follows:

1. Only Q1 crane is required
2. Only Q2 crane is required
3. Only Q3 crane is required
4. Q4 crane and 1 trolley are required
5. Q1 and Q2 cranes are required
6. Q3 and Q1 cranes are 1 trolley required
7. Q1, Q3, and Q4 cranes and 1 trolley are required
8. Q3, Q1 and Q2 cranes and 1 trolley are required
9. Q2 and Q4 cranes and 1 trolley are required
10. Q4 and Q3 cranes are required

A look up matrix has been created in Excel which shows the start quadrant as rows and the end quadrant as columns. An index function has been written in Excel to check the start and finish quadrant of each of the 15 movements and to return the corresponding number of the movement requirement.

The result is a 15x1 array. Figure 3-9 shows the look up matrix, the values within the matrix relate to the index above.

	Q1	Q2	Q3	Q4
Q1	1	5	6	7
Q2	5	2	8	9
Q3	6	8	3	10
Q4	7	9	10	4

Figure 3-9- A look up matrix to show the different possible factory movements. Values relate to the list above

An array named ‘MovementRequirement’ has been created within the simulation model which extracts the values from the Excel array for each of the 15 main movements. Within the simulation model each time a movement is required, the corresponding ‘MovementRequirement’ number is passed into the ‘AcceptMove’ function. The function subsequently checks if the matching materials handling equipment is available. If the equipment is available the equipment variable is decremented and the move is carried out, otherwise the part will wait.

In order to check the availability of the required materials handling equipment an ‘AcceptMove’ function has been created; this function is executed when a part is input into any machine. This function uses the task number of the process to assess which movement is required. The Excel spreadsheet is then used to assess which corresponding material handling equipment is required. The function ‘AcceptMove’ is used to check the availability of the required equipment; if the equipment is available, the corresponding variable will be decremented and the move will be carried out, otherwise the part will wait.

A corresponding ‘FinishMove’ function has also been created to increment the materials handling variables after each movement has been completed within the simulation model.

Build calculation of outputs in simulation model

The outputs for the simulation runs have been added to the simulation model. The outputs include: WIP, throughput, distance travelled by parts, utilisation of workers, utilisation of workstations, parts completed and parts in output buffer amongst others.

Generate interface for output variables

An Excel workbook has been created in the Excel spreadsheet for input variables, this workbook stores the output variables from the simulation runs. The Excel workbook contains the KPIs as columns and each row contains

output values from a simulated week. A week was chosen as the timeframe for the results as there is a fixed weekly target for the manufacturing line. Output variables in the columns include: number of parts completed, week number, WIP, number of parts in final buffer, utilisation of workers, utilisation of workstations and distance travelled by parts.

A ‘GetPosition’ worksheet has been created within the Excel spreadsheet which is used to print the coordinate locations of all of the elements in the simulation model. The built in WITNESS functions ‘GetXPosn’ and ‘GetYPosn’ were used to print the X and Y coordinates of the location of each workstation. A new toolbar has been created in the model with a ‘GetPosition’ button. Users can rearrange the workstations within the simulation model, and then click the ‘GetPosition’ button and see the new coordinates in the Excel spreadsheet.

The X and Y coordinates of each location have then been used to estimate the distance travelled by parts during the overhaul process. For each of the main movements within the factory the direct distance is calculated between workstations. Pythagoras’ theorem is used to calculate the direct distance between workstations. The sum of all of the individual distances is then used to estimate the total distance travelled. The total distance travelled is a main KPI to be used by managers to assess factory layouts, as this effectively minimises material handling.

Link interfaces to optimiser

The input interface has been linked with in-house optimisation platform, APOW, to enable optimisation. The interface can be used with the optimiser or directly by users to perform manual optimisation.

Select simulation run parameters

As there are no probabilistic parameters within the simulation model, it was expected there would be no need to use replications for the results. Testing was carried out to vary the random number sequence within the simulation model and the results showed no variation in output parameters, hence no replications have been used for the testing

Simulation run length has been determined through testing. A set of experiments have been carried out; the KPIs have been printed every simulated week to assess the variation and identify a run length. The majority of KPIs had very little or no variation between their values for each simulated week, the distance travelled did not change between simulated weeks. Fifteen KPIs were found to have variation in output results over the initial simulation period. In order to assess the best place to take results, the simulation was run for 53 simulated weeks. The 15 KPIs were printed every simulated week and a graph has been created, as shown in Figure 3-10, to show the convergence of each KPI over the 53 week period. Based on the results, 25 weeks was chosen as a reasonable balance between simulation run length and convergence of results. Hence for testing purposes, 25 simulated weeks was chosen as the best run length. The entire run length has been used in the run window as the KPIs are reasonably converged after 25 simulated weeks. The model was also found to run reasonably fast, taking around 20 seconds to finish each run.

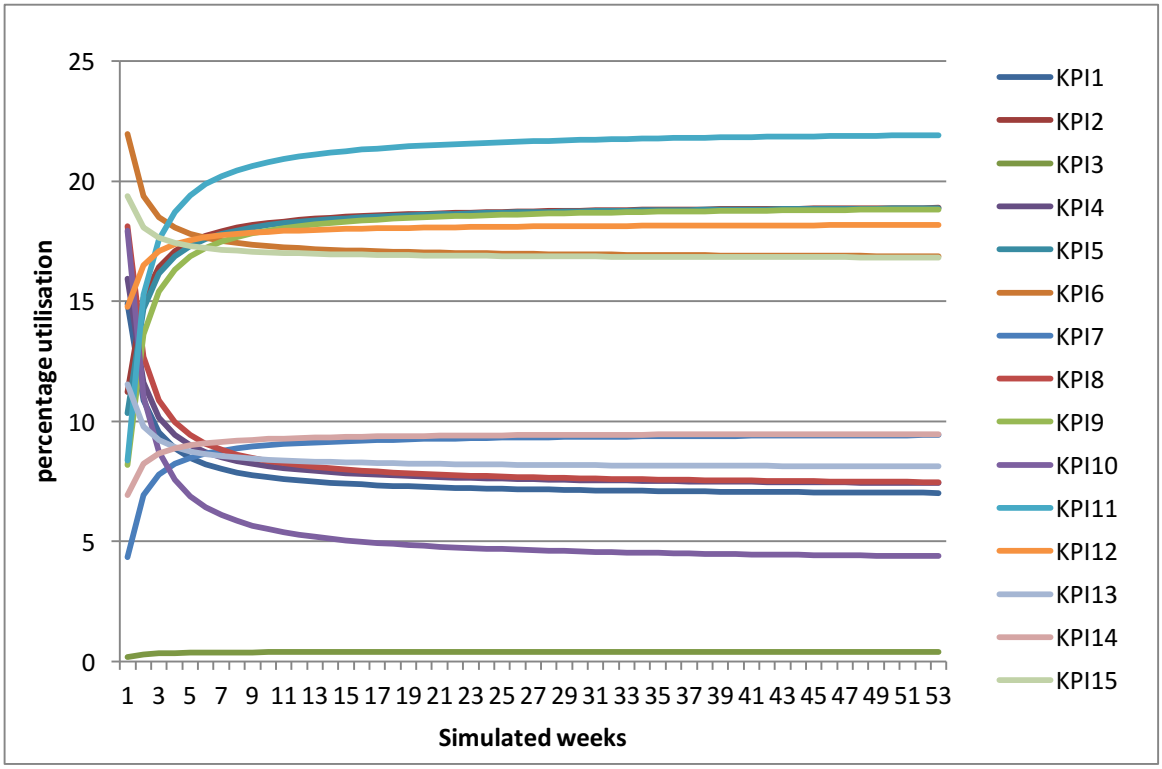


Figure 3-10- A graph to show the percentage time spent waiting for one task

Adjust simulation model after testing

The simulation model and optimisation link have been tested initially. During the test, linking the 'set position' input variables to an optimisation algorithm led to infeasible solutions; in particular, workstations were placed on top of one another. This is due to lack of geometrical constraints within the setup. It was deemed to be too complex to create a set of geometrical constraints for the optimisation. Hence, the generated interface is for carrying out what-if scenario analysis on the simulation model. The number of realistic layouts is reasonably small; hence the interface for what-if scenario analysis is highly useful for experimentation. A selection of 5 possible layouts have been created and saved into the Excel spreadsheet. These layouts are the potential solutions for the decision makers to choose between. This functionality allows users to leverage their knowledge of the facility and process to perform quick testing of realistic layouts, in order to optimise the final layout.

Initial testing has been carried out to assess and validate the model. Four different scenarios were tested to compare the model results with different inputs. All scenarios were run for 25 simulated weeks to assess the distances travelled, number of batches shipped and production rates. Table 3-5 shows the scenarios tested.

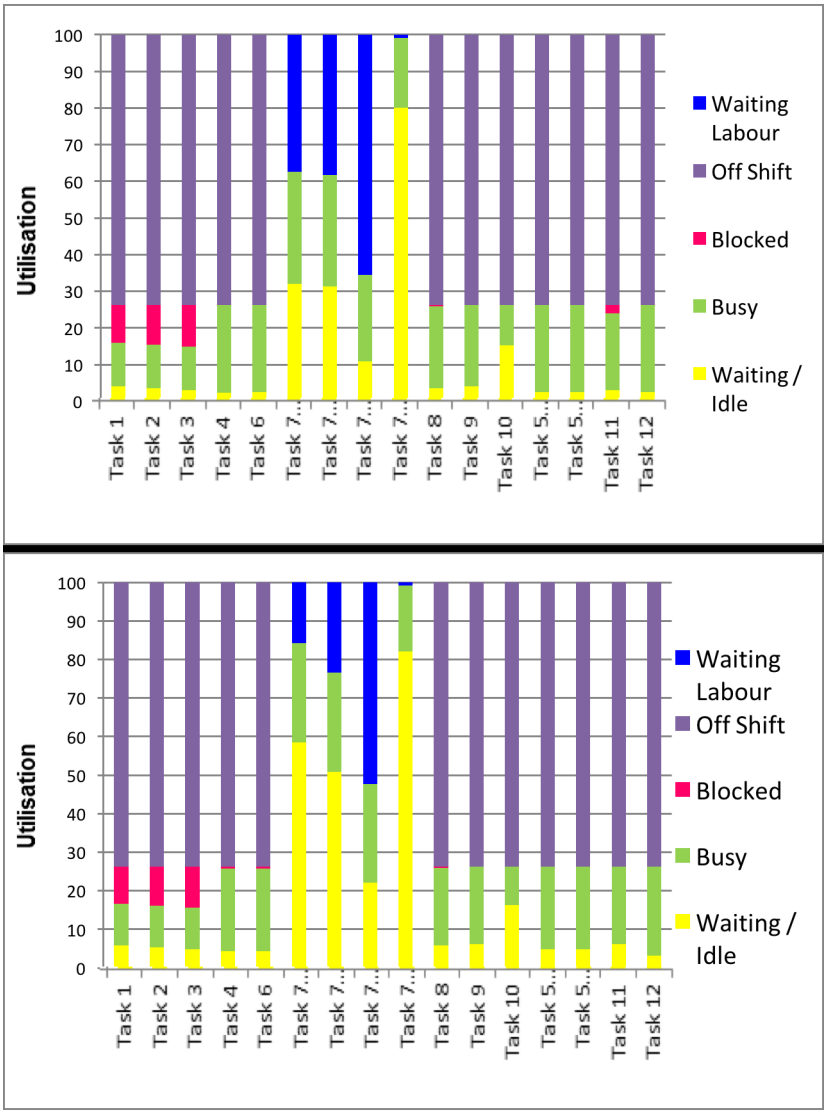
The first scenario tested was the current production scenario within the factory. Cycle times and movement times were added into the Excel spreadsheet. The model results showed that the distance travelled by the main assembly as 212 metres. The utilisation of the workstations is shown in the top bar chart in Figure 3-11. The model completed 22 batches within 25 weeks, with the Work In Progress (WIP) rising to 85 within the final simulation week; this value represents an additional batch waiting for overhaul. The results showed that, with the first cut input data, the model did not match the factory production rate of 1 batch a week (with an exception of the ramp up period in the first week).

Table 3-5- A table to show the set up for experiments- all experiments were run for 25 weeks

Scenario	Input Variables	Output Variables
1	Current production scenario with input variables as shown in Table 3-4	Distance travelled 212m 22 batches completed WIP in final week 85
2	Current production scenario with 10% reduction in cycle times	Distance travelled 212 m 24 batches completed WIP in final week 55
3	New layout with 2 workstation locations swapped. Input variables as shown in Table 3-4.	Distance travelled 302m 22 batches completed WIP in final week 85
4	Scenario 3 with 10% reduction in cycle times	Distance travelled 302m 24 batches completed WIP in final week 55

A second scenario was tested with an identical configuration to the first model, but with a 10% reduction in the cycle times of all workstations. This has been carried out to attempt to match the production rate of the factory. The results showed the factory completed 24 batches within the 25 week time period. The production rate was stable with 1 batch produced per week with static WIP over the 25 weeks.

The utilisation is also shown in the lower bar chart in Figure 3-11. The idle time of stations has increased; this is particularly clear for the 4 workstations for Task 7. In addition, the final assembly stage is no longer blocked.



[redacted]

A third and fourth scenario were also tested to assess the effect of changing the location of 2 workstations. Two of the workstations have been moved to a different part of the factory. The new distance travelled by parts for scenario 3 and 4 was found to be 302 meters.

The third scenario used the identical inputs to the first scenario, with original cycle times. The movement times were updated to account for the complexity of the new movements. Results produced were identical to the results from the

first scenario, with identical utilisation of all machines, as shown in the upper bar chart in Figure 3-11, and a production rate of 22 batches in the 25 week period. The fourth scenario used identical inputs to the third scenario with 10% reduction in cycle times. Results produced were identical to the results from the second scenario, with identical utilisation of all machines, as shown in lower bar chart in Figure 3-11, and a production rate of 24 batches for the 25 week period.

Model validation

Results suggest that the model production rate is somewhat insensitive to both movement times and to the factory layout with current data inputs. However, the model is sensitive to changes in cycle times, as would be expected given the long duration of some tasks. The improved match to observed data in model for scenario 2 indicates that the cycle times may be less than initially entered into the model. Discussions with production managers led to the conclusion that the movement times were being accounted for twice, once in the cycle times and again in the movement times variables. Hence the adjustment to create scenario 2 was deemed to be a valid model and the original simulation model was not deemed to be valid.

3.4.3 Test case discussion

A DES model has been created of a repair facility in WITNESS; the model enables factory managers to test various layouts and configurations in order to optimise production. Various features have been built into the simulation model and the corresponding Excel spreadsheet to allow users to carry out configuration testing and to identify optimal solutions. These features include functions to automate layout planning, in order to facilitate manual layout optimisation.

It was identified that the linking of optimisation algorithms to the model for layout optimisation would result in infeasible layouts; however, the model can be manually optimised by decision makers through the Excel interface.

3.5 Discussion

Through the application of the framework to the two test cases, a variety of important observations have been made.

For the test cases, the simulation run parameter selection had different results. For the repair facility test case, as there were no probabilistic parameters, no replications were required. For the assembly test case, an interdependency was identified between the input and output parameters. This interdependency led to skewed results when the optimiser was applied; hence the model formulation had to be modified.

Both models demonstrated the requirement to compromise between computation time and convergence of results. The repair facility model was found to converge to stable results faster than the assembly test case. This is expected to be due to the amount of variability within the assembly test case and as there are no probabilistic parameters within the repair facility test case.

Matrices have been used to store input variables and link with an optimiser for both test cases. For the assembly test case a matrix of input variables has been used. For the repair facility test case, a matrix of locations for 5 different layouts has been used in Excel to enable manual layout optimisation for a set of predefined solutions. Additionally, two arrays have been used to list input and output sequences. It was found that, for the repair facility test case, layout optimisation is not possible by simply applying an optimiser to the X and Y coordinates of workstations, as there are no geometrical constraints included in the optimisation. Hence it is proposed that for layout optimisation a set of predefined solutions should be created to enable manual optimisation using what-of scenario analysis. Users could also use the repair facility DES model to apply scheduling optimisation for input and output sequences, however this would also require a method to ensure each solution generated by the optimiser permutes the sequence. For the resource allocation optimisation, an optimisation algorithm can be directly applied to the model without addition of further constraints.

For both test cases, the number of potential solutions had to be reduced as much as possible in order to relieve the computational cost of running the simulation models. In the repair facility test case, a list of potential solutions was created in an Excel spreadsheet which could be chosen by the user. This creates a finite number of solutions from which to optimise. This enables experts to input information regarding realistic solutions into the modelling and experimentation process to reduce computation time. In the assembly test case, allowable values were used to limit the optimiser to choosing realistic values for the number of operators available for each assembly task on each shift.

The framework to generate ready-to-optimise DES models has been applied to two different test cases in this chapter: the repair facility test case and the turbine assembly test case. These two particular test cases have specific requirements. The repair facility test case is a layout optimisation problem and the turbine assembly test case is a resource allocation problem. It is expected that for other types of problems, there may be requirements for specific modifications to enable a link with optimisation through algorithms or what-if scenario analysis. For instance, problems which require testing of different assembly sequences (Marsh et al., (2010)) may require a method to enable perturbation in the order of parts. This framework should be tested with other problem types to ascertain the suitability of the framework for a range of generic problem types.

3.6 Chapter Summary

This chapter presented a framework to enable the development of ready-to-optimise DES models. The framework has been applied to two different test cases of different manufacturing processes. The assembly test case aims to reduce the ‘average assembly time’ through resource allocation optimisation. The repair facility test case is a factory layout optimisation which aims to identify the best layout to reduce distance travelled by parts.

The framework included: generating an interface for input variables, building a matrix within the DES model to hold input variables, building the simulation model, creating an interface for output variables, selecting simulation run parameters and adjusting the simulation model after testing.

For both test cases it was found that the simulation run parameters had to be defined through experimentation. It was found that the repair test case model with less variability converged to steady state faster than the assembly model. In addition, the search space had to be reduced as much as possible for both test cases to reduce computation time.

CHAPTER 4 – MULTI-OBJECTIVE OPTIMISATION OF DES MODELS

The literature review has identified meta-modelling and multi-objective optimisation as growing areas of interest. Many studies have used DES to model and improve chosen objectives with respect to resource allocation; however, limited work has been carried out to simultaneously optimise multiple conflicting production objectives using meta-models. In this chapter, a novel meta-modelling methodology is presented for multi-objective optimisation of DES models. The meta-modelling methodology involves the creation of two linear regression models of the output objectives to optimise simultaneously. The methodology is tested on the turbine assembly test case, which was introduced in Chapter 3.

This chapter aims to achieve the following:

- Present a framework for multi objective optimisation of DES models using meta-modelling
- Validate the optimisation methodology on 3 different experiments
- Identify the benefits and drawbacks of the proposed methodology

4.1 Background

In Negahban & Smith, (2014) simulation optimisation and meta-modelling of manufacturing systems is identified as an area which provides broad future research opportunities. In this section, this research opportunity is exploited by creating a methodology for multi-objective optimisation of a turbine assembly DES model using meta-models.

Performance objectives for assembly lines are always conflicting and competing with each other and cannot be met at the same time (Zeng et al., 2012); hence optimisation with respect to multiple objectives is vital. Research papers

identified in the literature review have largely dealt with multiple objectives through aggregation (Cochran & Kaylani, 2008; Mehrsai, Karimi, et al., 2013b; Neira Dueñas et al., 2007; Zeng et al., 2012); this can be mainly attributed to the large computation power and time required to run multi-objective optimisation algorithms. Mendes et al., (2005) test various demand scenarios for multiple objectives without aggregation, however they carry out 'what-if' scenario analysis instead of applying an optimisation algorithm. This section aims to apply multi-objective optimisation algorithms directly to create a Pareto front of solutions.

Analytic models are becoming more commonplace as DES models can be time consuming and computationally costly to run (Pradhan & Damodaran, 2009). Various analytic methods have been proposed as alternatives to DES models; these include: system dynamics (Yang et al., 2013), normal approximation methods (Desmet et al., 2010), regression models (McHaney & Douglas, 1997) and other deterministic models (Manns & Elmaraghy, 2009). Meta-models are a particular type of analytic model, a meta-model can be defined as a 'model of a model' (McHaney & Douglas, 1997). In McHaney & Douglas, (1997) authors propose a meta-modelling method for the operator allocation problem for an assembly line. The authors find that the meta-model is capable of handling resource allocation questions rapidly and efficiently and they conclude that, in systems where linear relationships can be demonstrated, meta-modelling can be a useful tool.

This chapter presents a framework for optimising DES models which have multiple objective functions. One of the key challenges with DES is the computational effort required to optimise the model. The framework proposes a method for multi objective optimisation which reduces the computational effort required for optimising the model using meta-models. The framework builds on the work of McHaney & Douglas (1997) by creating two linear regression meta-models which are subsequently simultaneously optimised using multi-objective optimisation. The Pareto Front is then re-run using the actual simulation model to find points in this region of the search space.

4.2 Framework for multi-objective optimisation of DES models

The proposed framework is demonstrated in Figure 4-1. The framework involves DOE generation, linear regression meta-model creation, multi-objective optimisation of the meta-models and subsequent comparison of the meta-model results against actual simulation results. This methodology is novel and provides a contribution to literature as it extends work previously carried out by (McHaney & Douglas, 1997) to include linear regression meta-modelling for multiple objectives and subsequent multi-objective optimisation of the meta-models.

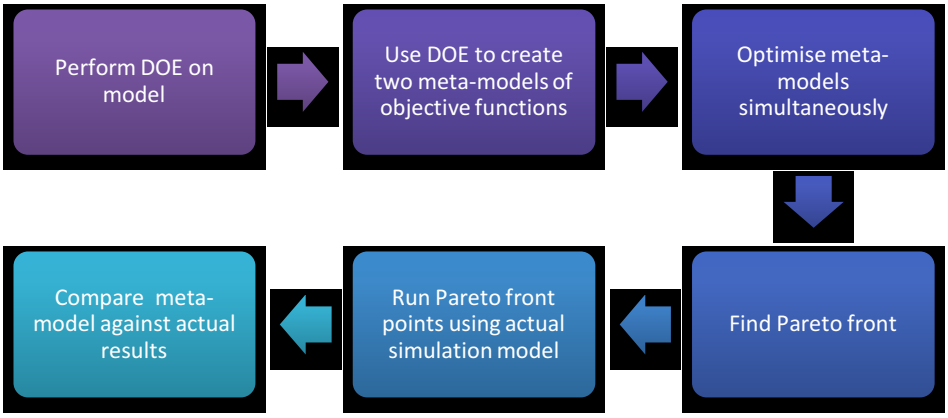


Figure 4-1 – A multi-objective optimisation framework for DES models

The linear regression meta-model is created using the DOE input matrix X and the output variables from the simulation model y. The linear regression model has the format shown in equation 4.1 where the B_n values represent the coefficients of the input variables X_n in the regression model. The coefficients are calculated using input variables matrix and output variables array using equation 4.2.

$$y = B_0 + B_1X_1 + B_2X_2 + \ldots B_nX_n \quad \textbf{(4-1)}$$

$$B = (X^TX)^{-1}X^Ty \quad \textbf{(4-2)}$$

4.3 Test Case- optimisation of a turbine assembly

The turbine assembly test case has been chosen to test the multi-objective optimisation framework in section 4.2. Linear regression models have been chosen as the meta-modelling method due to the model formulation for the assembly process, as the time taken for assembly is a linear function of the available workers. This test case has two objectives to be optimised: 'average assembly time' and the 'average number of fitters'. The 'average assembly time' objective function optimises the average lead time for assembly of turbines. This objective has been chosen as it is a time-based objective and it was demonstrated to be one of the most popular objectives in the literature review used in a range of papers (Marsh et al., (2010), Cochran & Kaylani (2008)). The 'average fitters' objective function minimises the total assignment of workers to minimise the assembly cost; this objective function has been chosen as a resource assignment objective which is similar to the formulation used by McHaney & Douglas, (1997).

Prajapat et al. (2015) present a study to test various multi-objective optimisation algorithms directly on a DES model of the turbine assembly process test case. The results of the study suggest that the chosen algorithms perform similarly, however Pareto Archived Evolutionary Strategy (PAES) gives poor results and OMOPSO, GDE3 and NSGAII perform the best. The GDE3 and NSGAII algorithms are shown to provide the best convergence to the Pareto front. In this chapter, the turbine assembly process test case is used and 72 decision variables are applied to ensure global optimisation is carried out. Due to the large search space, the framework from section 4.2 is applied, with the aim to

relieve the computational cost of directly running multi-objective optimisation algorithms.

The multi-objective optimisation framework for the chosen test case is proposed as follows: a 5000-point Optimal Latin Hypercube (OLH) with a chosen number of replications is carried out, linear regression models are created for each output objective, multi objective optimisation is carried out on the regression (meta) models (using the APOW platform), the Pareto front is created from results, the Pareto front points are run using the actual simulation model and the results are compared. This process is shown in Figure 4-2.

This novel framework has been applied on the three different experiments. For all experiments the run window was chosen to be the entire run length. Table 4-1 shows the differences between each experiment including, simulation run parameters chosen for each experiment. For the final experiment, the cycle times have been replaced with probability distributions to test the results with a larger number of stochastic parameters.

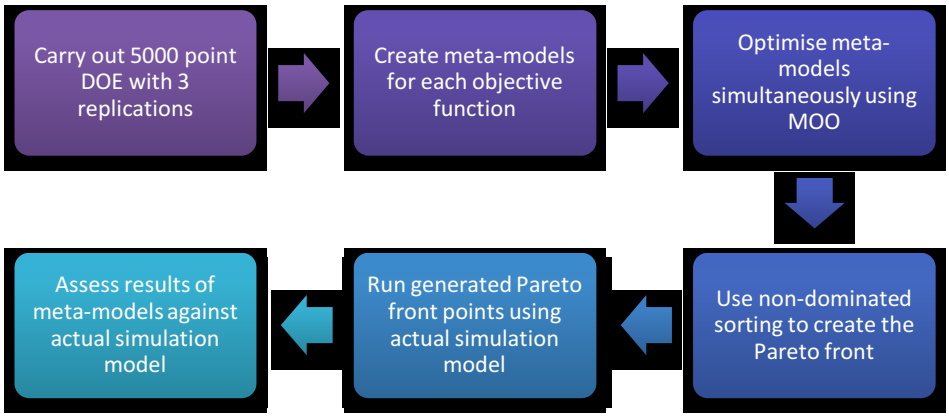


Figure 4-2 - A diagram to show the optimisation methodology followed for each experiment for the turbine assembly test case

Table 4-1 – A table to show the run parameters for the three experiments

	Run length	Replications	Distribution cycle times?
Experiment 1	1,000,000	3	No
Experiment 2	2,000,000	1	No
Experiment 3	2,000,000	3	Yes

4.3.1 Experiment 1

An initial 5000 point OLH Design of Experiments was carried out with three replications for each point, with a run length of 1,000,000 simulated minutes for each simulation run. The 5000-point DOE results were then used to create two different linear regression models to represent the relationship between the input variables and the two objectives of interest (average assembly time and average fitters). The regression models had the following statistical results.

The linear regression model for the 'average fitters' KPI had the following statistical results:

- Residual standard error: 0.04436 on 4927 degrees of freedom
- Multiple R-squared: 0.9138, Adjusted R-squared: 0.9125
- F-statistic: 725.5 on 72 and 4927 DF, p-value: < 2.2e-16

As the p-value is much less than 0.05, we reject the null hypothesis that $\beta 1 = 0$. Hence there is a significant relationship between the independent and dependent variables in the linear regression model of the data set. The R-squared value is high suggesting that the model accounts for a high proportion of the variability. The plots in the upper half of Figure 4-3 show the standardised residuals for the model, generally the model seems to fit well and there is no indication of requirement of a higher order model.

The linear regression model for the 'average assembly time' KPI had the following results:

- Residual standard error: 0.8821 on 4927 degrees of freedom

- Multiple R-squared: 0.7938, Adjusted R-squared: 0.7908
- F-statistic: 263.5 on 72 and 4927 DF, p-value: < 2.2e-16

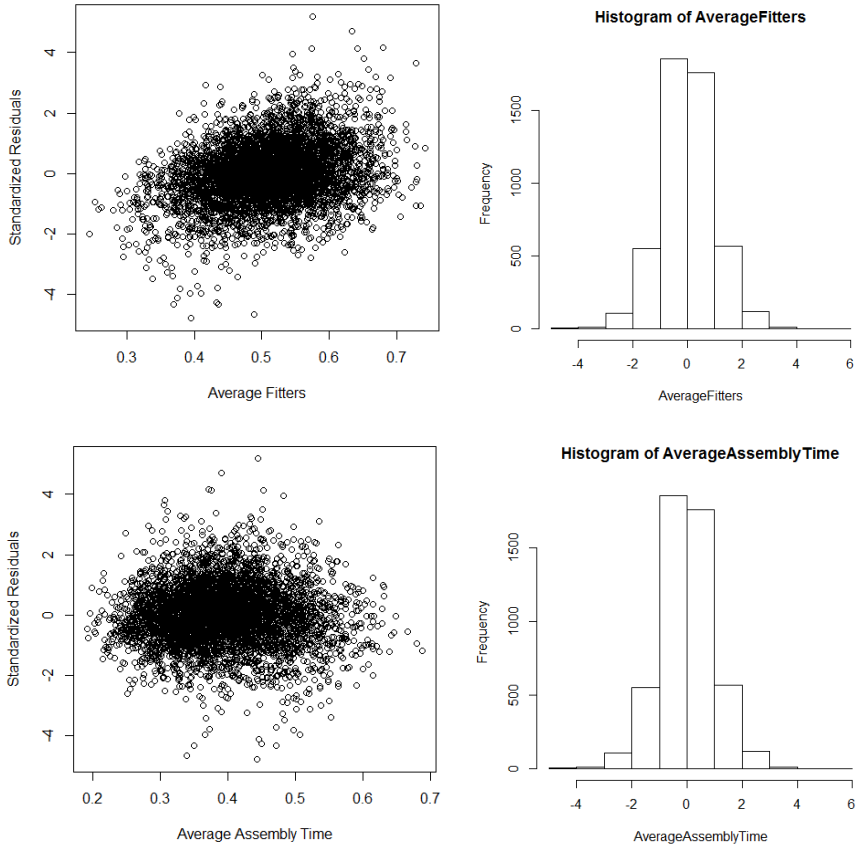


Figure 4-3 - Residual plots and histograms for the average assembly time and average fitters for experiment 1

As the p-value is much less than 0.05, we reject the null hypothesis that β1 = 0. Hence there is a significant relationship between the independent and dependent variables in the linear regression model of the data set. The R-squared value is reasonably high suggesting that the model accounts for a high proportion of the variability. The model has a high residual standard error suggesting that the model does not fit the data very well. Standardised residual plots have been created to visualise the results. For the purposes of an initial

test these meta-models were found to be reasonable. The plots in the lower half of Figure 4-3 show the standardised residuals for the model.

Following the creation of the two meta-models, the meta-models were optimised simultaneously using the NSGAII algorithm through the APOW platform. The algorithm was run for 50,000 evaluations with a population size of 100. The results were non-dominated sorted to create a Pareto front of 1544 points. The Pareto-optimal points were then taken and re-run using the original simulation model. The graph in Figure 4-4 shows the two series of points: the Pareto front points created using the regression model and the corresponding points which were run using the actual simulation model.

A consistent difference in the regression results and the actual results is shown; there is a significant gap in the graph in Figure 4-4. There could be a few potential reasons for the consistent difference including: not enough points in the DOE to capture the model, excessively short run length and an inadequate run window which includes the warm up period.

In order to improve the accuracy of the regression model prediction, it was decided carry out the same methodology again for an increased run length. The results in Table 4-1 suggest that an increase in the run length to 2,000,000 simulated minutes should decrease the average variance in KPIs. This is carried out in the next experiment.

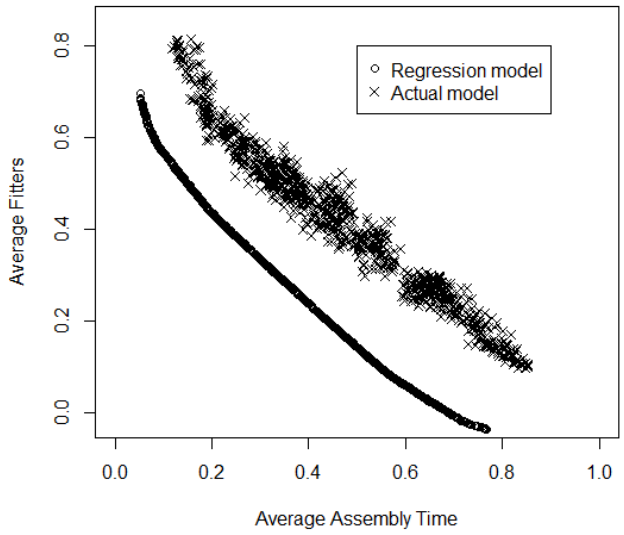


Figure 4-4 - A graph to show the meta-model Pareto front and the same points re-run using the actual simulation model

4.3.2 Experiment 2

The methodology outlined in Figure 4-1 has been carried out again with run length of 2,000,000 in order to see if the gap between the regression model and simulation model will be reduced. It is expected that the longer run length will enable the output results to show less variance, as the increased run length will enable the simulation model to reach a steady state.

Due to the significant amount of computational effort required to run the 5000 DOE points for 2,000,000 simulated minutes, it was decided to run this test without replications. Due to the increased run length, the DOE took around 17 days to complete. The regression models had the following statistical results.

The statistics for the linear regression model for the 'average fitters' KPI are as follows:

- Residual standard error: 0.01305 on 4927 degrees of freedom
- Multiple R-squared: 0.9316, Adjusted R-squared: 0.9306
- F-statistic: 931.9 on 72 and 4927 DF, p-value: < 2.2e-16

The low residual standard error and high R squared value suggest a good model fit. The statistics for the linear regression model for the 'average assembly time' are as follows:

- Residual standard error: 0.3008 on 4927 degrees of freedom
- Multiple R-squared: 0.7857, Adjusted R-squared: 0.7826
- F-statistic: 250.9 on 72 and 4927 DF, p-value: < 2.2e-16

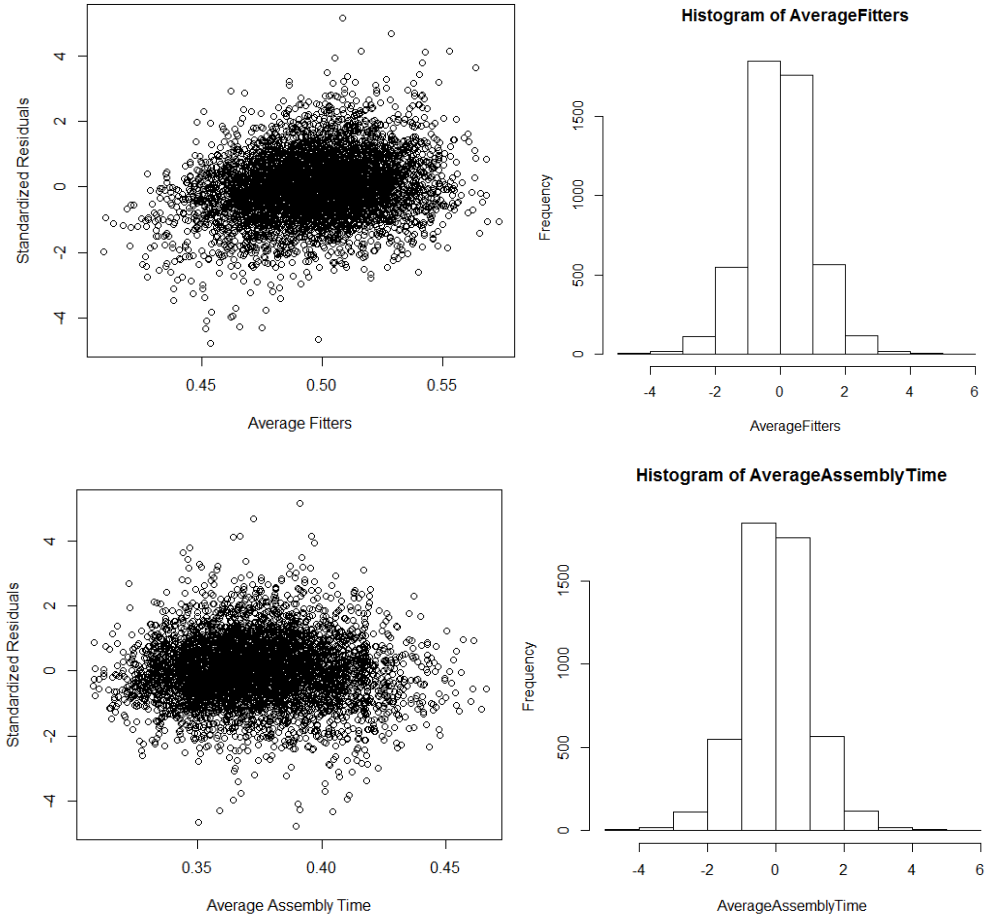


Figure 4-5 - Residual plots and histograms for the average assembly time and average fitters for experiment 2

The residual standard error for this model has dropped significantly compared to the linear regression model for the shorter run length; the value has dropped from 0.8821 to 0. 3008. However the R-squared value has dropped by 0.01.

The residual plots (shown in Figure 4-5) suggest the models are reasonable and hence they have been used to carry out the multi-objective optimisation.

The regression models were again used simultaneously to carry out the multi-objective optimisation. Upon completion of the optimisation the results were non-dominated sorted to give the Pareto front. The resulting 1124 points were re-run with the actual simulation model to compare the meta-models and actual simulation model. The results for these runs are compared with the results for experiment 1 in Figure 4-6.

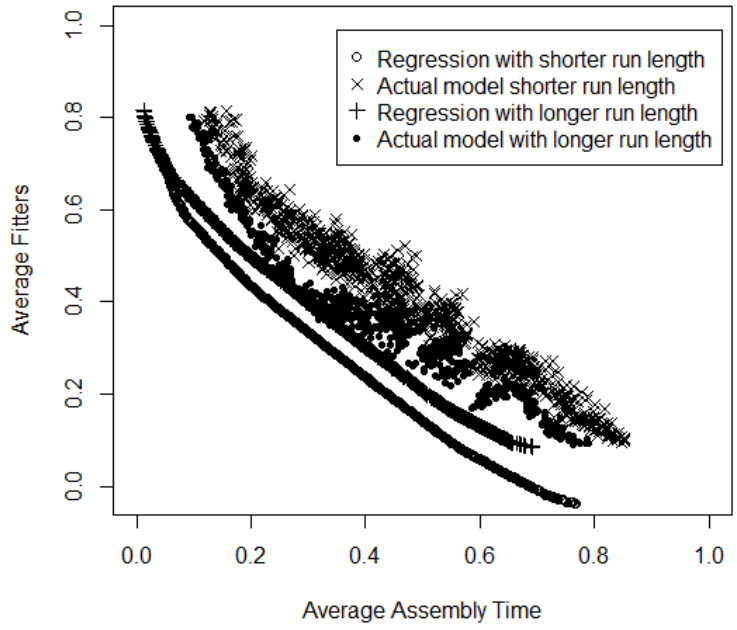


Figure 4-6 - A graph to show the comparison between the regression model Pareto fronts and actual results for experiment 1 and experiment 2

As expected the results for the longer run length regression optimisation and the actual model results are significantly closer than with the shorter run length. There are some points for the longer run length which overlap within the objective space as can be seen in the graph.

4.3.3 Experiment 3

A final experiment was carried out to assess the effect of the introduction of probability distributions for all of the cycle times on the results. To begin with another assessment was carried out on the effect of the run length on the average variance of the KPIs, Table 4-2 shows the results. Interestingly the average variance of the average assembly time for run length of 43,2000 is slightly lower when more probability distributions are added, however this could be coincidental and due to the stochastic nature of the sampling.

Table 4-2 - A table to show the average variance between replicates for varying run lengths for 3 output variables with probabilistic cycle times

	Average assembly time	Average NCRs	Average Fitters
Average variance for run length 432000	0.462422982	0.17172176	0.001998764
Average variance for run length 1,000,000	0.21652148	0.00838343	0.00127169
Average variance for run length 1,500,000	0.17668015	0.005134985	0.001961995
Average variance for run length 2,000,000	0.1178689002	0.00329375	0.001652597

A run length of 2,000,000 has been selected for this experiment based on the results in Table 4-2. Another 5000-point DOE was carried out with the new version of the model with probability distributions. Regression models were then created based on the DOE results. The regression models had the following statistical results.

The statistics for the linear regression model for the ‘average fitters’ are as follows:

- Residual standard error: 0.03892 on 4927 degrees of freedom
- Multiple R-squared: 0.9324, Adjusted R-squared: 0.9314

- F-statistic: 944.2 on 72 and 4927 DF, p-value: < 2.2e-16

The statistics for the linear regression model for the ‘average assembly time’ are as follows:

- Residual standard error: 0.8569 on 4927 degrees of freedom
- Multiple R-squared: 0.8023,   Adjusted R-squared: 0.7994
- F-statistic: 277.6 on 72 and 4927 DF, p-value: < 2.2e-16

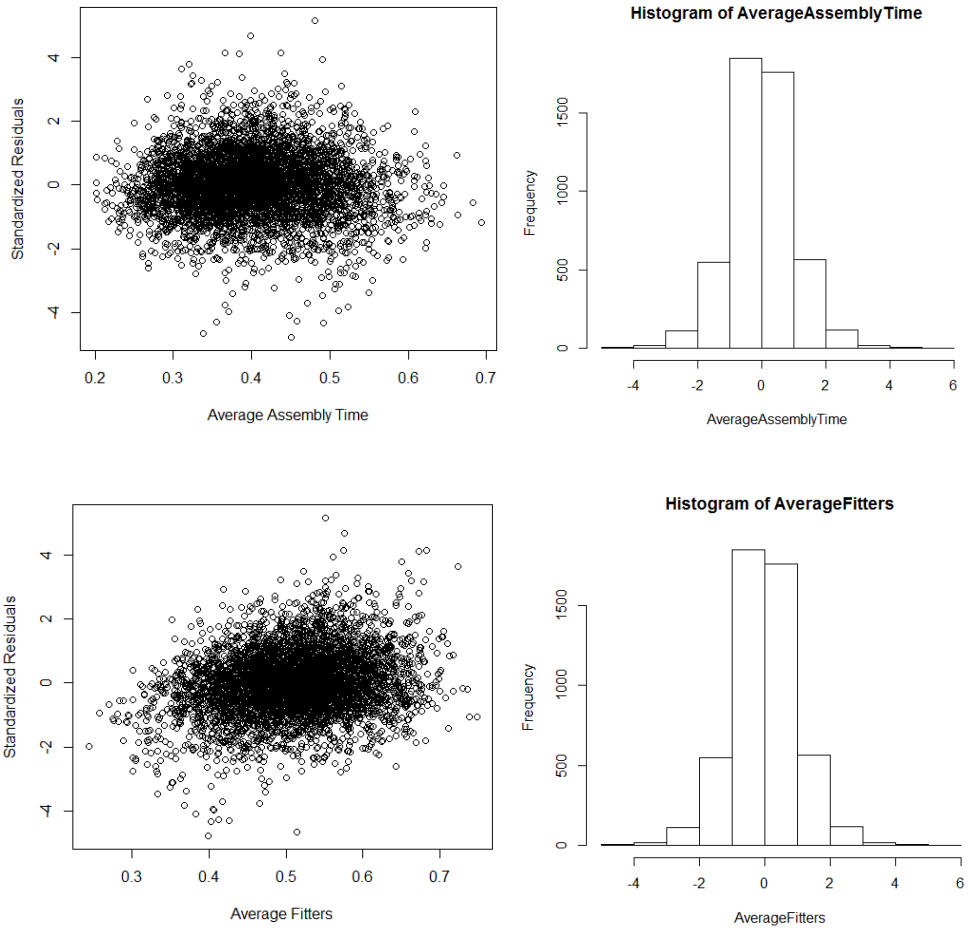


Figure 4-7 - Residual plots and histograms for the average assembly time and average fitters for experiment 3

The residual plots for these models are shown in Figure 4-7. The statistical results have been compared with the previous results with fixed cycle times. The statistical results for this test are comparable to the results for fixed cycle

times with a shorter run length. The statistical results for the meta-models with longer run length and fixed cycle times are the best.

These regression models were used simultaneously to carry out the multi-objective optimisation. Upon completion of the optimisation the results were non-dominated sorted to give the Pareto front. The resulting points were re-run with the actual simulation model to compare the meta-models and actual simulation model. The results for these runs are compared with the results for the simulation model with fixed cycle times and a longer run length; results are shown in in Figure 4-8.

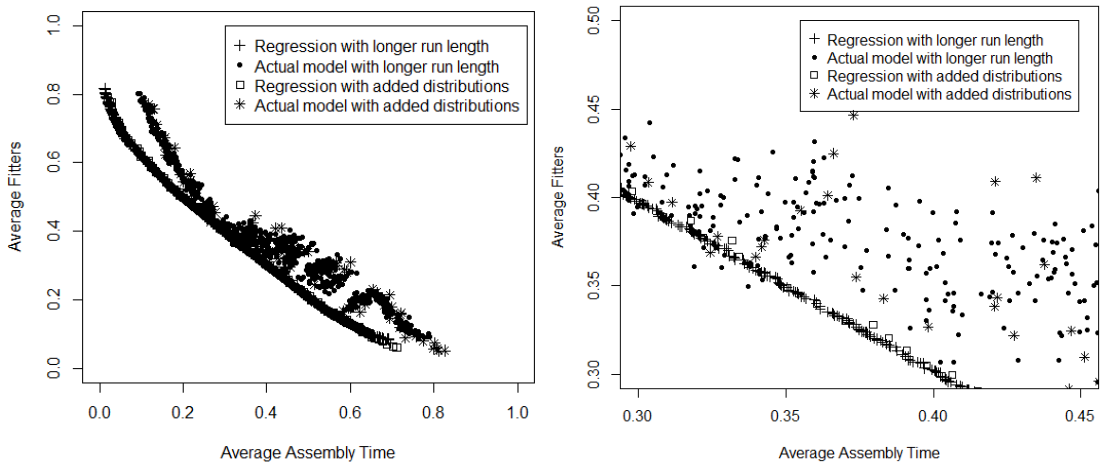


Figure 4-8 - A graph to show the comparison between the regression model Pareto fronts and actual results for experiment 2 and experiment 3

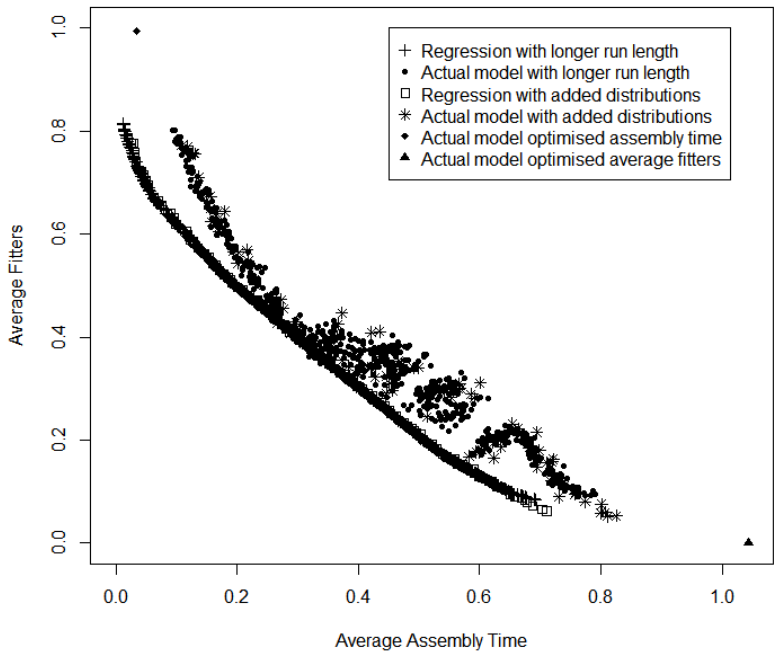


Figure 4-9 - A graph to show the comparison between the regression model Pareto fronts and actual results for experiment 2 and experiment 3 with the results of single objective optimisation for each objective function

The actual simulation model has been optimised for each objective function separately to create two points at the extremities of the objective space as shown in Figure 4-9. The results have been plotted on the same graph as the experiments in Figure 4-8. The results suggest that the area of the regression model and actual results is likely to be in the region of the true Pareto front. The results also confirm the multi objective nature of the two objective functions.

It can be seen from figures that the meta-models with and without probability distributions perform similarly and the results from the optimisation processes lie in the same region. Many of the points overlap within the objective space. It is noted that the results show points which are within the same location in the objective space, however these points do not necessarily reflect identical designs.

4.4 Results and Discussion

Within this section, three main experiments were carried out; the configurations for these experiments are shown in Table 4-1. Each of the experiments followed the optimisation methodology outlined in Figure 4-1.

Table 4-3 - A table to show the statistical results for regression models for each experiment and each objective

	Average Fitters			Average Assembly Time		
	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 1**	**Experiment 2**	**Experiment 3**
Residual standard error	0.04436	0.01305	0.03892	0.8821	0.3008	0.8569
Multiple R- squared	0.9138	0.9316	0.9324	0.7938	0.7857	0.8023
Adjusted R-squared	0.9125	0.9306	0.9314	0.7908	0.7826	0.7994
F statistic	725.5	931.9	944.2	263.5	250.9	277.6

Table 4-3 shows the statistical results of each experiment for each of the optimisation objectives. It can be seen that there is a high residual standard error for the 'average assembly time' objective for experiment 1 and 3; this suggests the models are a poor fit for these experiments. The meta-models for 'average fitters' have low residual standard error values for all experiments suggesting a good fit. This is expected as the average fitters is essentially a time weighted average of the input variables. The meta-models for the 'average fitters' objective account for a higher proportion of variability than for the 'average assembly time' objective.

For experiment 3, results show that there are a number of points which overlap (or come very close to each other) between the regression model and the actual model. To analyse the difference between the regression model and the actual model, two overlapping points were chosen. The points are shown in Table 4-4. For each point the corresponding decision variables have been analysed to understand if there is a difference in design.

Table 4-4 - A table to show the points where the regression model and actual simulation model overlap in objective space

	Regression model		Actual model	
	Average assembly time	**Average fitters**	**Average assembly time**	**Average fitters**
Point 1	0.308653292	0.389071	0.309777333	0.3873
Point 2	0.26560872	0.433681	0.265596	0.427266667

For point 1 it was found that, out of all 72 decision variables, only 8 decision variables were different between the regression model and the actual model. The 8 different decision variables for the regression model and actual model have been listed in Table 4-5. Values have been non-dimensionalised. It can be seen that the actual model has, on average, higher decision variable values compared to the regression model for a very similar solution within the objective function space.

Table 4-5 - Tables to show the difference between decision variable values for the regression model and the actual model for point 1 (above) and point 2 (below).

		Decision variables								Average value
	Regression model	0	0	0	0.5	0.5	0	0.5	0.5	**0.25**
Point 1	**Actual Model**	0.5	1	0.5	0	0	1	1	1	**0.625**

		Decision variables										Average value
	Regression model	0.5	0	0.5	0	1	0	0.5	0.5	0	0	**0.3**
Point 2	**Actual Model**	0	1	1	0.5	0.5	1	1	0	1.5	0.5	**0.7**

For point 2, 10 of the 72 decision variables were different between the regression model and actual model. Non-dimensionalised decision variables are shown in Table 4-5 for the regression model and actual model. Once again, the decision variable values are on average higher for the actual model compared

to the regression model. Results from both points suggest that the regression model is underestimating the decision variable values required for the given output KPIs.

In experiment 3, the average difference between the ‘average assembly time’ regression model output and the actual model result was calculated to be -0.11127 for identical decision variables. Every value was calculated as negative. This suggests the regression model consistently underestimated the average assembly time by more than 10%. The average difference between the ‘average fitters’ regression model output and the actual model results was calculated to be 0.028294. This value suggests that the regression model is providing a reasonably accurate estimate of the ‘average fitters’ output.

The average Euclidean distance between the identical decision variable points in the objective space was calculated to be 0.124133. This suggests that the points are around 12% of the search space away from each other. This is primarily due to the poor estimation of the ‘average assembly time’ output.

The graph in Figure 4-14 shows the points from the regression model and actual model in objective space. Two sets of points have been highlighted. Points 1 and 2 have been highlighted in red; each pair of points has identical decision variables. It can be seen these pairs of points are still considerably far away from each other within the objective space.

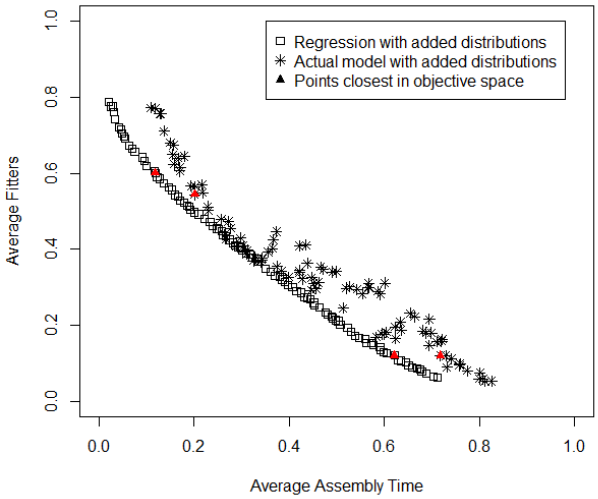


Figure 4-10 - A graph to highlight the pairs of points which have identical decision variables and are closest to each other with respect to each objective function

It is clear that the 'average assembly time' meta-model is not capable of predicting the simulation result to a reasonable level of accuracy. In order to improve the prediction capabilities, the simulation run parameters could be changed in future experimentation or a different meta-modelling strategy can be applied. In particular, by changing the run window to exclude the warm-up period the prediction could be improved. In addition, the simulation run length can be extended, however the computation time required for this is too high.

This optimisation methodology could be used for future multi-objective optimisation of DES models; however, it is expected that with more complex DES models the meta-models may struggle to accurately represent the simulation model behaviour. In this case higher order meta-modelling methods should be explored as an alternative. Other meta-modelling methods such as Artificial Neural Networks could also be investigated in the future as an alternative.

The presented optimisation methodology can be used in place of multi-objective optimisation on the DES model, in order to reduce computation time required for optimisation. It is proposed that this method will enable users to drive towards

the Pareto front region of the objective space with significantly less computation time than explicit multi-objective optimisation on the DES model. Users can subsequently choose single solutions and explore the nearby regions using single-objective optimisation methods.

It has been noted that interpreting the results may be difficult for manufacturing and assembly experts. The Pareto front clearly demonstrates the trade-off between the two output objectives; however, it does not assist with choosing the final optimal results and corresponding decision variable values. Chapter 6 provides a framework to assist decision makers in understanding the relationship between input and output variables for multiple objectives.

4.5 Chapter Summary

This chapter has presented a novel multi-objective optimisation methodology using meta-modelling for a turbine assembly test case. The methodology includes Design of Experiments generation, meta-model creation, multi-objective optimisation of the meta-models and comparison of the meta-model results against actual simulation results. The methodology is able to reduce the computational cost of carrying out multi-objective optimisation directly on the simulation model. Results indicate the meta-model of the resource allocation is able to represent the simulation model with reasonable accuracy. However, the meta-model of the average lead time is not of sufficient accuracy to replace the simulation model.

Further work could be carried out to assess the effect of simulation run window on the results and the effect of extended run lengths on the results; however this will be very computationally expensive to carry out. The meta-modelling method can also be extended to include higher order polynomial models to assess the impact on results. Other meta-modelling methods could also be explored to improve the prediction accuracy.

The produced results are not easy to interpret for decision makers, hence further work needs to be carried out to identify methods to improve decision making and engage stakeholders.

CHAPTER 5 -INTERACTIVE IMMERSIVE VISUALISATION OF DES MODELS

Recent years have seen the introduction of Virtual Reality (VR), a technology which enables users to experience a highly detailed immersive environment. Significant benefits can be achieved through enabling interactive immersive visualisation of factory layouts. In particular, the use of VR with DES can enable increased stakeholder engagement and improve decision making capabilities of DES.

This chapter presents the development of a framework for interaction between immersive visualisation in VR and DES models of factory layouts. This work furthers existing research by creating immersive DES models with a closed feedback loop to enable users to test various scenarios and ascertain optimal configurations. Hence this framework assists users by improving decision making capabilities of DES.

A maintenance test case has been selected for the development of the interface. Meetings with production managers of the facility are used to create the requirements for visualisation and interaction with the factory layout. The requirements are used to create a user interface in Matlab with a feedback loop to enable user interaction. The feedback loop is validated using two examples: layout reconfiguration and labour utilisation. The demonstrated framework can be used to enhance the decision making capabilities of DES models.

This chapter presents the following:

- the requirements for an immersive interaction interface for a maintenance test case

- a proposed integration framework using a User Interface (UI) in Matlab
- validation of the framework on two different case studies
- results of the model validation and recommendation for future work

This chapter in particular contributes to work in the area of Industry 4.0 and IoT as the use of virtual prototyping is key to enabling right-first-time reconfiguration of factory layouts. Industry 4.0 aims to integrate various concepts including mass product customisation, short batch sizes and flexible production facilities (Posada et al., 2015). In order to enable these concepts, the 'smart factory' must be highly reconfigurable to deal with regular changes in demands from the market. In a UK government report it is noted that "*The highly desirable future 'Reconfigurable Factory' requires major advances in, and integration of, many of the technologies we are currently familiar with. Rapid reconfiguration and fast ramp implies increased use of simulation and modelling, testing factory layouts and processes in a virtual reality environment, virtual factory design and virtual factories*" (N. N, 2013). This Chapter presents virtual prototyping of factory layouts with simulation and the introduction of a framework to enable decision makers to test various scenarios in a VR environment. This closed feedback loop enables decision makers to ensure that the factory layouts generated are right-first-time; this ensures that there is no unnecessary downtime within the factory and ultimately increases factory and machine utilisation.

5.1 Background

DES is an established tool for modelling complex real world processes with multiple dynamic interactions. DES is well suited for use comparing different layout scenarios of production lines as it allows for easy visualisation of computer animations, metrics and graphs (Sweetser, 1999). However, Jahangirian et al. (2010) show that DES does not possess the same level of stakeholder engagement as simulation gaming.

There are a wide range of potential benefits which can be achieved through linking DES and immersive visualisation using VR. Simulation and visualisation can assist decision makers in understanding underlying manufacturing systems

(Dangelmaier et al. 2005). Visualisation of manufacturing systems can be used to validate system designs before they are constructed (Rohrer 2000). Korves and Loftus (2000) have demonstrated the benefits of immersive VR visualisation for layout planning applications, such as to detect flaws in the system model.

In order to provide the most value to simulation practitioners the immersive environment should be highly interactive (Rohrer 2000). Interactivity between the visualisation and the simulation model can be used to correct problems within the model (Dangelmaier et al.2005), this leads to a closed loop system where users can make modifications from within in the virtual environment and then visualise the impact of those changes. This could be particularly valuable for decision makers as it will enable them to make informed decisions and can increase the level of stakeholder engagement.

The current development process for websites, for instance, uses rapid prototyping techniques; the design can then be iteratively refined before valuable time is spent in development (Alger, 2015). This method can be extended to the design of interfaces for virtual reality applications. Alger (2015) notes that aspects of UI are only small parts of the user experience. Hence the functionality prototyping method can be used to design VR interfaces. This can be followed by testing of the user experience and subsequent revision and modification.

There have been a number of studies which have carried out research in this area. Korves and Loftus (2000) evaluate different layout plans without functionality for interactive changing of the layout. They primarily compare a monitor based system and a head mounted display (HMD). Dangelmaier et al. (2005) propose a theoretical framework to integrate visualisation and real time data link; they also discuss how simulation and visualisation interaction can be used to correct problems. Lau et al. (2009) create an information visualisation platform to visualise KPIs and create visualisation viewpoints. They focus on the development of a VR visualisation system for the analysis of a logistics system performance, however they do not close the loop and consider feedback and

model adjustment. Yang et al. (2011) create a system to visualise and analyse information obtained from product lines or the whole manufacturing system. They use a simulation of sound propagation in a virtual factory as a case study. Effort has also been made to assess usability requirements, Stanney et al. (2003) create a multi-criteria virtual environment usability matrix and Bowman et al. (2002) survey usability evaluation approaches for virtual environments.

Through analysis of the literature a research gap can be identified and addressed with the creation of a DES model with immersive visualisation in VR and a closed feedback loop to enable full interactivity from within the immersive environment. Korves and Loftus (2000) consider VR for a similar layout planning application however they do not enable interactive layout modification and a closed feedback loop. This section presents an interactive framework for immersive visualisation which specifically addresses this research gap and the developed system is validated using two closed loop examples for a repair facility test case.

5.2 Requirements from an immersive interface

An interactive user interface framework is to be developed specifically for the repair facility test case described in Chapter 3. In particular, the test case has been created with the aim to analyse the feasibility of various layouts. The test case is a repair facility which carries out the periodic overhaul of a component. The repair facility layout requires rearranging for different maintenance regimes; hence DES can be used to prototype the different facility layouts.

In Chapter 3, a 2D DES model has been created to enable analysis of KPIs. However, in further work the 2D model has been augmented with 3D models of the maintenance facility components to enable immersive visualisation (Petti et al., 2016). The 3D model will be used as the basis for the developed immersive interaction framework.

In order to understand the requirements for the immersive interaction interface, various meetings were held with two production managers of the maintenance

facility. Six meetings were held with the production managers to identify the key features of the test case and to identify the main feature requirements for the immersive interface. During meetings, the production managers were asked a number of questions including:

- What aspects of the system can be modified?
- What scenarios should be tested using the model?
- What are the KPIs of interest?
- What are the main process bottlenecks?
- What information should be visualised?
- What are the main process drivers?

The meeting minutes were documented and used to create a list of the requirements of the interactive immersive interface. The requirements were split into three main areas: visualisation, interaction and specific tasks. Table 5-1 lists the requirements of the developed framework split into the three main areas. The first column lists the required information to be visualised in the user interface, this information is to be sent from the simulation model to the user interface. The second column lists the interaction requirements; these tasks are commands which need to be sent from the user interface to the simulation model. The third column shows more complex tasks which are of interest for production managers, these tasks require a closed feedback loop for interacting with the model.

Table 5-1 – A table to show the requirements from an immersive interface framework

Visualisation	Interaction	Tasks
• Overall production rate • Current WIP • Throughput/lead time • Distance travelled by parts • Overall clock of model • Current shift • Machine names • Part types	• Run, stop, and step model • Fast forward to chosen time • Read machine name • Read labour utilisation • Read machine utilisation • Read buffer utilisation • Break down a machine • Restart a machine	• Change factory layout and view impact on distance travelled • Change the production sequence of part types and visualise impact on throughput • View labour utilisation and to change number of workers • Read part type and reroute part • Read time a part has spent in system • Visualise buffer utilisation and change capacity • Move the location of a buffer

5.3 Framework for interactive immersive visualisation of DES models

A framework has been proposed to enable the functionality prototyping for immersive interaction with a DES model. The framework includes a prototype interface written in Matlab; this will be used to enable testing of the functionality and user experience prior to direct implementation within the immersive environment. The Matlab interface will have two-way interaction capabilities with the DES model. In addition, the DES model will have capability to interact with and create the immersive visualisation environment. The proposed framework is shown in Figure 5-1.

The proposed framework will be used to enable users to analyse various scenarios and the corresponding KPIs from within the immersive environment.

This framework will then be used to test the user experience and assess the functionality.

The WITNESS API and Visionary Render aspects of the framework have been previously developed to enable immersive visualisation of DES models as part of the 'Towards Zero Prototyping of Factory Layout and Operations Using Novel Gaming Technologies' project. The project is outlined in section 1.5.

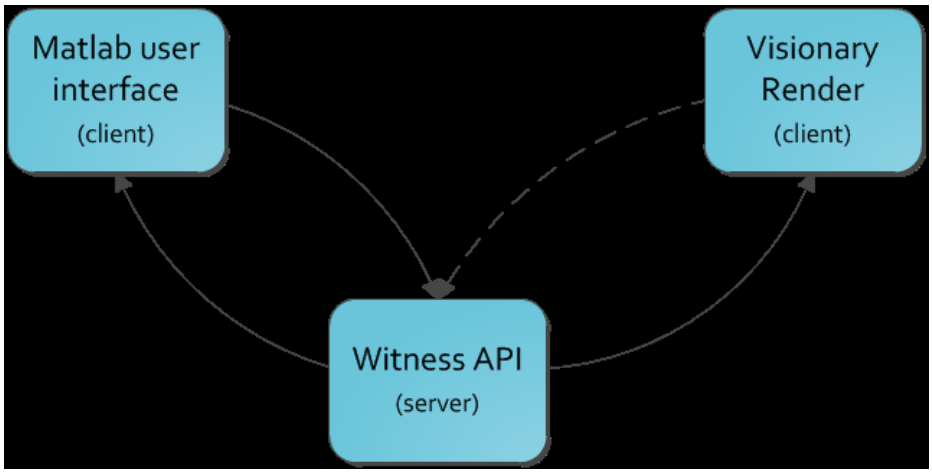


Figure 5-1 - Proposed framework with Matlab GUI for virtual reality visualisation of DES models

This framework involves the integration of the DES model (WITNESS API) and the immersive visualisation (using Visionary Render software). Socket communications were developed within the original framework to allow the WITNESS API to send commands to the Visionary Render visualisation. The prototype user interface has been created to enable two-way interaction between the immersive environment and the underlying simulation model. The user interface has been developed in Matlab as it enables quick functionality prototyping. The user interface can be used in collaboration with the immersive visualisation to interact with the immersive DES model.

The proposed integrated framework is shown in Figure 6-3; the framework enables two-way communication between the simulation model and the user interface. TCP/IP socket communications from the original framework have

been used to create a client/server communication between the WITNESS API simulation model and the Matlab user interface. In addition, a file-based communication process has been created to enable visualisation of simulation model statistics within the Matlab user interface. A file-based process has been selected as this leverages the robust file reading and writing capabilities within WITNESS.

The user interface has been designed to include all of the functionality requirements within three main sections: simulation run data, machine data and functions. The ‘simulation run data’ section shows overall simulation KPIs to enable users to understand model behaviour. A refresh button has been created to enable users to update the data within this section. The ‘machine data’ section allows users to select specific machines and visualise the corresponding utilisation statistics. In addition, users are able to visualise statistics for labour pool utilisation. The ‘functions’ section contains all of the commands which are to be sent to the simulation model. A number of the functions mimic the run toolbar within WITNESS. This enables users to easily navigate through different simulation runs from within the immersive environment. The user can choose to run, stop, step, reset or batch run to a chosen time. Upon pressing one of these commands within the user interface, an XML string is generated and sent to the WITNESS API. The other functions provide capability to carry out a number of the tasks as defined within Table 5-1.

In particular, two main tasks were implemented: changing the factory layout and changing the number of labourers. For each task a function was created within the WITNESS model and a corresponding button to carry out the function was provided in the immersive environment. Upon clicking the task button, an XML string is generated which includes the function name. The XML string is then sent to the model to be executed. Users can visualise the impact of their tasks by running the model and visualising the new KPIs in the user interface. This closes the feedback loop.

5.4 Test cases- layout reconfiguration and labour adjustment

Two different experiments have been carried out to validate the proposed framework. The experiments have been carried out to demonstrate the feedback loop. Two main tasks were chosen for the two experiments to enable testing of the two-way interaction: layout reconfiguration and labour adjustment. The following subsections detail the two experiments.

5.4.1 Layout reconfiguration experiment

Factory layout reconfiguration was chosen as the first experiment to demonstrate the feedback loop. This task is particularly important for production managers as the model has been created specifically to assist with layout planning for future operations. The purpose of this experiment is to assess the feasibility of layout planning and reconfiguration within an immersive environment with an interactive user interface and to identify if there are any shortcoming of such an approach.

Four different layouts have been preconfigured within the WITNESS model to enable factory layout reconfiguration within the immersive environment. These four layouts have been created based on knowledge of requirements and spatial constraints within the facility. The first layout is the current configuration within the facility. The three other layouts have been designed as potential alternative future options based on a new sequence of maintenance tasks. The alternative future layouts have been designed by changing and swapping locations of existing workstations. The WITNESS 2D layout has been used to carry this out, as the 2D layout is automatically linked to the 3D layout.

A comparison between layout number 2 and the original design (layout 1) has been chosen for the experiment. Layout 2 has been designed to test the swapping of two main work stations within the factory. Figure 5-2 shows the process steps required to carry out the factory layout reconfiguration using the developed user interface. The steps involve running the original configuration and assessing the results, then changing the layout and re-running the simulation to produce new results.

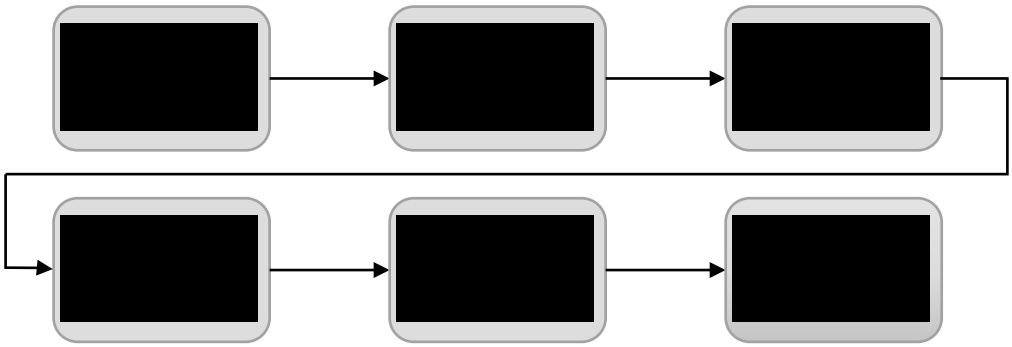
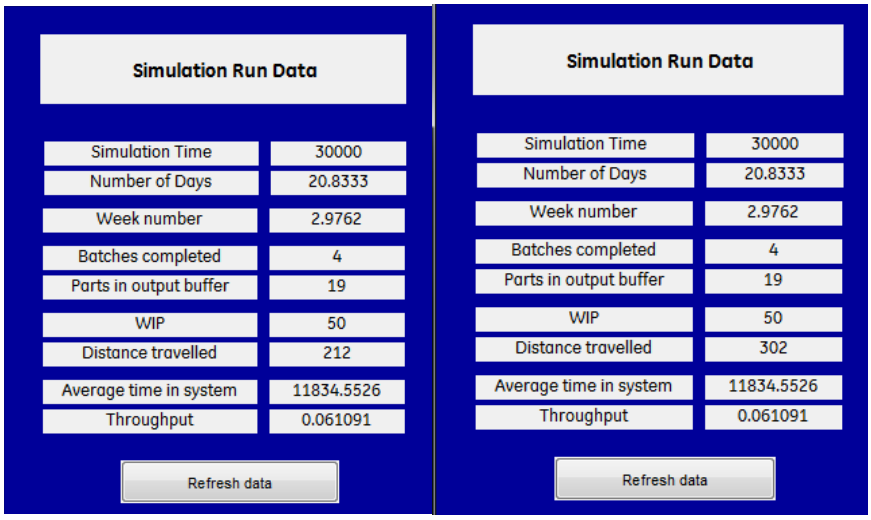


Figure 5-2 - Process steps required to carry out the factory reconfiguration

Figure 5-3 shows the simulation run data within the user interface after 30,000 simulated minutes for the original layout (left) and the new proposed layout (right). It can be seen that the new proposed layout has a significantly higher distance travelled for parts. Figure 5-4 shows layout 1 (a and b) above compared to layout 2 (c and d) below. It can be seen that the workstations have sufficient space in the new layout.



(a) (b)

Figure 5-3 - Simulation run data for the original layout (left) and for the new proposed layout (right) showing the new distance travelled

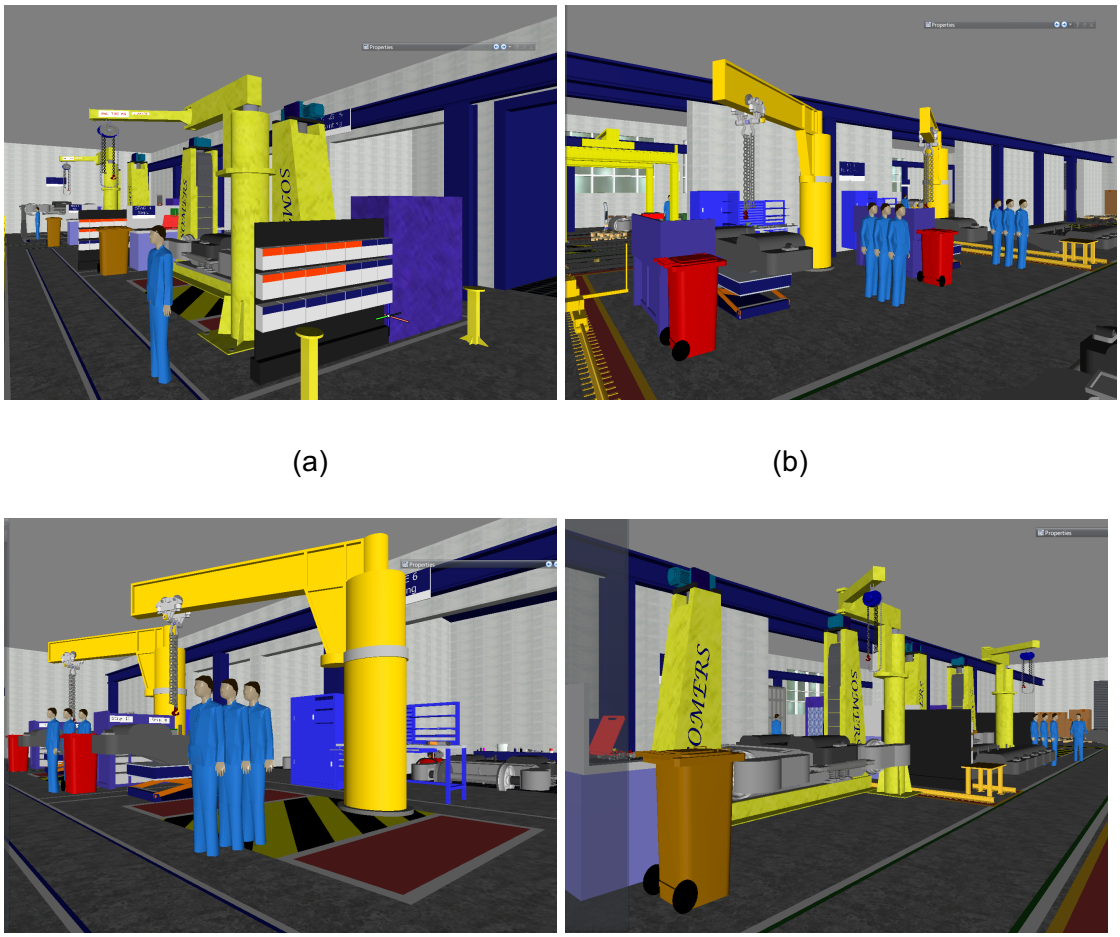
5.4.2 Labour utilisation experiment

Another experiment has been identified to demonstrate the feedback loop. Production managers are interested in maximising the utilisation of the labour pool. This example demonstrates the ability to visualise worker utilisation within the immersive environment and to subsequently modify the number of workers and visualise the impact on the utilisation. Figure 5-5 shows the process steps.

The model is run for 30,000 simulated minutes with the original setup. The UI demonstrates the utilisation of the labour pool with this setup in Figure 5-6a. Within the example the user can reduce the number of workers and rerun model for the 30,000 simulated minutes. The number of workers was set to be reduced to 18 as this value was chosen by production managers. The results are shown in Figure 5-6b. Figure 5-6 shows that the reduction in workers increases the labour utilisation and decreases the idle time. Results are shown in the 'Machine data' section under 'labour busy' and 'labour idle'.

5.4.3 Results

A number of observations have been made as a result of the two experiments. For the factory layout reconfiguration experiments, a number of inconsistencies were found to occur when the factory layout was reconfigured. Upon layout configuration it was found that the workstations fitted well within the new layout, however upon observation of the 3D model by production managers the locations of the buffers were found to cause an issue although there was no difference in KPIs. In particular, two buffers for parts were found to be in the wrong location after workstation configuration. This observation demonstrates the importance of using 3D immersive layout planning to identify errors. Figure 5-7 shows the buffer for parts lying within a red flooring area indicating potential collision with other parts. Figure 5-8 shows another part buffer which is too close to the workstation and overhead equipment.



(a) (b)

(c) (d)

Figure 5-4 - Screenshots of the immersive environment from two locations. Location 1 (a and c) and location 2 (c and d). Screenshots a and b show the original layout; screenshots c and d show the layout after reconfiguration.

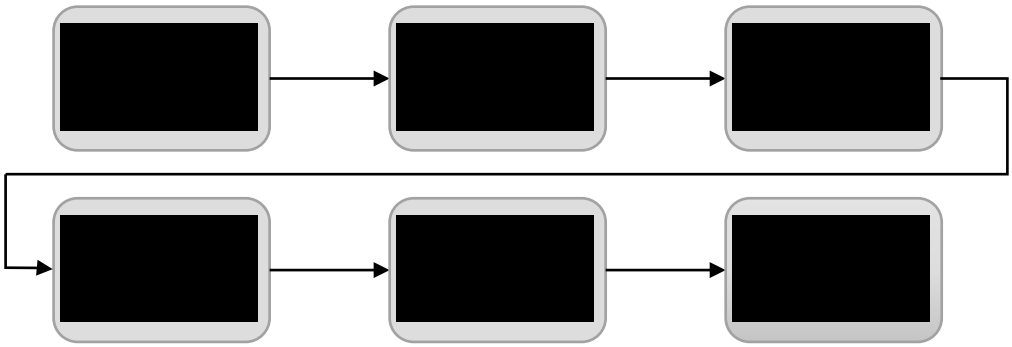
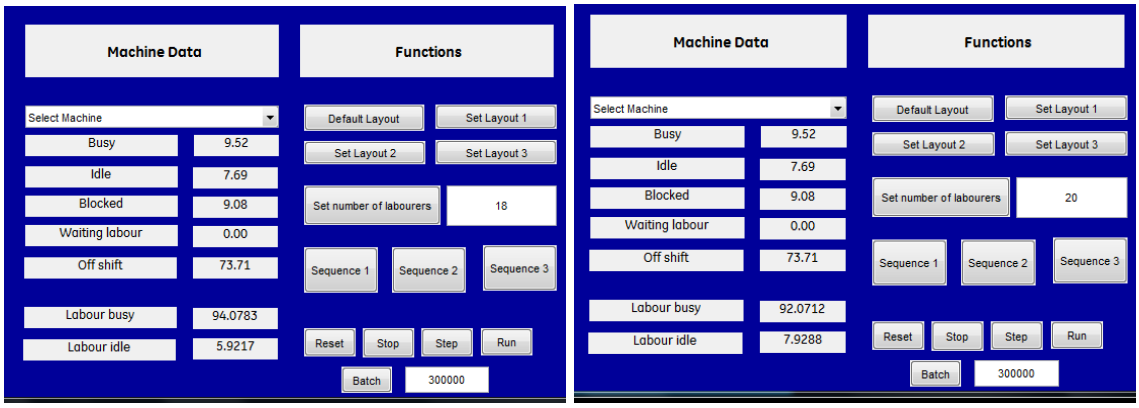


Figure 5-5 - Process steps required to carry out the labour changes



(a) (b)

Figure 5-6 - Screenshots of the Matlab UI with the original 20 workers (a) and after a reduction in the number of workers (b) with the labour utilisation for each scenario shown at the bottom of the 'machine data' section



Figure 5-7 - A screenshot to show the buffer for a part in the incorrect location (part lies on the red floor area posing a potential collision risk)

These figures suggest the new layout needs further revision and modification to ensure that it is a feasible solution. In addition, the paths which are used for parts have been fixed for the 3D immersive layout, hence upon reconfiguration of the layout the original paths are still used by parts for movement. This leads to parts moving in the opposite direction to the required destination and upon reaching the end of the path the parts will jump to the correct workstation. In addition, the results for many of the KPIs were found to be identical before and

after the layout reconfiguration. For example, the average time in system and throughput were identical for both configurations.



Figure 5-8 - A screenshot to show the workstation buffer too close to the workstation

For the second case study it was found that the labour utilisation adjustment did not have a clear impact on the 3D immersive model. This case study is particularly important as it demonstrates the importance of the user interface to show users results which aren't immediately visible in the immersive environment. Although the results have been clearly reflected in the user interface; there is no method to visualise which workstations have been impacted by the change.

5.4.4 Discussion of Results

This chapter presents the development of a Matlab based UI to create a feedback loop between immersive visualisation and a DES model for a repair facility test case. The developed framework has been validated using two experiments; the experiments demonstrate the two-way interaction between the interface and the simulation model. The framework presented is novel as it extends existing literature by creating an immersive DES model with a closed feedback loop and a user interface to enable interactivity with the model and configurability.

The requirements for the interface have been identified through meetings held with two production managers. Requirements have been split into three main sections: visualisation, interaction and tasks. The requirements are specifically for the repair facility test case, however many of concepts within the specification will be universal requirements for other DES models. For instance, machine, labour and buffer utilisation will be KPIs for many DES models. In addition, the run toolbar created will be universally applicable for immersive visualisation of DES models. In particular, the production managers are interested in reconfiguring the layout for future repair requirements. It is vital to ensure that only key information is displayed at any given time within the immersive environment. It is envisaged that, in the future, the interface sections may be options for visualisation allowing users to choose appropriate combinations of information.

Two versions of the DES model were originally created, one version for 2D analysis and another version for 3D visualisation. The 3D model has been simplified to exclude some material handling processes and part types; this enabled faster processing and smoother visualisation. In particular, it was found that a number of specific functions had to be added to the 3D DES model to enable features within the developed user interface. These features include functions to change the layout of the facility This additional functionality conflicts with the requirement to streamline the model for immersive visualisation, the resulting model was found to be slower to run.

Within the 3D DES model, paths were explicitly placed between workstations to enable accurate visualisation of movement within the immersive environment. However, this directly conflicts with the requirement to quickly redesign the layout of the facility. Upon setting a new layout within the 3D model the paths needed to be manually reconfigured to show the new route. The 2D DES model has been created using the ‘pseudo paths’ construct in WITNESS; parts and materials move ’as the crow flies’. It is envisaged that in the future the 3D model will initially be created with ‘pseudo paths’ to enable quick redesign and

analysis of spatial requirements. Users can then select specific layouts and create fixed paths accordingly for accurate visualisation.

In addition, the experiments have demonstrated the requirement for further functionality. In particular, the reconfiguration experiment demonstrated that buffer locations clashed with requirements within the new layout. In order to resolve this, it is proposed that in future work the workstation should be considered as the union of the workstation and the associated input and output buffers. The labour utilisation experiment demonstrated that there is no visual effect of the reduction in labour on the 3D immersive visualisation. It is proposed that further functionality could be added within the immersive environment to reflect the workstation status in the future. For instance, 3D coloured lighting could be used to indicate the workstation state, similarly to how it is indicated in the 2D WITNESS model. This could provide a good familiar visualisation method for DES users.

5.5 Chapter Summary

This chapter has presented the development of an interface to enable two-way interaction with a DES model from within an immersive visualisation environment. Requirements for the immersive visualisation and interaction have been collected from production managers for a repair facility test case. The feedback loop has been validated using two different experiments: layout reconfiguration and labour utilisation. Drawbacks of the two case studies have been identified as: difficulties relating to buffers when moving workstations and difficulties understanding workstation states. It is proposed that buffers and workstations could be combined for layout reconfiguration in future work. A coloured lighting scheme can also be introduced within the 3D environment to visualise workstation states, similar to the status lights used in 2D DES visualisation.

It is envisaged that this integration framework will enable a new level of interactive decision-making for manufacturing environments downstream within the manufacturing process and thereby improve decision making.

CHAPTER 6 -VISUALISATION TO SUPPORT DECISION MAKING WITH DES MODELS

DES has been widely used to assist with decision making. In particular, scenario analysis is often used to test various options and assess the best system configurations. Within literature, data mining techniques have been widely applied to gain insights into data sets across different industries, however, little work has been carried out to visualise the relationships between the input and output variables for DES models using data mining to assist with decision making for stakeholders. This section presents a framework to enable DES users to visualise the effect of input variables on output objectives using regression trees. This section proposes the extension of the use of regression trees for multiple objectives by overlaying the decision tree with expected values for multiple objectives. In addition, the use of random forests is applied to improve the prediction capabilities of regression trees.

The following chapter aims to achieve the following:

- Introduce the use of regression trees to assist with decision making
- Provide a framework to create regression trees and random forests using DES models
- Apply and validate the framework to a number of case studies
- Present the results of the case studies and discuss the benefits and drawbacks of the framework

6.1 Introduction

One of the key benefits of simulation modelling is the ability to provide multiple scenarios to aid in the decision making process (Lu & Sundaram, 2002). Through the assessment of various scenarios, optimal configurations and decisions can be identified and applied. This process ensures that companies

remain competitive with respect to their production costs and times. Within literature little work has been carried out to visualise and interpret the results of DES scenarios and to compare them. Visualisation of the trade-off between output objectives has already been achieved in Chapter 4; however, for DES results it is also valuable to understand the impact of each input variable on the output objectives. In particular, a method to visualise the relationships between input and output variables can be used to inform stakeholders and decision makers of vital insights relating to their processes. This can be used to ensure that decision makers can understand their system performance under varying input parameters and to ensure they can take optimal decisions in various situations.

Data mining techniques can be used to assist with this visualisation and to improve the decision making capabilities of DES. In particular data mining techniques enable users to discover patterns and information about large sets of data, these methods can be used to gain insights into DES model behaviour. In an extended abstract, Watson, Cross, & Albiston, (2010) briefly present the ongoing development of a methodology combining experimental design, regression, variable importance charts and Classification and Regression Trees (C&RTs). They note that the data mining and visualisation techniques can enable new insights to be gained and a simpler perspective for interpretation. However, they do not detail the methodology or present an example of the data mining visualisation.

This chapter presents a detailed framework to enable visualisation of DES results using regression trees. In particular regression trees have been chosen to assist with the visualisation as regression models were used for the optimisation methodology in Chapter 4.

The subsequent sections detail the developed framework and the application of the framework to 3 case studies with increasing complexity. Results, discussion and limitations are presented thereafter.

6.2 Framework for visualisation to support decision making with DES models

In this section a framework is defined to enable the visualisation and interpretation of DES results. The framework and regression tree growing algorithm are explained in detail, subsequent sections detail the application to 3 different case studies and the results and conclusions.

In order to create a regression tree, a Design of Experiments must first be generated with relevant inputs being varied on a number of different levels. A regression tree growing algorithm can be used to create a regression tree on the resulting data set. Each regression tree is created with respect to a given objective function. Upon completion of the algorithm the regression tree can be used to show the most important variables for the output objective. The top of the tree will show the expected value of the output variable for all observations, and a decision (binary split) for the variable with the biggest impact on the output variable. Each subsequent node will also present this information.

The generated regression trees can then be used to assist decision makers and stakeholders who are unfamiliar with DES to understand the impact of input variable values on the output objective. For example, users can identify the current state leaf and move to other leaves of the preceding node to understand how the expected value of their output objective changes.

The proposed process is presented in Figure 6-1 which involves designing the experiment, creating the DES results and then growing regression trees. The resulting regression trees can be used to visualise the impact of input variables on the output variables and the expected value of the output variable. The method for growing a regression tree is given as follows.

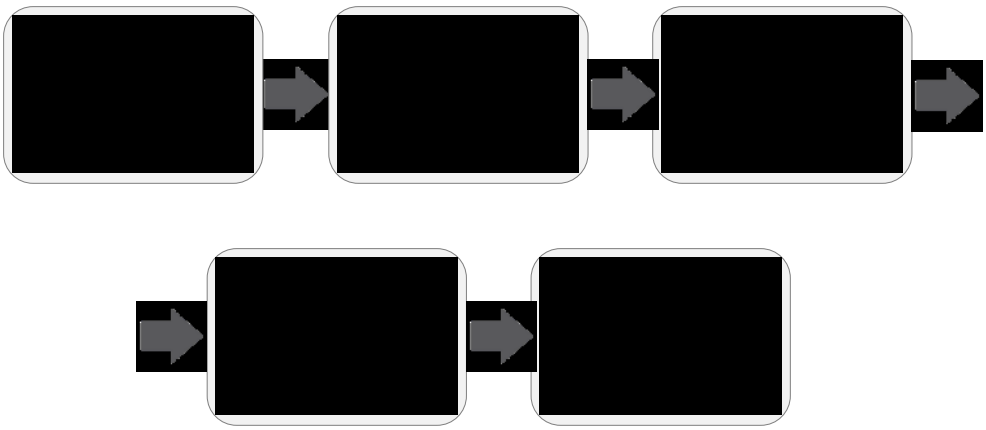


Figure 6-1 – A diagram to show the process for creating regression trees

This section details the regression tree growing algorithm applied within this work. Table 6-1 outlines the nomenclature used. Figure 6-2 shows the flowchart of the regression tree growing algorithm. The algorithm is as follows:

- Step 1- Start at a node – calculate the expected value of the objective (given in equation 6.1) given all of the output results and the sum square errors S (as defined in equations 6.2 and 6.3)
- Step 2- If all the points in the node have the same value for all the input variables, stop.
    - Otherwise, search over all binary splits of all input variables for the one which will reduce S as much as possible. If the largest decrease in S would be less than some threshold δ, or one of the resulting nodes would contain less than q points, stop.
- Step 3- Otherwise; take that split, creating two new nodes. Repeat from step 1

$$E[y] = \frac{1}{n_c} \sum_{i \in T} y_i \qquad \textbf{(6-1)}$$

$$S = \sum_{C \in T} \sum_{i \in C} (y_i - m_c)^2 \quad \textbf{(6-1)}$$

(6-2)

$$m_c = \frac{1}{n_c} \sum_{i \in C} y_i$$

Table 6-1 – A table to show the nomenclature for the decision tree growing algorithm

C	A leaf in the decision tree
m_c	The expected value for leaf c
i	Each point within the data set
n_c	The number of points in leaf c
S or SSE	Sum of squared errors
y	Objective function
T	The regression tree
q	Predefined limit for min no of points in a leaf C
δ	Threshold for reduction in S – termination criterion

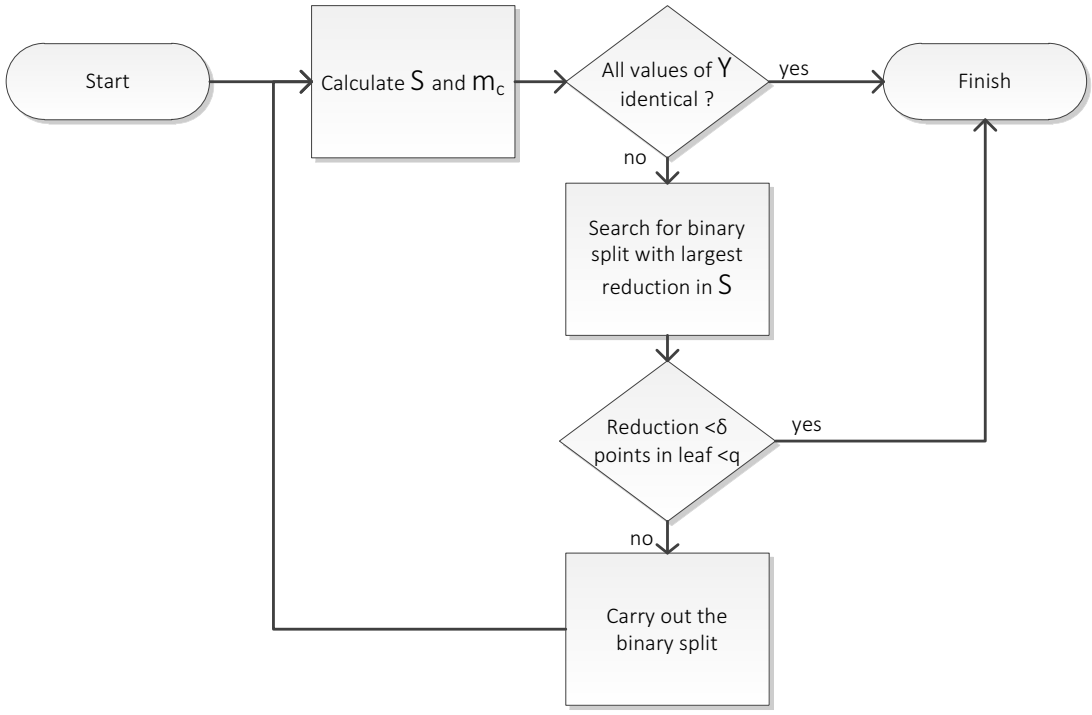


Figure 6-2 A flowchart to show the regression tree growing algorithm

6.3 Test Case- A Small Manufacturing System

A test problem has been generated as a case study in order to demonstrate the proposed framework. Case study 1 is a small manufacturing system which has been created with input variables used to represent the key features of a manufacturing system including: manpower, transportation resources, variable arrival times and buffers.

6.3.1 Description of the manufacturing process

The case study is a small production system which produces two parts, Part 1 and Part 2. A DES model of the manufacturing system has been created in WITNESS, a screenshot of the model is shown in Figure 6-3.

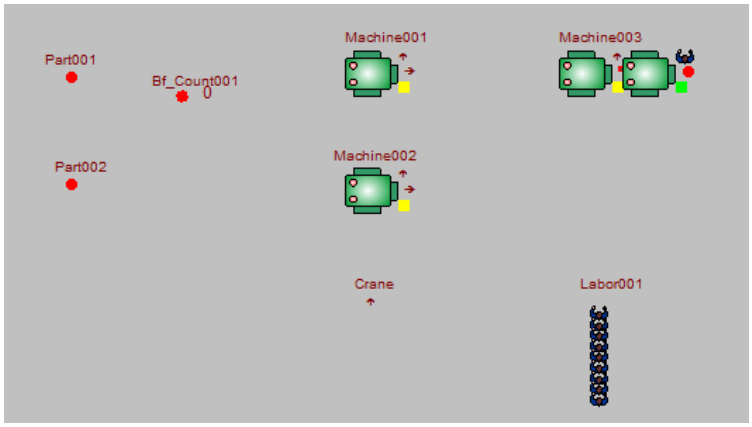


Figure 6-3 - A screenshot of the case study DES model

The parts enter a buffer when they arrive in the facility, the buffer capacity is 1000. All parts require a crane to be transported between the machines and the input buffer.

Machine 1 and Machine 2 represent manual processes; Machine 1 and Machine 2 process Parts 1 and 2 respectively. The number of workers at Machine 1 and Machine 2 can vary, hence the cycle times also vary. The cycle time for Machine 1 is 100 minutes; this is divided by the number of workers assigned to the machine. The cycle time for Machine 2 is 200 minutes; this is divided by the number of workers assigned to the machine.

After each part is processed by the first machine it must go to Machine 3 which is a machining process. The cycle time for Machine 3 is fixed at 500 minutes and this process requires 1 worker, however the quantity of Machine 3 can be varied to reduce blocking at the previous workstations. After being processed by Machine 3, the parts are shipped.

All of the input variables and their possible values are shown in Table 7-2. The following output variables are created for every scenario. The brackets show the shortened names used to refer to these variables within this section.

- Throughput

- Mean time in system for Part 1 and Part 2 (MeanTimePart1, MeanTimePart2)
- Percent utilisation for crane and labour pool including busy and idle time (LabourIdle, LabourBusy, CraneIdle, CraneBusy)
- Percent utilisation for machines 1,2 and 3 including busy, idle, waiting for labour and blocked time (Machine1Busy, Machine1Idle, Machine1WaitLab, Machine1Blocked etc.)

6.3.2 Application of Framework for Test Case 1

The framework outlined in Figure 6-1 has been applied to Case Study 1 to demonstrate the framework and assess if there are any benefits of the visualisation method.

Initially a Design of Experiments was created which included all permutations of all of the variable input values from Table 6-2. The Design of Experiments was then run using the simulation model to create results for each of the 216 runs for 132,000 simulated minutes, in order to ensure the model has settled sufficiently.

The results have been used to grow regression trees for a number of the outputs. Subsequently pivot tables have been created in Excel to create the average values for alternative objectives based on the chosen configurations.

Figure 6.4 shows a number of the regression trees generated. The first regression tree is generated for the ‘throughput’ objective function. Initially the sum of squares is calculated, then a number of different variables are tested to find the variable which produces the biggest reduction in sum of squares when applying the binary split. For the throughput objective function this is found to be the ‘NLabMach2’ variable. Hence the binary split is applied and the new values of sum of squares are calculated. Subsequently, new binary splits are carried out on the resulting leaves.

Table 6-2 - A table to show the fixed and variable inputs for case study 1

	Input	Values
Fixed values	Cycle time machine 1	100/number of workers
	Cycle time machine 2	200/ number of workers
	Cycle time machine 3	500
Variables	Total labour pool	5,7,10
	Inter-arrival time of parts	10,100,1000
	Number of workers at machine 1	1,2
	Number of workers at machine 2	1,2
	Number of cranes available	1,2,3
	Quantity of Machine 3	1,2

The following subsections detail testing used to identify any trends for output objectives of regression trees and an example application of the framework on case study 1 to visualise multiple objectives on the same regression tree. The example is validated using testing.

6.3.2.1 Identifying Trends for Regression Trees with Multiple Objectives

With large scale manufacturing systems there may be a significant number of potential objective functions. In this sub-section a number of regression trees have been generated for case study 1 for a large variety of output objective functions in order to ascertain if there are trends relating to regression trees for a variety of different output objectives. If there are trends relating to multiple objectives, it may provide the opportunity for a simplified regression tree creation method which simultaneously summarises regression trees for many objectives. In particular, correlated variables are tested to ascertain if they produce identical regression trees.

In order to identify any trends, regression trees were created initially for 6 of the different output objectives in R. The 'throughput', 'mean time part 1' and 'mean time part 2' outputs were chosen as they summarise system performance and 3 other machine specific output variables were chosen. The regression tree growing algorithm identifies the best binary splits to reduce the sum of squared errors *S.* It is of particular interest to identify if there are any trends in the definition of the binary splits amongst the various regression trees. The resulting regression trees are shown in Figure 6-4.

From Figure 6-4 a number of observations have been made. From the figure it appears that the number of nodes changes between generated trees with no visible trend, however the first binary split for the output objectives involves either the 'NLabMachine2' or 'ArrTime' variable.

It has been noted that the graphs for the 'mean time for part 1' variable and 'mean time for part 2' variable are identical. It is expected that these variables may be have identical regression trees due to an underlying relationship between the variables. In order to determine the relationships between all of the output variables, a correlation matrix has been created as shown in Table 6-3. The correlation matrix shows Pearson's correlation coefficient for each pair of variables x, y. The formula for calculating Pearson's correlation coefficient on a pair of variables is given in equation 6.4.

$$\frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}} \quad \textbf{(6-4)}$$

From the correlation matrix it can be seen that, as expected, the 'mean time for part 1' and 'mean time for part 2' output variables are very highly correlated; the variables have a correlation coefficient of 1. A number of other pairs of output variables were identified as being highly correlated from the correlation matrix in Table 6-3, hence further analysis of pairs of correlated output variables is required. It is of particular interest to ascertain if pairs of correlated output variables produce identical regression trees.

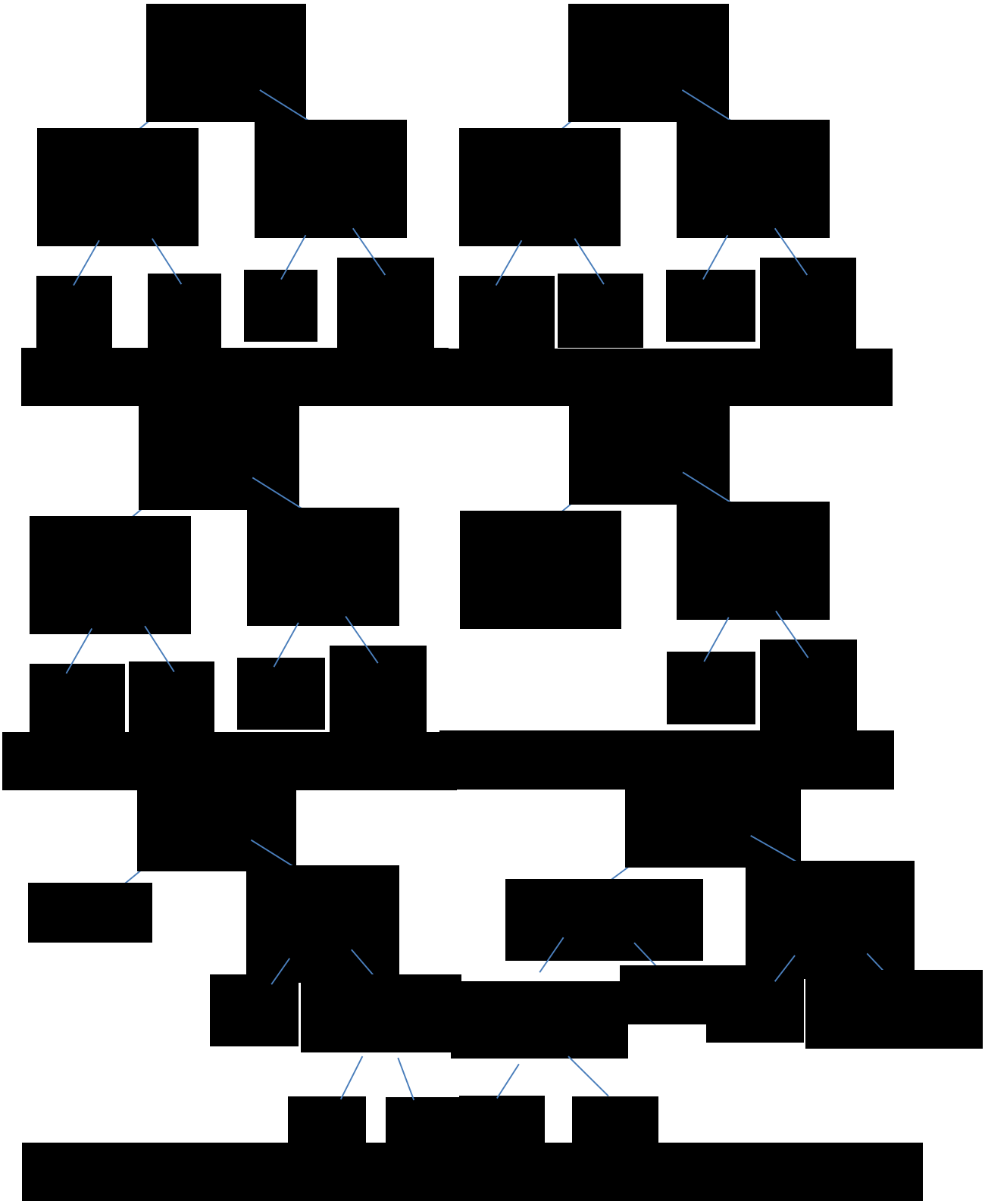


Figure 6-4 – Regression trees created for 6 different outputs

Table 6-3 – A table to show the correlations between output variables for case study 1

	MeanTimePart1	MeanTimePart2	CraneBusy	CraneIdle	LabourBusy	LabourIdle	Machine1Busy	Machine2Busy	Machine3Busy	Machine1Idle	Machine2Idle	Machine3Idle	Machine1Blocked	Machine2Blocked	Machine1WaitLab	Machine2WaitLab	Machine3WaitLab	Throughput
MeanTimePart1	1.00	1.00	-0.05	0.05	-0.02	0.02	0.05	0.12	-0.04	0.02	-0.02	0.00	-0.14	-0.11	0.11	0.11	0.04	-0.13
MeanTimePart2	1.00	1.00	-0.06	0.06	-0.03	0.03	0.03	0.15	-0.04	0.05	-0.05	0.03	-0.16	-0.13	0.10	0.15	0.04	-0.15
CraneBusy	-0.05	-0.06	1.00	-1.00	0.07	-0.07	-0.08	-0.26	-0.58	-0.16	0.33	0.28	0.20	-0.05	-0.05	-0.12	0.58	0.35
CraneIdle	0.05	0.06	-1.00	1.00	-0.07	0.07	0.08	0.26	0.58	0.16	-0.33	-0.28	-0.20	0.05	0.05	0.12	-0.58	-0.35
LabourBusy	-0.02	-0.03	0.07	-0.07	1.00	-1.00	-0.03	-0.15	0.04	-0.05	0.03	-0.07	0.16	0.13	-0.10	-0.14	-0.03	0.18
LabourIdle	0.02	0.03	-0.07	0.07	-1.00	1.00	0.03	0.15	-0.04	0.05	-0.03	0.07	-0.16	-0.13	0.10	0.14	0.03	-0.18
Machine1Busy	0.05	0.03	-0.08	0.08	-0.03	0.03	1.00	-0.42	-0.06	-0.67	0.50	-0.31	-0.37	0.11	0.80	-0.38	0.11	-0.19
Machine2Busy	0.12	0.15	-0.26	0.26	-0.15	0.15	-0.42	1.00	-0.24	0.67	-0.42	0.57	-0.61	-0.74	0.06	0.93	0.18	-0.81
Machine3Busy	-0.04	-0.04	-0.58	0.58	0.04	-0.04	-0.06	-0.24	1.00	-0.09	-0.34	-0.59	0.39	0.58	-0.29	-0.37	-0.99	0.26
Machine1Idle	0.02	0.05	-0.16	0.16	-0.05	0.05	-0.67	0.67	-0.09	1.00	-0.22	0.52	-0.23	-0.60	-0.53	0.69	0.02	-0.30
Machine2Idle	-0.02	-0.05	0.33	-0.33	0.03	-0.03	0.50	-0.42	-0.34	-0.22	1.00	-0.20	-0.05	-0.21	0.18	-0.35	0.39	0.15
Machine3Idle	0.00	0.03	0.28	-0.28	-0.07	0.07	-0.31	0.57	-0.59	0.52	-0.20	1.00	-0.44	-0.70	0.01	0.79	0.50	-0.40
Machine1Blocked	-0.14	-0.16	0.20	-0.20	0.16	-0.16	-0.37	-0.61	0.39	-0.23	-0.05	-0.44	1.00	0.67	-0.71	-0.61	-0.36	0.90
Machine2Blocked	-0.11	-0.13	-0.05	0.05	0.13	-0.13	0.11	-0.74	0.58	-0.60	-0.21	-0.70	0.67	1.00	-0.15	-0.84	-0.52	0.72
Machine1WaitLab	0.11	0.10	-0.05	0.05	-0.10	0.10	0.80	0.06	-0.29	-0.53	0.18	0.01	-0.71	-0.15	1.00	0.04	0.31	-0.58
Machine2WaitLab	0.11	0.15	-0.12	0.12	-0.14	0.14	-0.38	0.93	-0.37	0.69	-0.35	0.79	-0.61	-0.84	0.04	1.00	0.29	-0.76
Machine3WaitLab	0.04	0.04	0.58	-0.58	-0.03	0.03	0.11	0.18	-0.99	0.02	0.39	0.50	-0.36	-0.52	0.31	0.29	1.00	-0.22
Throughput	-0.13	-0.15	0.35	-0.35	0.18	-0.18	-0.19	-0.81	0.26	-0.30	0.15	-0.40	0.90	0.72	-0.58	-0.76	-0.22	1.00

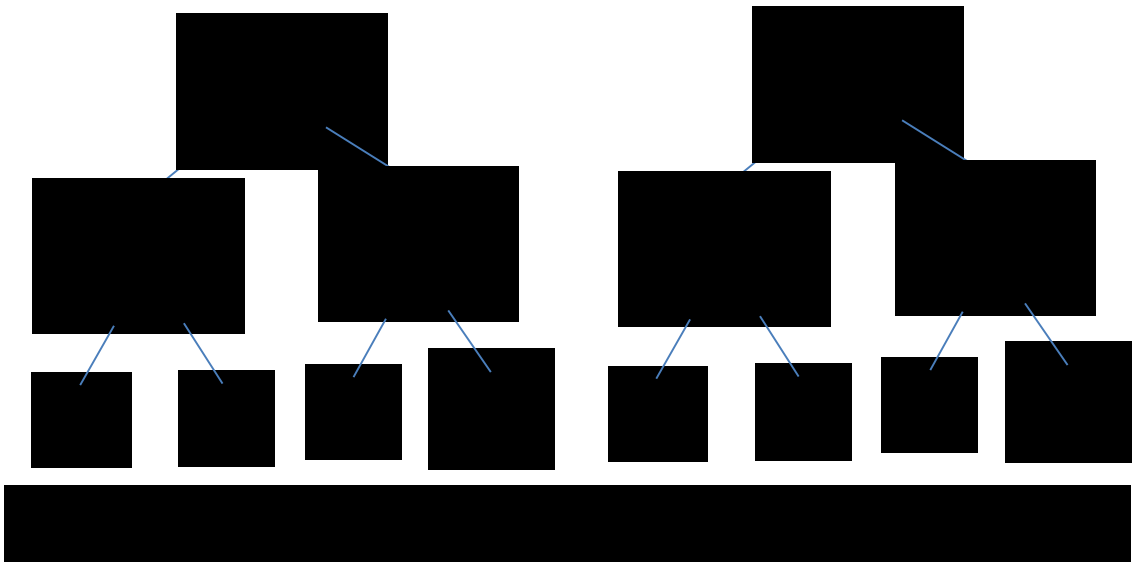


Figure 6-5 - Regression trees for 'LabourIdle' and 'LabourBusy' variables with correlation coefficient -1

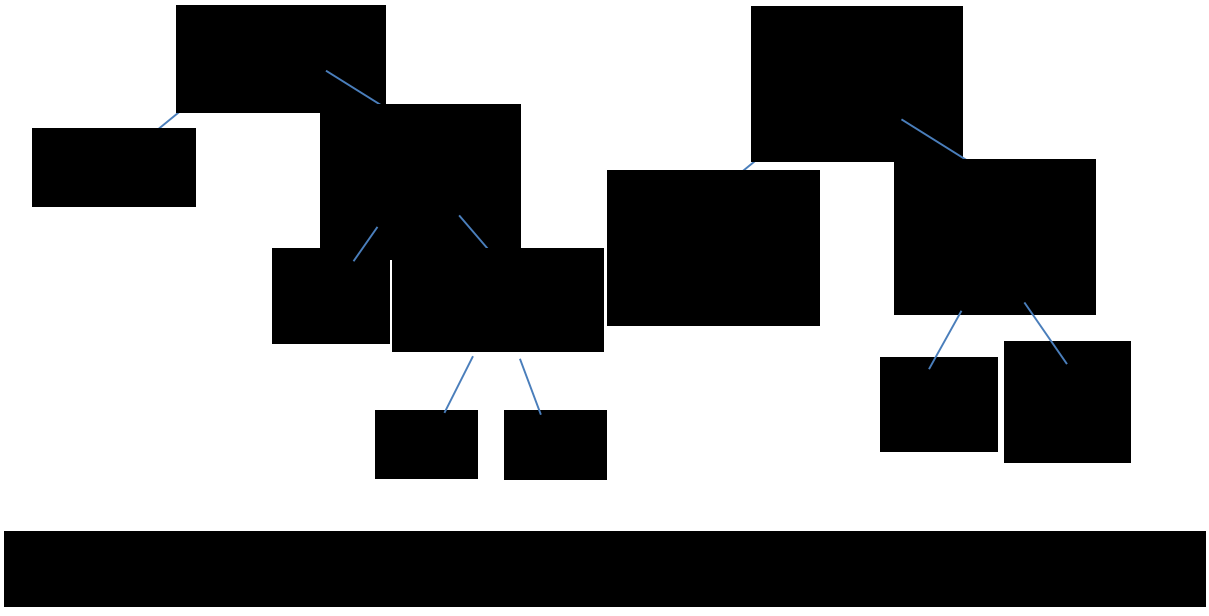


Figure 6-6 - Regression trees for 'Machine2WaitLab' and 'Machine2Busy' variables with correlation coefficient 0.93

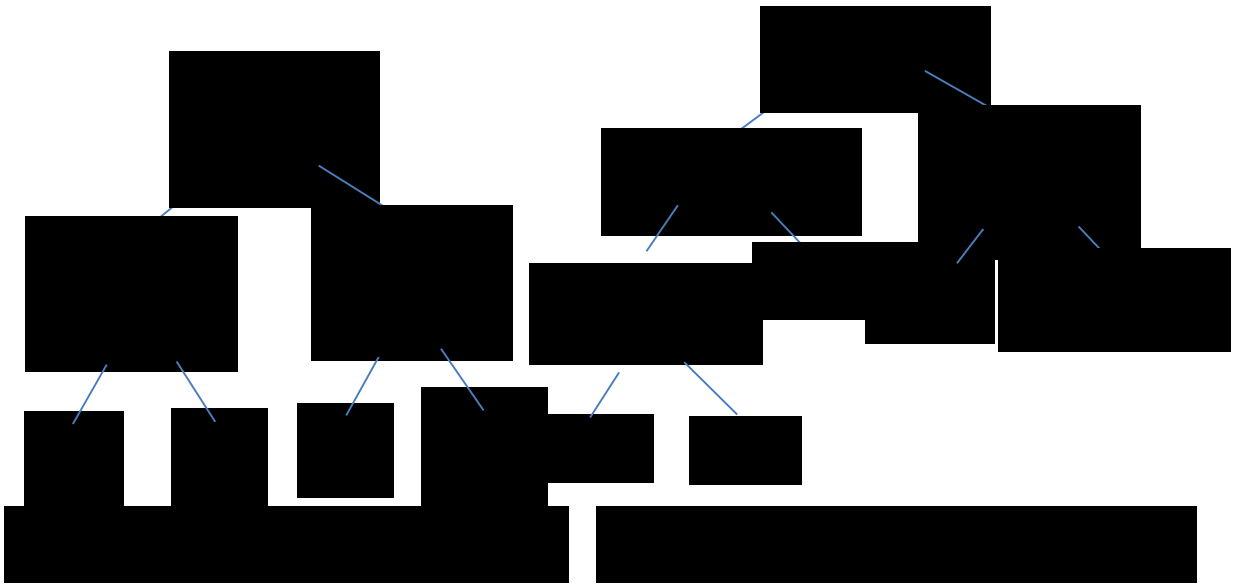


Figure 6-7 - Regression trees for ‘Throughput’ and ‘Machine1Blocked’ variables with correlation coefficient 0.90

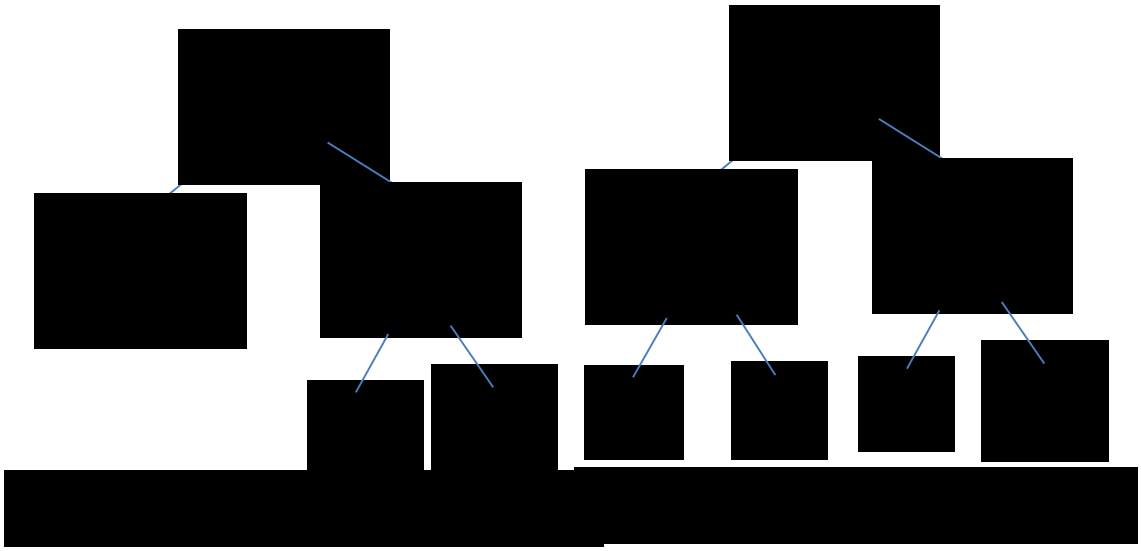


Figure 6-8 - Regression trees for ‘Machine2Busy’ and ‘Throughput’ variables with correlation coefficient -0.81

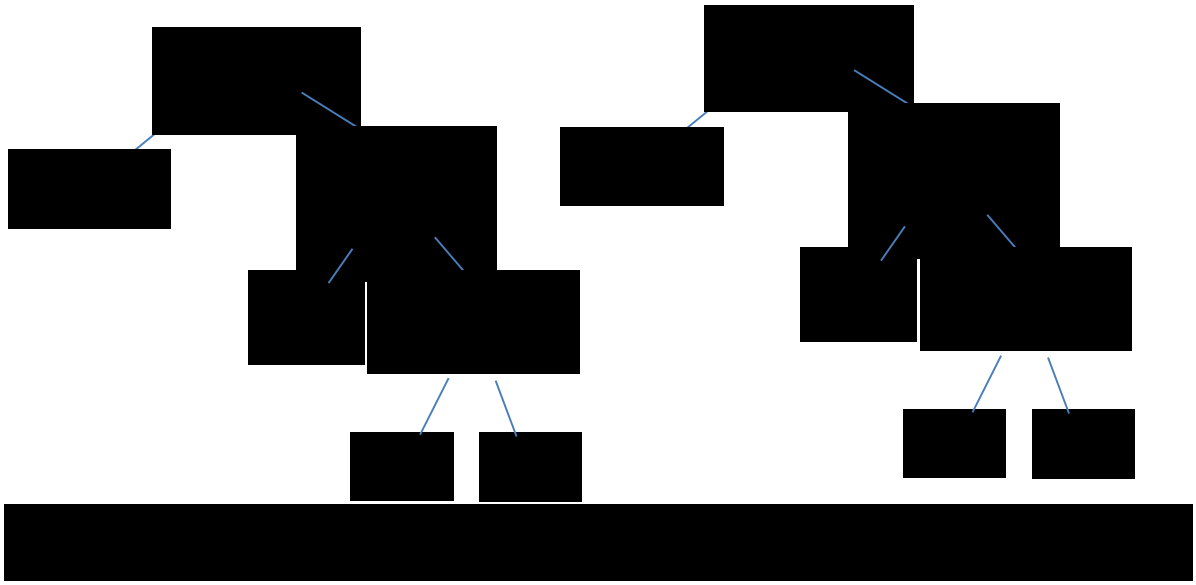


Figure 6-9 - Regression trees for 'Machine2WaitLab' and 'Machine3Idle' variables with correlation coefficient 0.79

From Table 6-3 pairs of correlated have been used to create regression trees in R. Some of the results are shown in Figures 6-5 to 6-9 with the correlation coefficient given for each pair of regression trees. Pairs were chosen which have a correlation coefficient of 0.79 or higher or -0.79 or lower.

Figure 6-5 shows the regression trees created for output variables 'LabourBusy' and 'LabourIdle' which have a correlation coefficient of -1, these are utilisation values where the sum of the two output variables is 100. Each of these generated regression trees is a mirror of the correlated output variable tree. The variables are highly correlated due to the nature of the variables as one can be calculated from the other; hence the creation of a unified tree by overlaying the information onto one tree will not provide users with any further information.

Figures 6-7 and 6-8 show the 'throughput' against machine utilisation statistics. For each of these scenarios the first binary cut is identical, with the remainder of the tree containing different cuts. Figure 6-9 is the only comparison shown to have a different binary split for the first node.

For Figure 6-9 the utilisation output variables have a correlation coefficient of 0.99 due to their interdependency, however the generated regression trees are not mirror images of one another. The first node is mirrored; however, the

‘Machine3WaitLab’ variable has one extra node within the tree and a different binary split on the second level. This is particularly surprising due to the high correlation between these two output variables and the interdependency between the variables as they are both utilisation statistics for the same machine.

The utilisation output variables for ‘Machine3Busy’ and ‘Machine3WaitLab’ have a correlation coefficient of 0.99 due to their interdependency, however the generated regression trees were found not to be mirror images of one another and had a different number of nodes. This is particularly surprising due to the high correlation between these two output variables and the interdependency between the variables as they are both utilisation statistics for the same machine.

From the results it is appears that output variables with a correlation coefficient of -1 have regression trees which are mirror images of each other; however, regression trees for pairs of variables with any other correlation coefficients vary considerably.

It is not clear how the information from these results can be used to generate simplified regression trees for multiple objectives. For variables with a correlation of 1 or -1 there is no real benefit of providing the information about both variables on the same regression tree. It is expected that the most benefit will arise from the visualisation of uncorrelated and conflicting output objectives on the same regression tree.

In the following sub-section the methodology demonstrated in Figure 6-1 is carried out on case study 1 with the results for multiple objectives overlaid on a regression tree.

6.3.2.2 Example Application for Multiple Objectives

This sub-section demonstrates the process of regression tree creation for multiple objectives for case study 1. For this particular example a regression

tree is grown for the ‘mean time for Part 1’ objective and the information regarding the ‘throughput’ objective average results are overlaid on the regression tree at the nodes and leaves of the tree. The generated regression tree is validated using 4 different tests.

In the first stage the results have been created for the simulation model. As the results are difficult to understand through analysis of a spreadsheet, a regression tree has been created using R. The generated regression tree is demonstrated in Figure 6-10. The information displayed on the tree includes: the question at each binary split, the average value of the output variable at each split and leaf and the total number of results within each node or leaf.

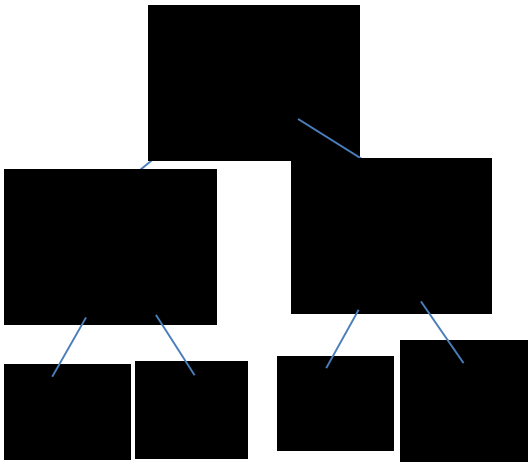


Figure 6-10 – A regression tree for case study 1- for ‘mean time for Part 1’ output variable.

Following the creation of the regression tree within R, the average values of ‘throughput’ objective have been calculated for each node or leaf in the tree using pivot tables within Excel. Table 6-4 demonstrates the creation of average throughput values for varying arrival times for the first node of the tree. The grand total of the average throughput has been used for the first node. This table has also been used to select the average throughput for the second node on the second level.

Table 6-4 – A table to show the pivot table created for case study 1 for average throughput

Arrival Times	Average of Throughput	Average of MeanTimePart1
100	2.5131875	60245.75048
250	2.511825	51771.70165
500	2.4968375	37709.20165
Grand Total	**2.507283333**	**49908.88459**

Following the initial table (6-4), a further pivot table has been created to analyse the average throughput value for the first node on the second level of the regression tree, this is shown in Table 6-5. The highlighted value shows the relevant average throughput value for the first leaf on the 3rd level of the regression tree. Following the creation of the pivot tables, the relevant average throughput values have been collected and used to create a regression tree which includes all of the information. This is demonstrated in Figure 6-11. Within this regression tree, nodes include questions highlighted in bold, average 'mean time for part 1' values, average 'throughput' values and the number of observations at each node.

Table 6-5 - A table to show the pivot table created for case study 1 for average throughput

Arrival times/ number of labourers at machine 2	Average of Throughput	Average of MeanTimePart1
100	**2.5131875**	**60245.75048**
1	2.29635	60558.5117
2	2.730025	59932.98925
250	**2.511825**	**51771.70165**
1	2.293625	52874.1953
2	2.730025	50669.208
500	**2.4968375**	**37709.20165**
1	2.282725	40061.6953
2	2.71095	35356.708
Grand Total	**2.507283333**	**49908.88459**

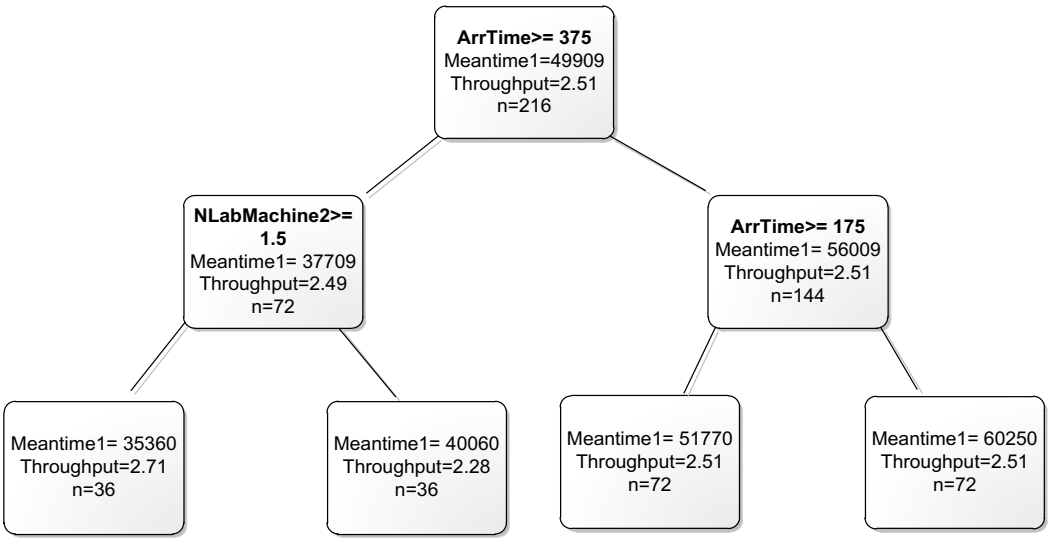


Figure 6-11 – A regression tree for Mean Time for part 1 with average values for throughput

The regression tree in Figure 6-11 demonstrates the application of the proposed framework in Figure 6-1 for visualising multiple objectives. The generated regression tree is capable of demonstrating the impact of input variable decisions on multiple objectives simultaneously.

In order to validate the proposed regression tree, 4 tests have been carried out to ensure the results produced for each leaf node are reasonably accurate. The results are printed in Table 6-6. The table outlines the 4 tests with the nodes tested and the resulting expected values against actual values. The percentage difference is given in the last column. The input values for all other variables were chosen randomly and kept fixed for all 4 tests.

The results suggest that for the 4 different tests carried out, the generated results were generally within 5% of the expected output values, with the exception of one of the tests for the output variable where the value was 5.5% lower than the expected value- this is highlighted in bold in the table.

Table 6-6 – A table to show the validation of the generated regression tree for 4 different test conditions

Test conditions	Output objective	Expected output values	Actual output values	% difference
ArrTime>=375 and NLabMachine2>=1.5	MeantimePart1	35360	33560	5.09
	Throughput	2.71	2.86	**-5.53**
ArrTime>=375 and NLabMachine2<1.5	MeantimePart1	40060	38669	-3.47
	Throughput	2.28	2.39	-4.82
ArrTime<375 and ArrTime>=175	MeantimePart1	51770	52044	-0.53
	Throughput	2.51	2.39	4.78
ArrTime<175	MeantimePart1	60250	60062	0.31
	Throughput	2.51	2.39	4.78

6.3.3 Summary of results from Test Case 1

This section has created a small manufacturing system test case and applied the framework in Figure 6-1 to the case study. Various regression trees have been created for different objective functions and a correlation matrix has been created to assist with identifying trends. Regression trees of output objective functions with a correlation coefficient of -1 were found to be mirror images of each other. Output objective functions with other correlation coefficients were found to have different regression trees. The framework in Figure 6-1 was

applied to the test case to create a regression tree which shows the impact of decision variables on multiple objectives simultaneously. The generated regression tree is validated using 4 different tests; the results were found to generally lie within 5% of the actual simulation model.

Based on the validation and results this method of generating regression trees can be used to understand the main trends in output variables for varying conditions for a small scale test case. It is expected that this method of visualisation will be particularly valuable for conflicting objectives. Subsequent sections test out this methodology on 2 larger scale test cases with conflicting objectives.

6.4 Test Case 2 – Medium value production line

A medium size DES model has been developed as a second test case with conflicting objectives in order to test the framework outlined in Figure 6-1. This test case has been developed as an alternative medium volume manufacturing production line.

6.4.1 Description of the Manufacturing Process

Case study 2 is a medium size DES model for testing the regression tree visualisation framework with multiple conflicting objectives. The DES model has been created in WITNESS to represent a generic assembly line for small to medium size components. The DES model represents a ‘single piece flow’ medium volume assembly line which processes 3 different parts. A screenshot of the developed DES model is shown in Figure 6-12. The assembly line has 8 workstations with buffers between each workstation and 1 input buffer. All parts much be processed by all workstations, there are no cranes or materials handling processes required between workstations. The last workstation (Machine 8) represents an automated process which requires a labourer to set

up the machine to run; subsequently the labour is released whilst the automated task is carried out by the workstation.

The input buffer and last buffer both have variable capacities; all of the other buffers have a capacity of 3. Every workstation must have a labourer to complete the associated task with the exception of Machine 8, as previously described. The labour pool works on a daily 8 hour shift; this is outlined in Table 6-7.

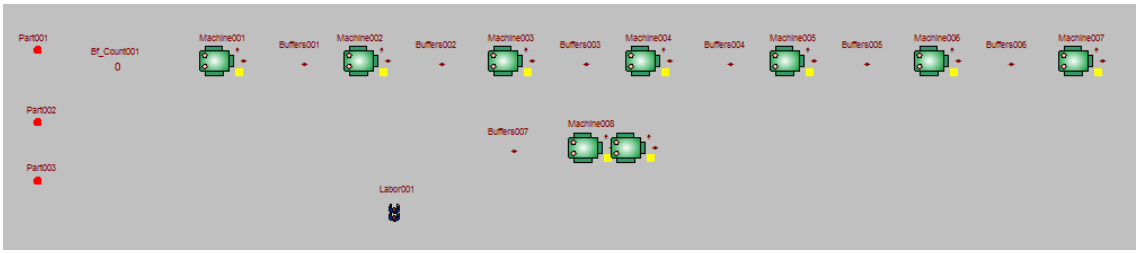


Figure 6-12 – A screenshot of the WITNESS DES model for case study 2

The key outputs of interest for this particular system are the throughput and the labour cost. It is expected that these two output objectives are conflicting with each other due the formulation of the simulation model. The labour cost is measured as the total number of man hours required per week to run the production line.

Outputs of the system to be measured are:

- Throughput
- Number of parts shipped
- Man hours per week
- Weighted average WIP (Part 1, Part 2 and Part3)
- Mean time in system (Part 1, Part 2 and Part3)
- Percent utilisation: busy, idle (labour)
- Percent utilisation: busy, idle, wait labour, blocked (machine 1,2,3,4,5,6,7 and 8)

Table 6-7 - A table to show the daily shift pattern of workers for case study 2

	Input	Values
Fixed values	Cycle time machine 1	15
	Cycle time machine 2	17
	Cycle time machine 3	10
	Cycle time machine 4	17
	Cycle time machine 5	12
	Cycle time machine 6	10
	Cycle time machine 7	12
	Cycle time machine 8 set up	5
	Cycle time machine 8 automated	35
Variables	Total labour pool	3,5,7
	Inter-arrival time of parts	10, 40, 70, 100
	Input buffer size	3,5
	Number of workstations Machine8	1,2,3
	Machine 8 buffer size	5,7

Table 6-8 – A table to show the fixed and variable inputs for case study 2

Time period	Period length
Working time	180
Break	15
Working time	105
Lunch	60
Working time	120
Break	15
Working time	75
End of shift	870

In order to select the simulation run length, a test has been carried out with the average WIP output for each part type. The average WIP for each part has been recorded every 5 simulated minutes in order to generate the graph in Figure 6-13. The graph can be used to visualise the convergence of the output value. It can be seen that at reading 3340 the values of the 3 KPIs are converging. The values seem well converged around reading 6202, hence a simulation run length of 6000*5 (30,000) has been chosen to ensure convergence of the KPIs.

6.4.2 Application of the framework for Test Case 2 for two conflicting objectives

The variable values from Table 6-8 were used to create a Design of Experiments. All of the combinations of the input variables were created resulting in 143 different simulation runs. The generated Design of Experiments has been subsequently run using the simulation model. The generated results have been used to create pivot tables in Excel.

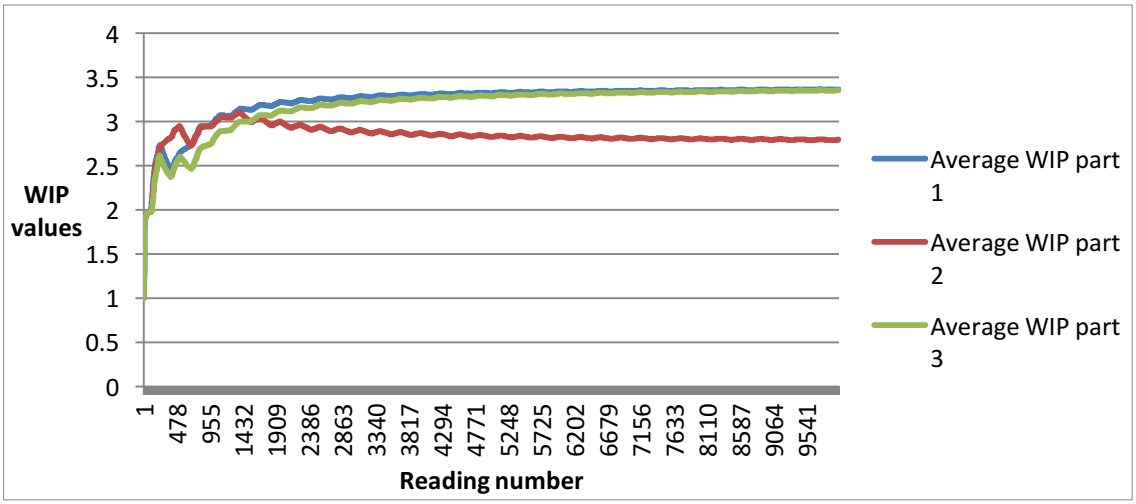


Figure 6-13 – A graph to show the weighted average of WIP values for each part type, reading were taken in 5 simulated minute intervals

The results have been used to grow regression trees for given output objectives. In particular, for this test case conflicting output objectives of throughput and man hours per week were chosen to be analysed. The framework from Figure 6-1 has been applied for each of these objectives. Figure 6-14 demonstrates the initial regression trees created for each objective.

The generated regression trees are noticeably different from one another. For this particular test case the framework in Figure 6-1 has been applied to each of the conflicting objectives.

For the throughout objective function, a pivot table has been created in Excel for the average man hours per day as shown in Table 6-9. This information can be overlaid on the regression tree for the throughput.

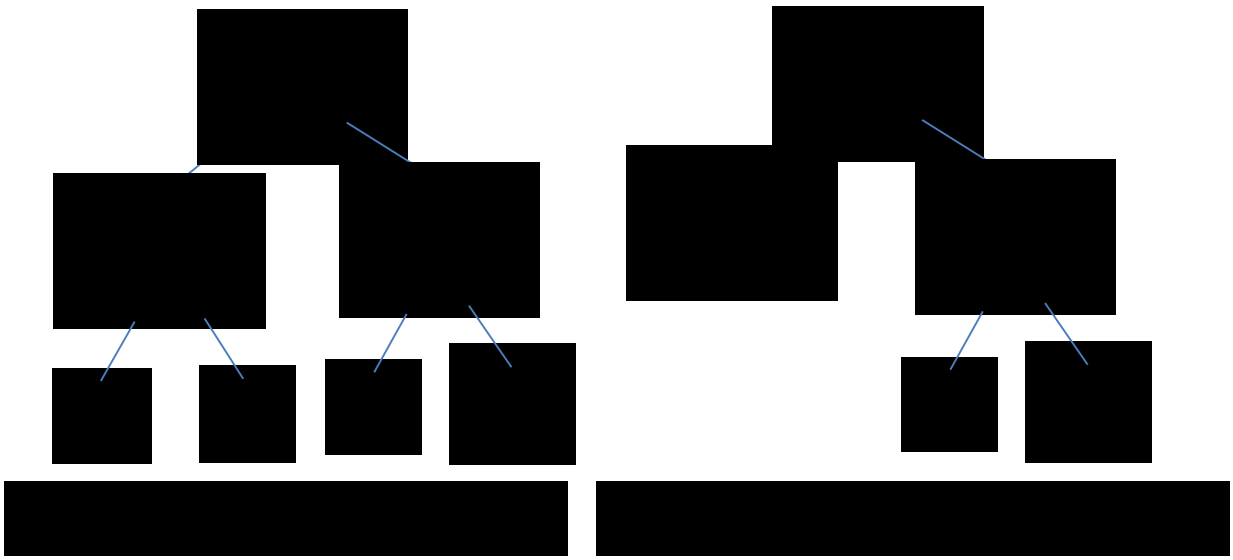


Figure 6-14 – Regression trees for throughput and man hours per day, correlation coefficient of 0.50

Table 6-9 – A table to show the pivot table created for case study 2 for average man hours per day

Average Man hours per day		
NMachine8	NLabAvail	Total
1	3	24
	5	40
	7	56
1 Total		**40**
2	3	24
	5	40
	7	56
2 Total		**40**
3	3	24
	5	40
	7	56
3 Total		**40**
Grand Total		**40**

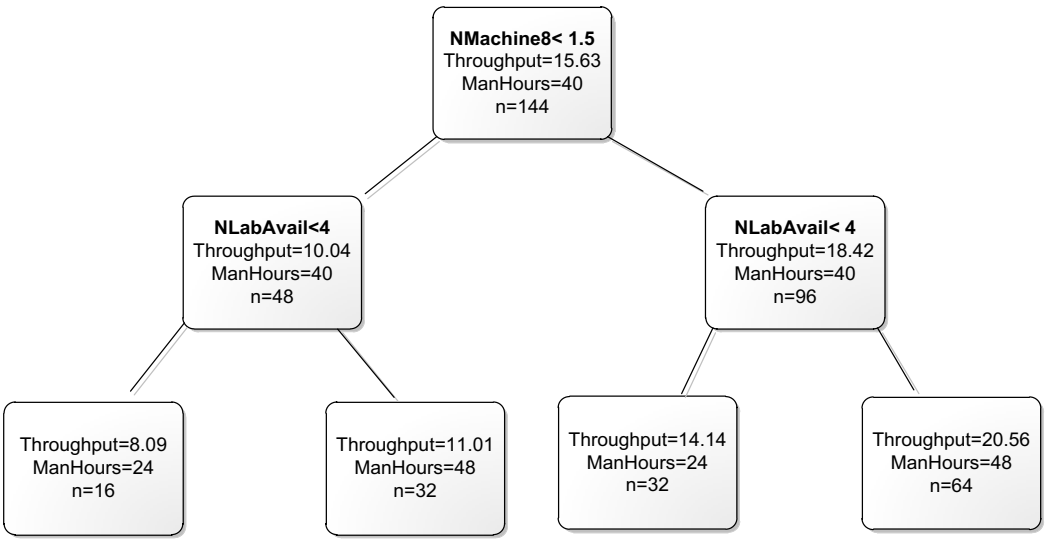


Figure 6-15 – A regression tree to show the throughput and 'man hours per day' results overlaid

The pivot table in Table 6-9 and original regression tree for throughput have been used to create the final regression tree with respect to throughput with the overlaid values for man hours and number of observations in each node or leaf; the tree is given in Figure 6-15.

The regression tree has been validated using different tests for each of the leaf nodes. For all of the tests one set of input values has been chosen randomly and kept fixed; only the variables for each node have been changed between each test. Table 6-10 shows the results of the validation for 4 different tests. It can be seen that the percentage difference between the expected outputs and actual values from the simulation model varies considerably between 0 and 20%.

Table 6-10- A table to show the validation of the generated regression tree for 4 different test conditions

Test conditions	Output objective	Expected output values	Model output values	% difference
NMachine8<1.5 and NLabAvail<4	ManHours	24	24	0
	Throughput	8.09	8.1120	0.27
NMachine8<1.5 and NLabAvail>=4	ManHours	48	40	**-20**
	Throughput	11.01	10.032	**-9.75**
NMachine8>=1.5 and NLabAvail<4	ManHours	24	24	0
	Throughput	14.14	14.448	2.13
NMachine8>=1.5 and NLabAvail>=4	ManHours	48	40	**-20**
	Throughput	20.56	20.160	-1.98

For the ‘man hours per day’ objective function the same method has been followed to create a regression tree with all relevant information. A pivot table has been created as shown in Table 6-11. This pivot table has been used together with the regression tree in Figure 6-14 to generate the final regression tree for this objective function, the tree is given in Figure 6-16.

Table 6-11 - A table to show the pivot table created for case study 2 for average throughput

Average of throughput	
NLabAvail	Total
3	12.12
5	16.837
7	17.92
Grand Total	15.63

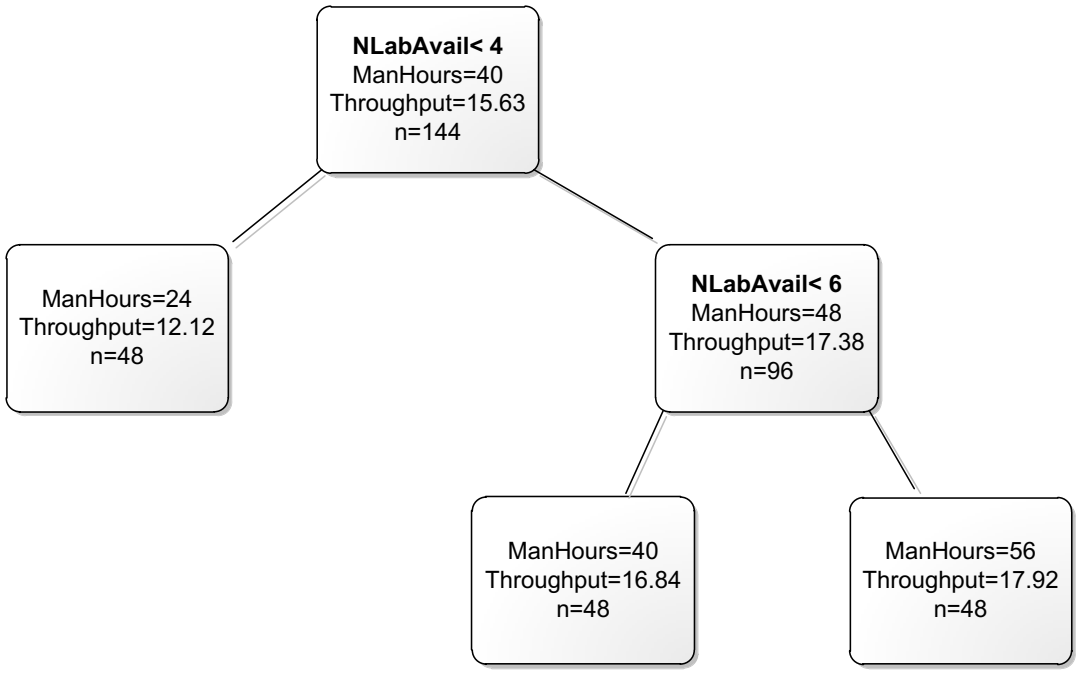


Figure 6-16- Regression tree for manhours per day with throughput values overlaid

The generated regression tree with multiple objectives has been validated with tests for each of the leaves of the tree. The model has been run with each of the test conditions for the tree leaves; the results are given in Table 6-12.

From Table 6-12 it can be seen that the results for ‘man hours’ are identical to the expected value given by the regression tree, however the results for the throughput are

all around 15% higher than the expected values. Sensitivity analysis could be carried to confirm if this is due to the single run input variables being chosen at random.

Table 6-12 - A table to show the validation of the generated regression tree for 4 different test conditions

Test conditions	Output objective	Expected output values	Actual output values	% difference
NLabAvail<4	ManHours	24	24	0
	Throughput	12.12	14.45	**16.12**
NLabAvail>=4 and NLabAvail<6	ManHours	40	40	0
	Throughput	16.84	20.16	**16.47**
NLabAvail>=4 and NLabAvail>=6	ManHours	56	56	0
	Throughput	17.92	20.88	**14.18**

Although the percentage difference between the expected output values and actual output values can be up to 20%, the regression trees created in Figures 6-15 and 6-16 can be used to visualise the trends and relationships between the input and output variables. The ‘man hours per day’ variable was found to have 0% difference between predicted and observed values for the test, suggesting that the method provides good prediction for this objective function.

Figure 6-15 demonstrates that the most important independent variable for throughput is ‘number of machine 8’. The regression tree shows that, as the number of ‘number of machine 8’ increases, the throughput also increases. In addition, it can be seen that as the labour availability increases, the throughput also increases considerably. It is noted that the leaf nodes in this Figure have a

varying number of observations; one leaf has 16 observations and another leaf has 64.

Interestingly Figure 6-16 shows the number of labour available as being the main variable with has an impact on the ‘man hours per day’. This independent variable is used at the only 2 nodes to create the binary split. This choice of independent variable is intuitive due to the formulation of the simulation model where the ‘man hours per day’ is the labour available multiplied by 8. Figure 6-16 also has an identical number of observations at each leaf node, n=48 at each leaf.

The results of the regression trees for these two objectives have shown errors of up to 20% between predicted and actual values. These errors suggest that although these decision trees may be able to assist users in understanding trends, they are not ideal for predictions. Regression trees can have a variety of drawbacks. For instance, regression trees have high variance; a small change in the data can result in a completely different tree. In particular, regression trees are highly dependent on the initial binary split due to the hierarchical nature of the tree generation (Hastie, Tibshirani, & Friedman, 2008). Alternative statistical methods can be applied to alleviate these issues. Statistical methods such as random forests can be used to generate more reliable predictive capabilities and to generate key insights into variable importance. In the next sub-section random forests are applied to each conflicting objective function for case study 2 in order to attempt to improve prediction accuracy.

6.4.3 Application of Random Forests to Test Case 2

Random Forests are a popular machine learning method used to improve prediction accuracy. Random forests create an ensemble of trees to improve the accuracy of individual decision trees. Random forests use sampling with replacement and aggregation, in particular ‘bootstrapping’ and ‘bagging’, on a number of decision trees. This method is able to alleviate the issues raised by regression trees.

The main algorithm for random forests, based on (Hastie et al., 2008), is given as follows:

1. Draw a bootstrapped sample of data from the training dataset
2. Grow a random-forest tree T_b for the bootstrapped data by repeating the following steps until termination criteria is reached
    - Select m variables from the total list of variables where $m < p$
    - Pick the best variable split point amongst the m variables
    - Split the node into two daughter nodes
3. Output ensemble of regression trees $\{T_b\}_1^B$

Then in order to carry out predictions for a given new point x for a regression tree equation 6-3 can be used. For regression trees the prediction is the average of the values generated by all trees in the forest.

Similarly, equation 6-4 can be used to predict the class of a new point: x. For classification trees the class is predicted by taking the majority vote of all the predicted classes for every generated classification tree.

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \quad \textbf{(6-3)}$$

$$\hat{C}_{rf}^{B}(x) = majority\ vote\{\hat{C}_B(x)\}_1^B \quad \textbf{(6-4)}$$

Parameters should be set in order to create the random forest. Main parameters are the value of m and the total number of trees in the generated forest. For this case study, $m = 3$ has been chosen initially, this value will be adjusted if it is found that the results are not converging sufficiently and producing reliable predictions. The number of trees to be produced has also been set as 1000.

In order to test random forests, the method has been applied to case study 2 for each conflicting objective function. The same data used to create the decision

trees in section 6.4.2 has been used to create random forests for each objective function The data has been partitioned into test and training data for the analysis of prediction accuracy; the ratio of test to training data used is 3:7. The following two sub sections detail the application of random forests for each objective function.

6.4.3.1 Throughput Objective Function

For the throughput objective function, the random forest package in R has been used to generate the random forests.

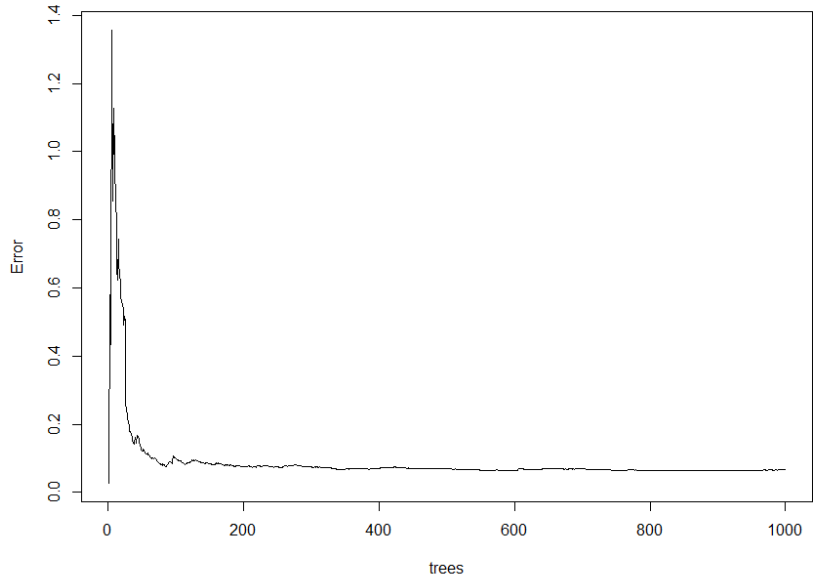


Figure 6-17 – A graph to show the prediction error against the number of regression trees

Upon creation of the random forest, Figure 6-17 has been used to determine if the number of trees is sufficient to minimise the prediction error. It can be seen from the plot that the error has converged, hence the number of trees is high enough.

$$MSE = \frac{S}{n} \quad \textbf{(6-5)}$$

Results included: $MSE = 0.06624$ where MSE. Is defined in equation 6.5 and S is defined in equation 6.2 The results suggest that 99.7% of the model variation is explained by the random forest.

The random forest results have been used to ascertain importance for the throughput objective function. Table 6-13 demonstrates the input variables with two measures of variable importance: percentage increase in MSE (%incMSE) and increase in node purity (IncNodePurity). The percentage increase in MSE is a value which gives the increase in MSE of predictions as a result of the chosen variable being permuted; higher values indicate more important variables. The equation is given in 6.6, where MSE_o is the out-of-bag mean square error.

$$\%incMSE_j = \frac{MSE_j - MSE_o}{MSE_o} 100\% \quad \textbf{(6-6)}$$

The IncNodePurity value represents the total decrease in sum of square errors S (defined in equation 6.2) from splitting on the variable, again higher values are preferred. Figure 6-18 shows the graphs of the two measures for each of the variables. Both graphs demonstrate the 'number of machine 8' variable and 'number of labour available' variable are the most important for the throughput objective function.

The test data has been used to validate the random forest method accuracy. The test data has used the random forest prediction given in equation 6-3. The results for the predicted values against observed values are plotted in Figure 6-19. The graph shows the predicted against observed results lying very close to the $y = x$ line. The mean square error for the test data produced was MSE=0.080931 which is reasonably close the value provided by training data.

Table 6-13 – A table to show the results of variable importance from the random forest for the 'throughput' objective

Input variables	%IncMSE	IncNodePurity
InputBufferSize	-6.02407	10.00861
LastBufferSize	-4.37528	12.32677
NMachine8	132.9116	1427.956
NLabAvail	120.3443	670.8516
Arrtime	-6.30275	23.85584

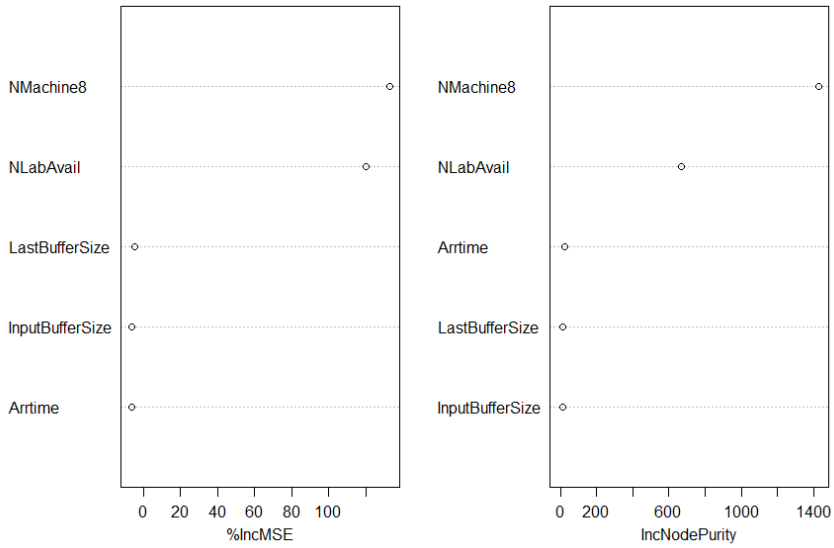


Figure 6-18 – Graphs to show the %incMSE and incNodePurity for each input variable for the throughput objective function

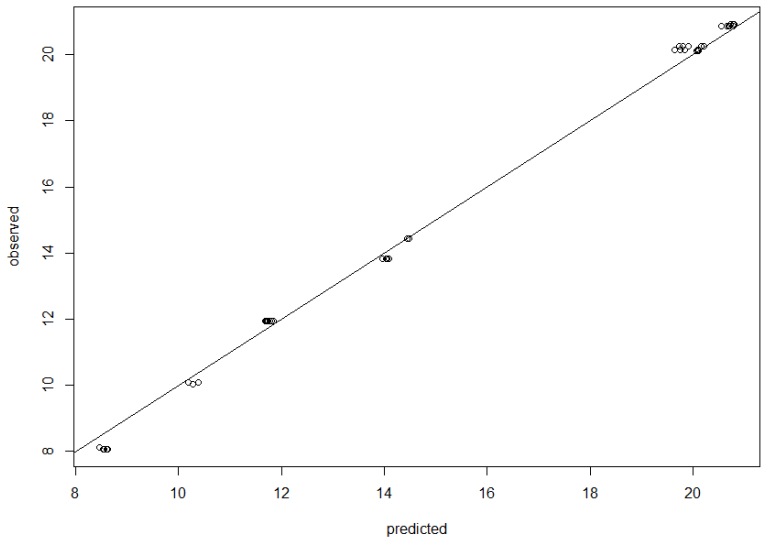


Figure 6-19 – A graph to show the observed values against predicted values for test data for the throughput objective function

Although the results show that predicted values aren't within 5% of the actual values, the results do indicate higher levels of accuracy than the predictive capabilities of the original regression tree created in Figure 6-15.

Further work can be carried out to tune the random forest algorithm by modifying the m parameter to test if this can improve prediction accuracy.

6.4.3.2 'Man Hours per day' Objective Function

A random forest has also been created for the 'man hours per day' objective function. The same training set and test set of data has been used for this example. For this output objective there are only 3 possible output values; hence a classification tree has been applied as opposed to a regression tree.

Due to the discrete nature of classification trees, an alternative method is required for creating binary splits. The Gini index is used as the measure to select the appropriate split for nodes. The Gini index is calculated by multiplying the probability of a data point being chosen by one minus the probability of misclassification. The formula for the Gini index is given in equation 6-7.

$$Gini\ Index = \sum_{i=1}^{C} p_i(1 - p_i) \quad \textbf{(6-7)}$$

For classification trees, alternative measures of variable importance are also required; these include mean decrease in accuracy and mean decrease in Gini. Mean decrease in accuracy is calculated by using out-of-the-bag samples on the generated decision trees. For a given variable *j,* the out-of-the-bag sample is passed down the tree and the prediction accuracy is measured, the value of *j* is then permuted and the accuracy is computed again. The decrease in accuracy of the prediction by permuting *j* is averaged over all trees to give the mean decrease accuracy value, higher values indicate important variables. The mean decrease in Gini is calculated by comparing the Gini index before and after a split for variable *j* and averaging over all generated trees.

The random forest has been created for the man hours per day objective function. The number of trees has been set to 1000 and the number of variables attempted at each split is set at 3.

The prediction error against the number of trees is shown in a graph in Figure 6-20. It can be seen that the prediction error falls to 0 after less than 100 trees; hence the out-of-the-bag estimate of the error rate provided for the random forest is 0%.

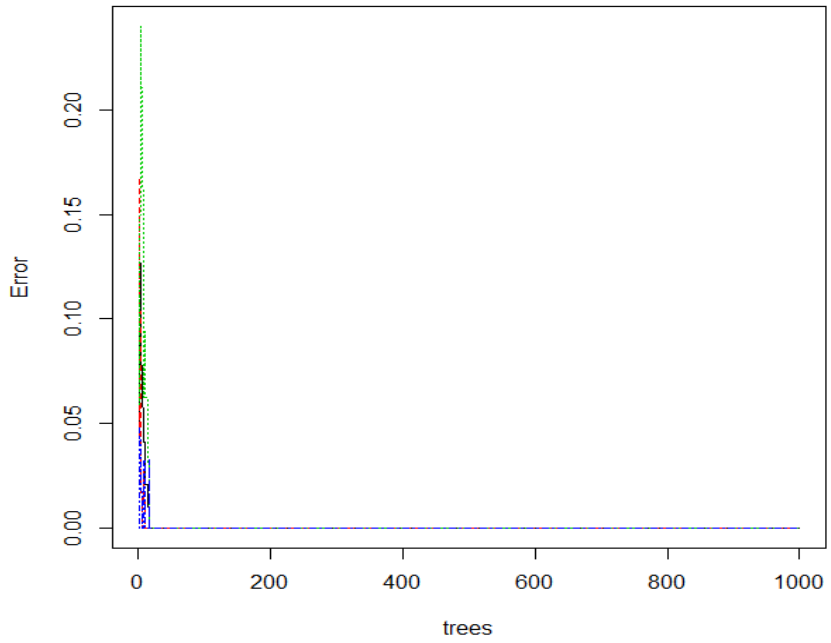


Figure 6-20 – A graph to show the prediction error against the number of classification trees within the random forest for the man hours per day objective function

Table 6-14 - A table to show the results of variable importance from the random forest for 'man hours per day' objective function

	24	40	56	MeanDecrease Accuracy	MeanDecrease Gini
InputBufferSize	-3.60892	-6.40576	-5.19392	-8.36373	0.459307
LastBufferSize	-5.32348	-6.18677	-2.35466	-7.70628	0.66331
NMachine8	-8.05729	-4.81818	-3.64336	-8.80484	1.094954
NLabAvail	134.2903	125.3658	131.7052	188.7738	61.42168
Arrtime	-8.15169	-7.99668	-3.48583	-10.1714	1.304934

The variable importance results are shown in Table 6-14. The results have also been used to create plots, as shown in Figure 6-21, to visualise the important variables. It can be seen that the 'NLabAvail' input variable has a much higher significance than the other variables. These results confirm that the original regression tree created in Figure 6-16 has summarised the key input variables for the problem.

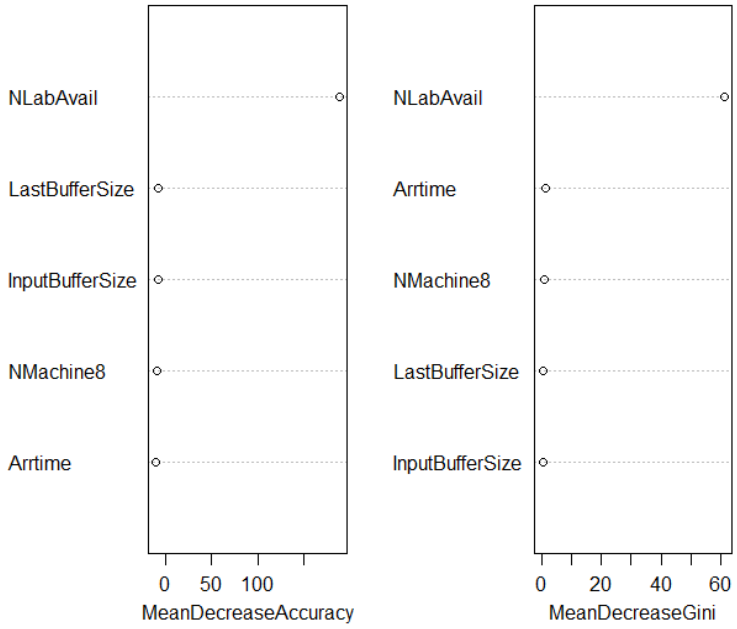


Figure 6-21 - Graphs to show the mean decrease in accuracy and the mean decrease in Gini index for each input variable for the man hours objective function

The random forest has been checked using the remaining test data. The model has been used to make predictions for the test data and the predicted values have been checked against the observations. It was found that there was no misclassification error for the test data. The graph shown in Figure 6-22 shows the predicted values against observed values for the test data. It can be seen that there is no error between the predicted and observed values.

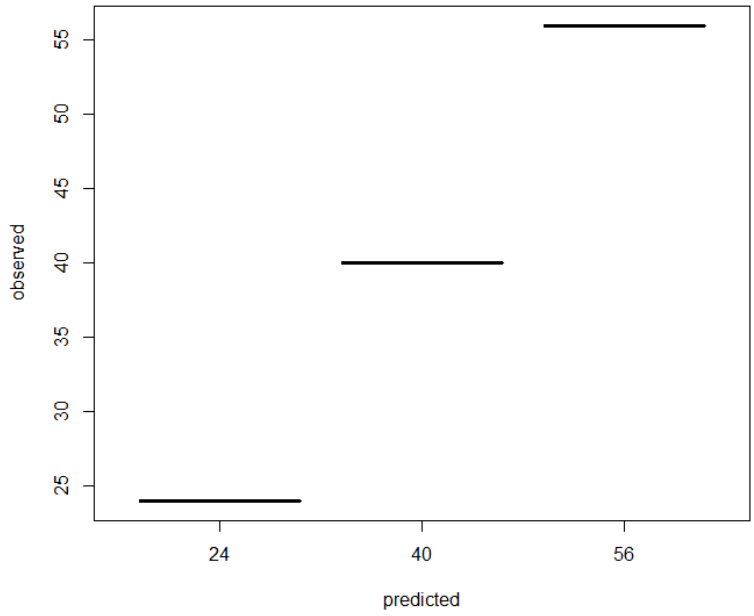


Figure 6-22 - A graph to show the observed values against predicted values for test data for the 'man hours per day' objective function

6.4.4 Summary of Results from Test Case 2

A number of observations and key insights have been made through the application of the decision trees and random forests to case study 2.

The regression trees that have been created for each conflicting objective function in Figures 6-15 and 6-16 are capable of capturing the general trends within the problem and displaying the trends in a simple manner; the trees are also quick and easy to interpret. The overlaying of information regarding conflicting objectives is highly useful as users can quickly ascertain the impact of decisions on multiple objective functions. However, a simple validation exercise was carried out on each of the generated regression trees and the results showed there was a 20% difference between the predicted and observed values. This prediction error is quite high and hence random forests have been proposed to assist with creating higher accuracy predictions.

The random forest method has been implemented based on the 'randomForest' package in R. Interestingly when creating a random forest for the 'man hours

per day' objective function, the problem is considered to be a classification problem due to the discrete nature of the outputs. Hence the method to create a random forest uses classification trees as opposed to regression trees. The random forest created for the throughput objective continues to use regression trees due to the continuous nature of the output objective function.

The random forest created for the throughput objective function produced a mean square error of ~0.066 on the training data. Upon validation using a set of test data the mean square error was found to be ~0.081. This method is shown to reduce the prediction error of the original regression tree in Figure 6-15.

The random forest created for the man hours per day objective function produced an out-of-the-bag estimate of the error rate of 0%. This result echoes the results from the original regression tree validation in Table 6-12 where the percentage difference between predicted and observed values for man hours per day was 0 for all instances. This is likely to be attributed to the small number of permutations of the output variable.

Although the random forests created are capable of providing similar or better prediction accuracy, they do not necessarily provide users with a quick interpretable method for understanding trends between relationships. The random forests are also not capable of demonstrating how many observations lie within different sections of the data; this information is visible by using the regression tree method.

The random forest results produced include a more detailed breakdown of variable importance for all of the input variables. This breakdown is not available using the regression or classification tree method and is highly valuable for users to understand the entire problem.

It is proposed that the regression tree and random forest methods can be used together for the analysis of problems. In particular, the regression tree method can be used to visualise main trends and the random forest method can be used to make more accurate predictions on given data points. These methods are applied together for the assembly test case problem in the following section.

6.5 Test Case 3 – Assembly System

The regression tree and random forest methods have produced reasonable results for two generated test cases; however, they have not been tested on a real life test case. In this sub-section the regression trees and random forest methods are applied to the assembly test case which was used in Chapters 3 and 4.

This particular test case has two main conflicting objective functions and 72 input variables which relate to resource assignment for shifts and assembly tasks. This test case is much larger than the previous two test cases; hence this will enable insight into the applicability of the generated method for larger scale test cases.

Initially the regression tree modelling method from Figure 6-1 will be applied to the test case for each conflicting objective function and subsequently random forests will be created for each objective function to enable accurate prediction capabilities. The data used to generate these regression trees is the 5000 point optimal Latin hypercube design of experiments from Experiment 3 in section 5.3.3. The actual output values from the design of experiments have been non-dimensionalised.

6.5.1 Regression Tree for 'Average Assembly Time' Objective Function

An initial regression tree has been created in R for the 'average assembly time' objective function. The generated regression tree is shown in Figure 6-23. It can be seen that the regression tree has 5 levels, this is considerably more levels than the trees generated for the previous two test cases.

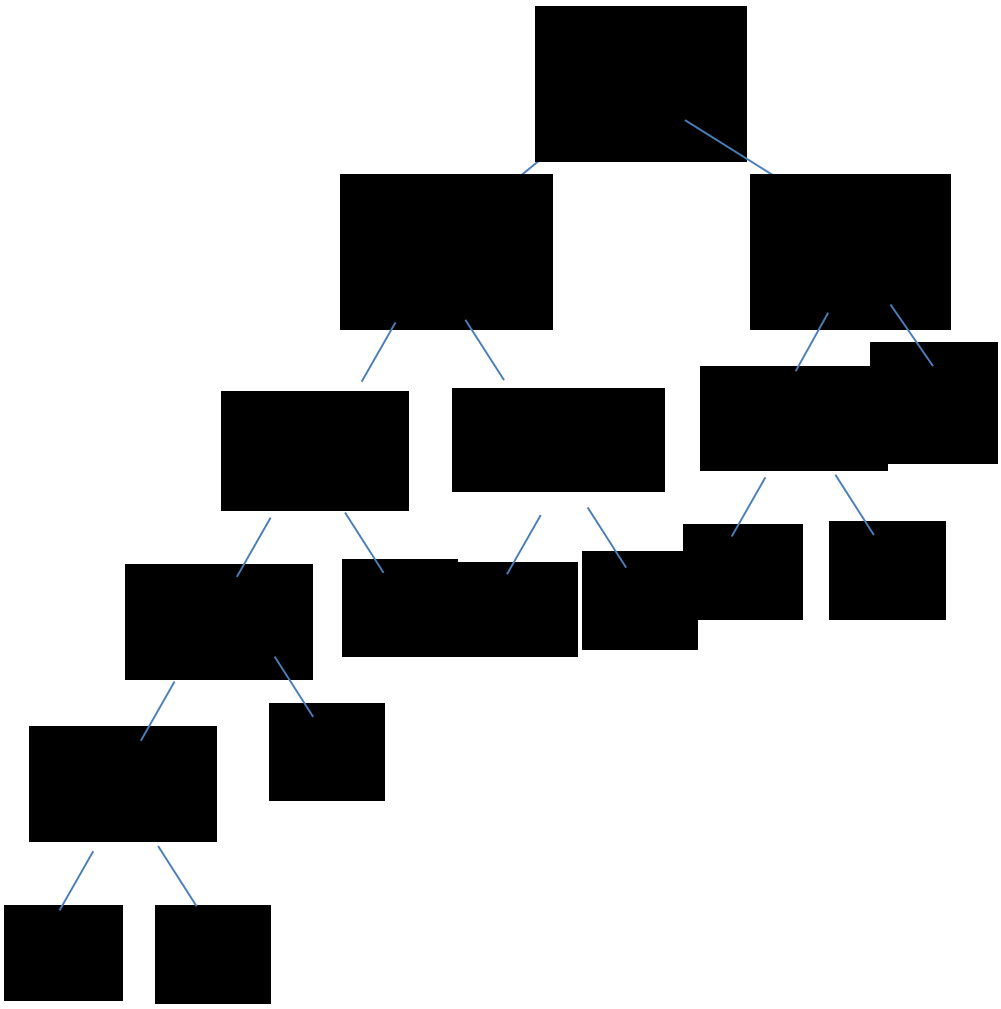


Figure 6-23 – Initial regression tree created for ‘average assembly time’ objective function

After the creation of the initial regression tree in R, pivot tables have been used to create the combined regression tree with overlaid values for the conflicting objective function. For every node or leaf of the regression tree a new pivot table is produced. The filter on the ‘row labels’ column can be used to filter values for the first node. For subsequent nodes each variable of interest can be added as a new row label to the pivot table. Then multiple selection options can be used to filter each additional variable to select or omit relevant values from the pivot table calculation.

Table 6-15 shows one of the pivot tables used in the generation of the final regression tree. In this table the first node variable is used to split the data into sub sections. Upon splitting the data, the first row results summarise the values for 'LS3_20<1.5'. Hence the values for the average of 'average assembly time' have been checked against the original tree and the average of the 'average fitters' has been overlaid on the regression tree.

Table 6-16 shows the pivot table used to calculate the values for node 5. The first variable is filtered under row labels and then additional variables can be added to the table. Rows which should not be included in the calculation have been selected and omitted. Values from the grand total row are used in node 5.

Table 6-15 – A pivot table to show calculation of values for node 7

Row Labels	Average of averageAssemblyTime	Average of averagefitters
1	0.517517856	0.442917944
2	0.382705535	0.533516341
3	0.333186424	0.60663995
Grand Total	**0.404028838**	**0.529147644**

Table 6-16 -–A pivot table to show calculation of values for node 5

Row Labels	Average of averageAssemblyTime	Average of averagefitters
2	**0.311696653**	**0.587361228**
2	**0.328755205**	**0.565286852**
2	0.344559455	0.53636329
3	0.296567215	0.624194506
3	**0.278911125**	**0.629786876**
2	0.288083979	0.612441558
3	0.262203426	0.661380133
3	**0.264799288**	**0.647980278**
2	**0.280011046**	**0.632081285**
2	0.295227469	0.615798135
3	0.24794099	0.666399571
3	**0.232325493**	**0.681921171**
2	0.249927125	0.667498789
3	0.198024876	0.710026324
Grand Total	**0.295641097**	**0.608114473**

The final regression tree is shown in Figure 6-24, all of the nodes are numbered and the values for the conflicting objective function are overlaid on the tree. The values which have been calculated in Table 6-15 and 6-16 are shown in the final regression tree.

The objective functions have been shortened to T and C for ‘average assembly time’ and ‘average fitters’ respectively to reduce space taken up on the regression tree.

The regression tree in Figure 6-24 can be used to visualise the important decision variables and the relationships between the input and output variables. The output variables have been scaled between 0 and 1 to ensure data confidentiality. The expected values for two output variables are shown in the first node and it can be seen that the expected values are both skewed in opposite directions from the midpoint 0.5.

From visualising the regression tree it can be seen that labour assignment for task 20 for each shift is the most important variable for the average assembly time. Task 7 is also shown to have a high importance. It is clear that the variables are conflicting with each other from the tree. Leaves from node 6 have lower values of T (average assembly time) and higher values of C (average fitters) and conversely the opposite side of the tree at node 7 has high values of T and low values of C.

The regression tree is also capable of showing the distribution of points within different sections of the original data. This is highly interesting as it can assist users with understanding the relationship between input and output variables.

Testing has been carried out on the regression tree to ascertain if it is capable of predicting values with reasonable accuracy. In order to carry out the testing, 4 different test conditions have been chosen based on different leaves in the regression tree. The model has then been run with the chosen test conditions leaving all other input variables the same. The regression tree predicted values have then been compared to the observed values from the testing. Results are shown in Table 6-17.

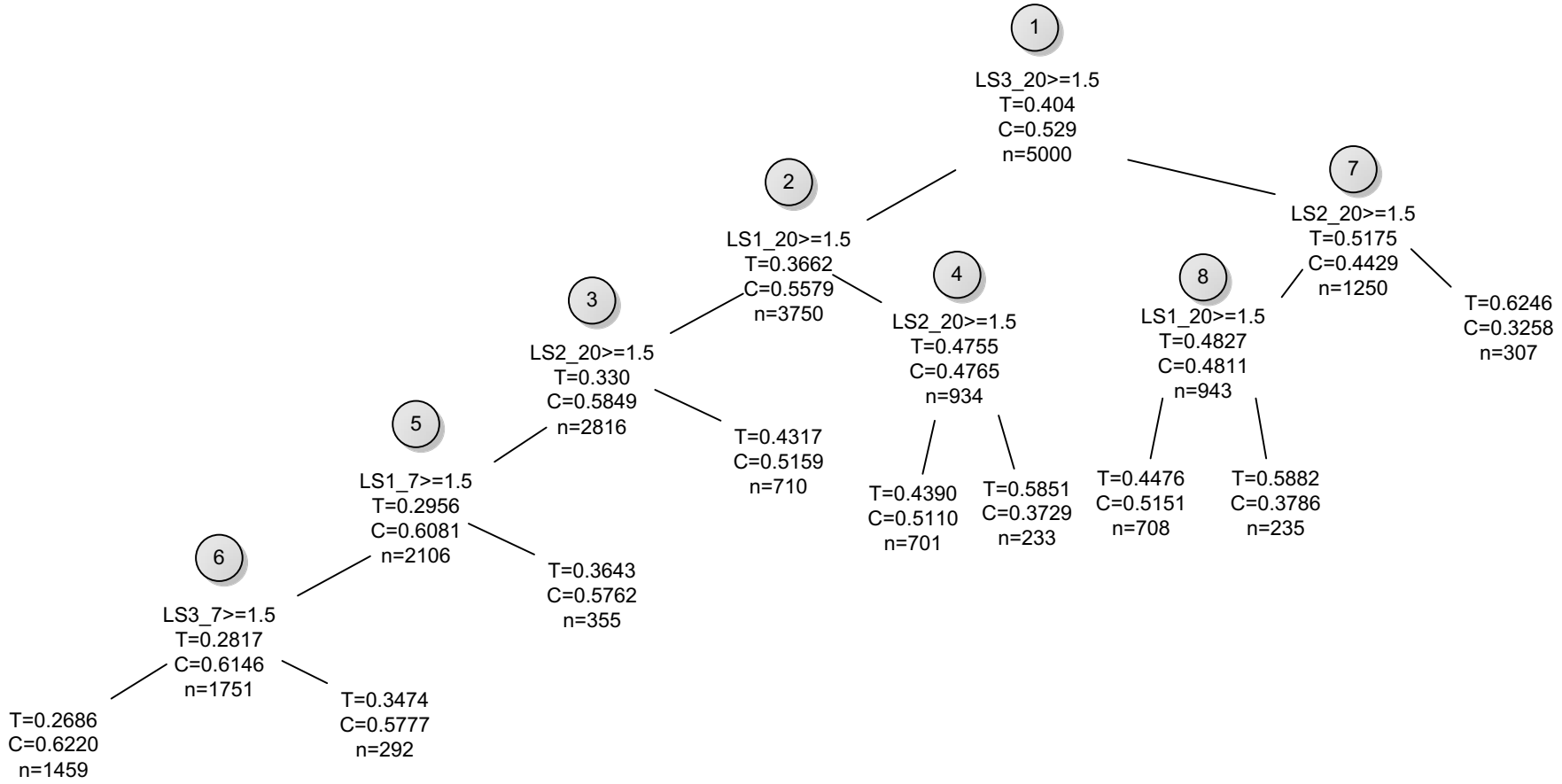


Figure 6-24 – A regression tree for the 'average assembly time' with 'average fitters' values overlaid and numbered nodes

From analysing the results in Table 6-17, it is clear that the generated regression tree is not capable of accurately predicting the output objective function values for varying input parameters. The predictions for the 'average assembly time' objective function are significantly worse than they are for the 'average fitters' objective function with 2 predictions having a percentage difference of greater than 100%.

Table 6-17 - A table to show the validation of the generated regression tree for 4 different test conditions

Test conditions	Output objective	Expected output values	Actual output values	% difference
Node 6 and LS3_7>=1.5	C	0.66220	0.6910	4.17
	T	0.2686	0.1931	-39.10
Node 4 and LS2_20<1.5	C	0.3729	0.3375	-10.49
	T	0.5851	0.5985	2.24
Node 8 and LS1_20>=1.5	C	0.5151	0.5851	11.96
	T	0.4476	0.2134	-109.75
Node 7 and LS2_20<1.5	C	0.3258	0.4710	30.83
	T	0.6246	0.2859	-118.47

This is likely to be due to the probabilistic nature of the 'average assembly time' objective function; multiple repetitions can be carried out and the average can be taken to confirm the effect of the objective function. The highest percentage difference for the average fitters objective function is 30%. These results suggest that the regression tree in this example should not be utilised for predicting new data points.

6.5.2 Regression Tree for 'Average Fitters' Objective Function

A regression tree has also been created for the 'average fitters' objective function. The generated regression tree is shown in Figure 6-25. Similarly to the tree in Figure 6-23, the regression tree has 5 levels. The main variables of interest are similar to the tree for the average assembly time objective function; however. the assignment variables for task 8 are used as opposed to task in Figure 6-23.

Subsequently, pivot tables have been used to create the combined regression tree with overlaid values for the 'average assembly time' objective function. For every node or leaf of the regression tree a new pivot table has been produced.

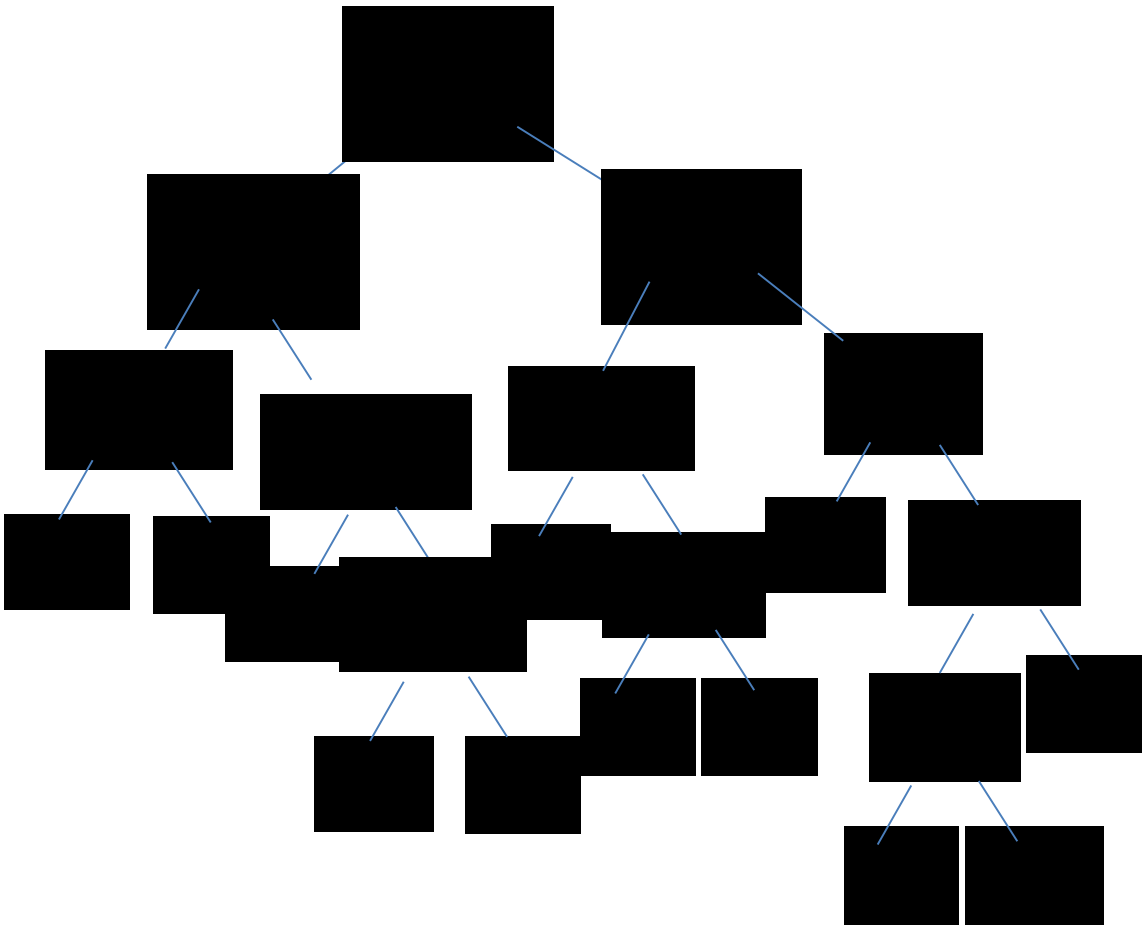


Figure 6-25 - Initial regression tree created for 'average fitters' objective function

The final regression tree for the ‘average fitters’ objective function is shown in Figure 6-26. In the figure, the values for the ‘average assembly time’ are overlaid and the nodes of the tree have been numbered. It is clear that there are more nodes in this tree than there are in the previous tree in Figure 6-24. As a result, there are more leaves within this tree leading to a deeper splitting of the data; the largest leaf in the average assembly time tree has 1459 observations however in the average fitters tree the largest leaf is 1055 observations.

The tree can be used to see the most important variables for the average fitters objective. It can be seen that the key variables are largely the same for ‘average fitters’ and ‘average assembly time’.

The regression tree can also be used to visualise the relationship between the two objective functions. The tree confirms that the objectives are conflicting as around node 3 the value for ‘average assembly time’ is high and the average fitters is low and by around node 9 this is reversed.

The regression tree has also been used to carry out testing for prediction of new data points. Four different test conditions have been chosen based on different leaves in the regression tree and the model been run with the chosen test conditions with no changes in other input variables. The regression tree predicted values have then been compared to the observed values from the testing. Results are given in Table 6-18.

From analysing the results in Table 6-18, it can be seen that in this test some predictions are highly unreliable. One of the predictions has more than 100% difference from the observed value and another prediction is almost 40% different from the observed value. All other predictions were within 20% of the observed values. These results suggest that, again, the regression tree is not capable of accurately predicting the model behaviour.

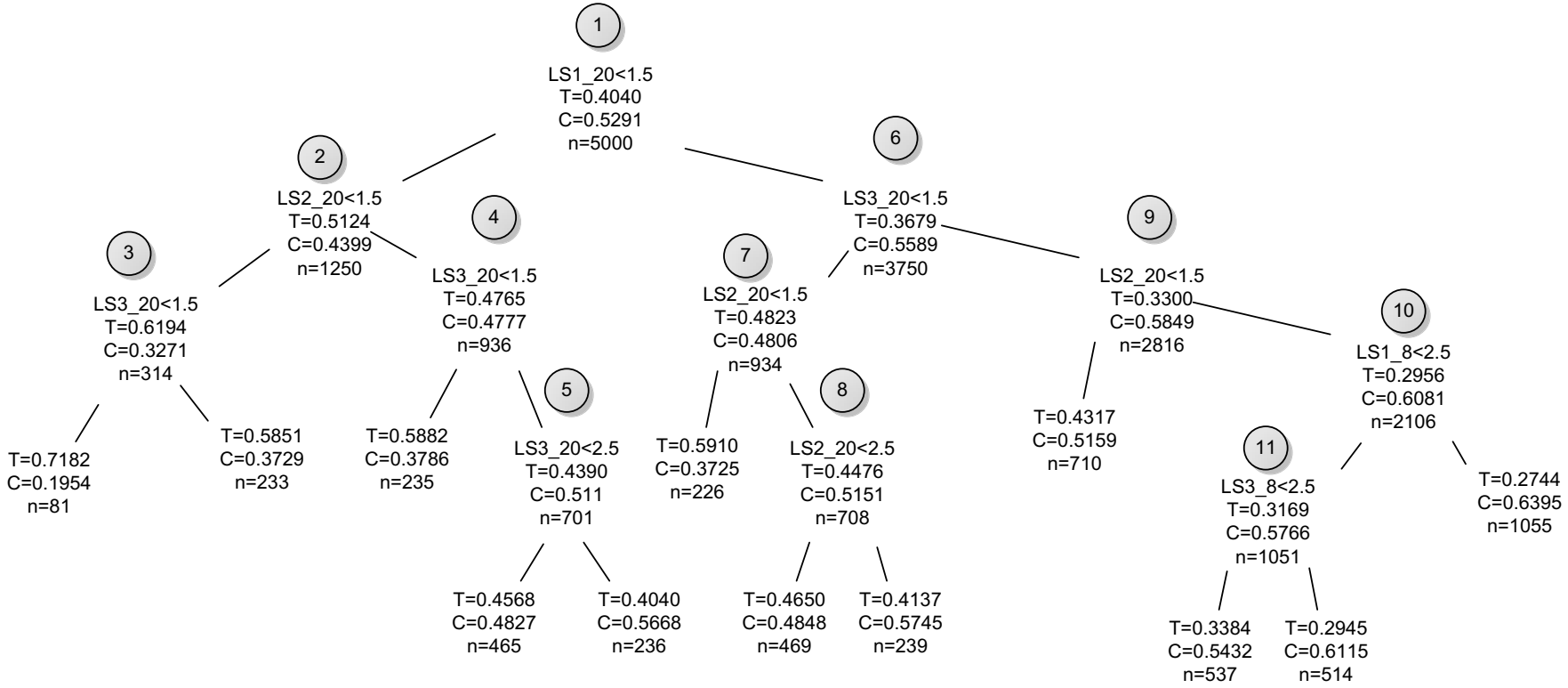


Figure 6-26 - A regression tree for average fitters with 'average assembly time' values overlaid and numbered nodes

Table 6-18 - A table to show the validation of the generated regression tree for 4 different test conditions

Test conditions	Output objective	Expected output values	Actual output values	% difference
Node 3 and LS3_20<1.5	C	0.1954	0.2396	18.45
	T	0.7182	0.6560	-9.48
Node 5 and LS3_20>2.5	C	0.5668	0.5919	4.245
	T	0.4040	0.2910	-38.835
Node 8 and LS2_20<2.5	C	0.4848	0.5851	17.14
	T	0.4650	0.2134	-117.90
Node 11 and LS3_8>2.5	C	0.6115	0.6185	1.13
	T	0.2945	0.2841	-3.66

As the results of regression trees for both objective functions for the assembly test case have demonstrated large percentage errors for predicting values, random forests will be applied for each objective function in order to improve prediction accuracy for new data points.

6.5.3 Random Forest for 'Average Assembly Time' Objective Function

A random forest has been created for the 'average assembly time' objective function using the randomForest package in R. The data for the test case has been partitioned into training and test data with the ratio of 7:3. The random

forest has been generated using training data with the default of mtry= 24, hence 24 variables are tested at each node. In addition, the random forest number of trees has been set at 5000. This value has been chosen nominally to begin with and further testing may be used to ascertain if the number of trees needs to be extended.

The generated random forest has a value for the mean of squared residuals of 0.006503 and explains 73.4% of the variation in the results. Figure 6-27 shows the convergence of the residual error as the number of trees increases. It can be seen that the value of is converged after 1000 trees. This suggests the number of trees does not need to be extended.

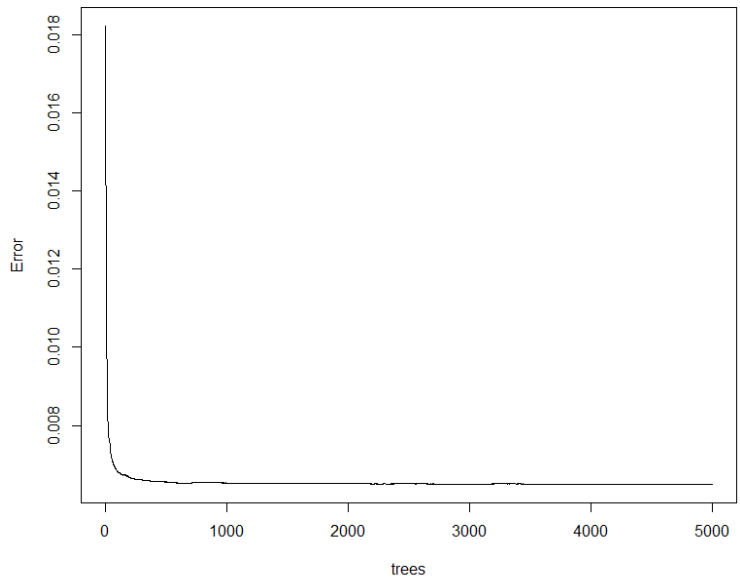


Figure 6-27 – A graph to show the residual error as the number of trees increases

Figure 6-28 shows the variable importance plots created for the random forest. The plots clearly demonstrate the 3 variables relating to task 20 being of highest importance.

The test data has been used to carry out predictions using the random forest results. The predicted against observed values are plotted in Figure 6-29 with the $y = x$ line. From the results it can be seen that the results fall primarily

within the region of the line, however there is a general trend of underestimating the lower values and overestimating higher values of the ‘average assembly time’ objective function.

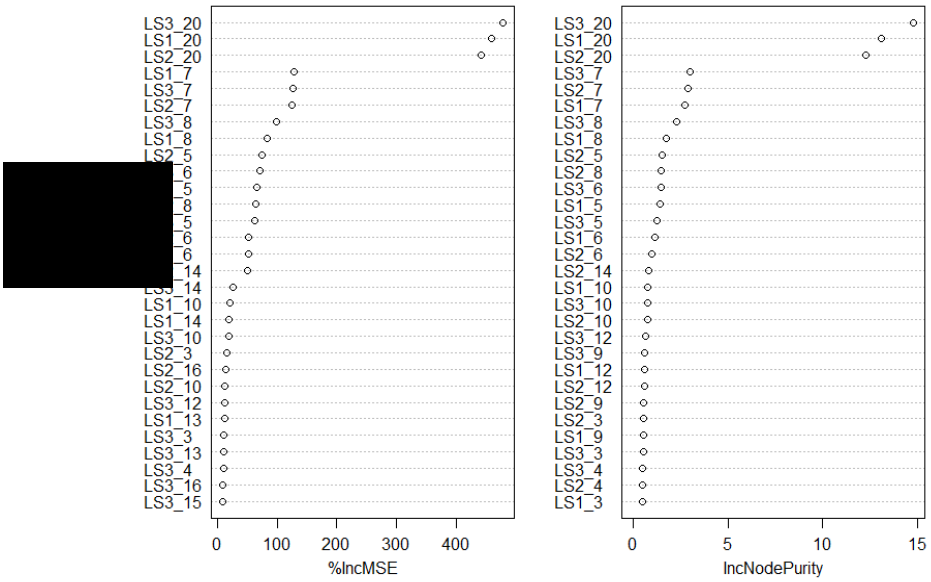


Figure 6-28 – Graphs to show the variable importance using % increase in MSE and increase in node purity

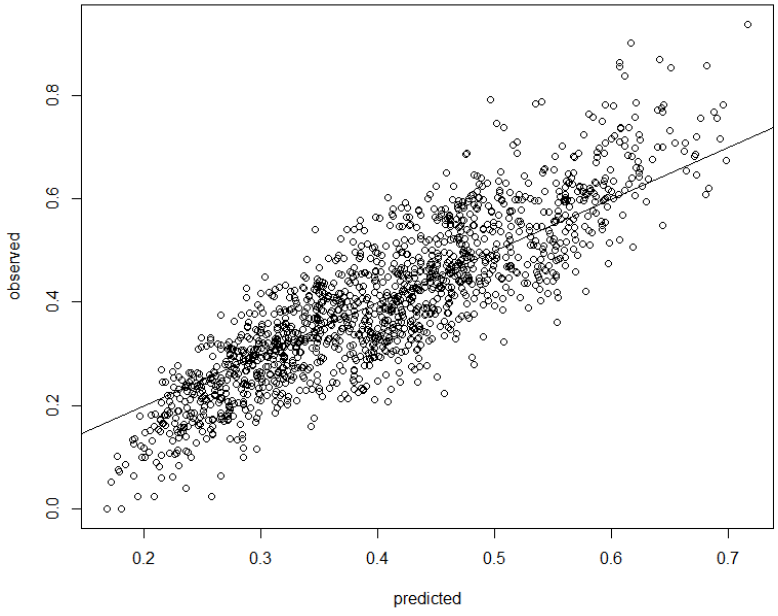


Figure 6-29 – A graph to show the observed against predicted values for the test data for ‘average assembly time’ objective function

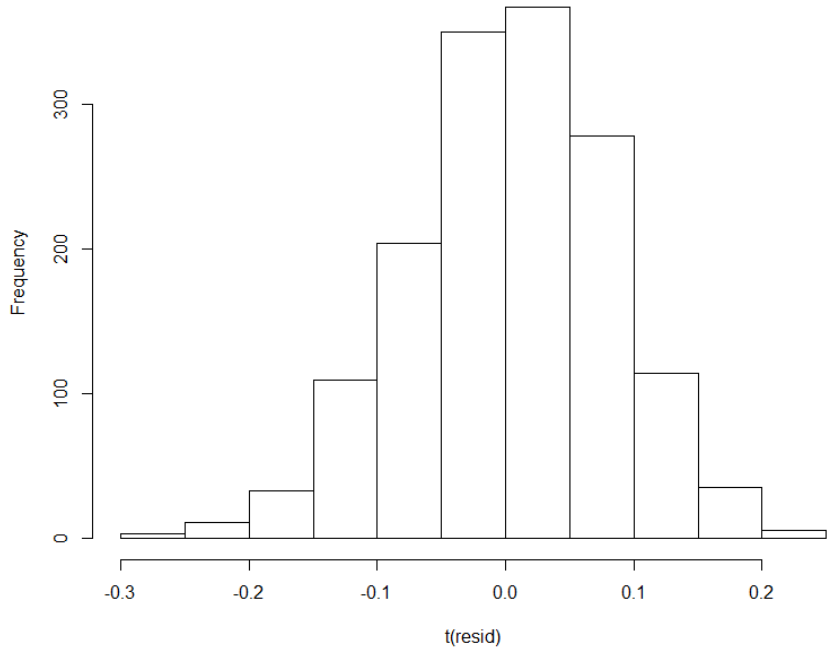


Figure 6-30 – A histogram of the residuals for predictions using the random forest on the 'average assembly time' objective function

Amongst the results the largest residual was found to be -0.30. The standard deviation of the residuals was found to be 0.0789 and the mean value of the residuals was 0.003076. In addition, a histogram of the residuals has been created as shown in Figure 6-30. The histogram reasonably represents a normal distribution and 67.992% of the results were found to fall within 1 standard deviation of the mean, hence the random forest is deemed to be reasonable to use for predicting new data points.

6.5.4 Random Forest for 'Average Fitters' Objective Function

A random forest has been created for the 'average fitters' objective function. Identical training and test data has been used to the random forest calculations for the 'average assembly time' objective function. The random forest has been created using 5000 trees and the default value of 24 variables tested at each node.

The generated random forest produced a mean of squared residuals of 0.006974 and the forest accounted for 69.24% of the variation in results.

Figure 6-31 shows the convergence of the residual error as the number of trees increases. It can be seen that after around 1000 trees the error is converged, hence no further testing is required with larger values of n.

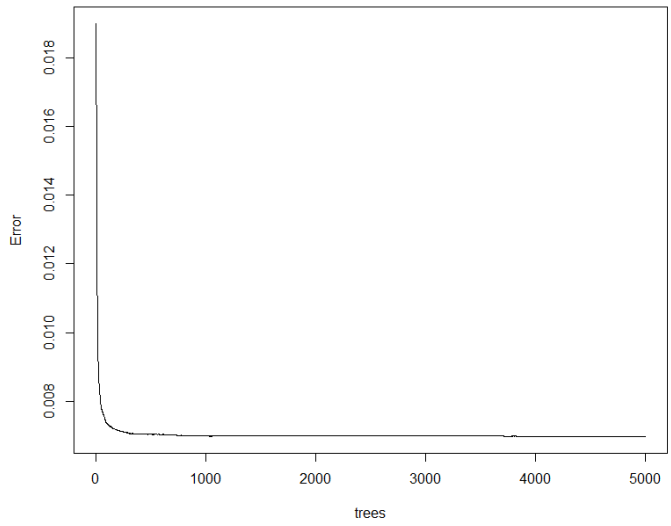


Figure 6-31 – A graph to show the convergence of residual error as the number of trees increases

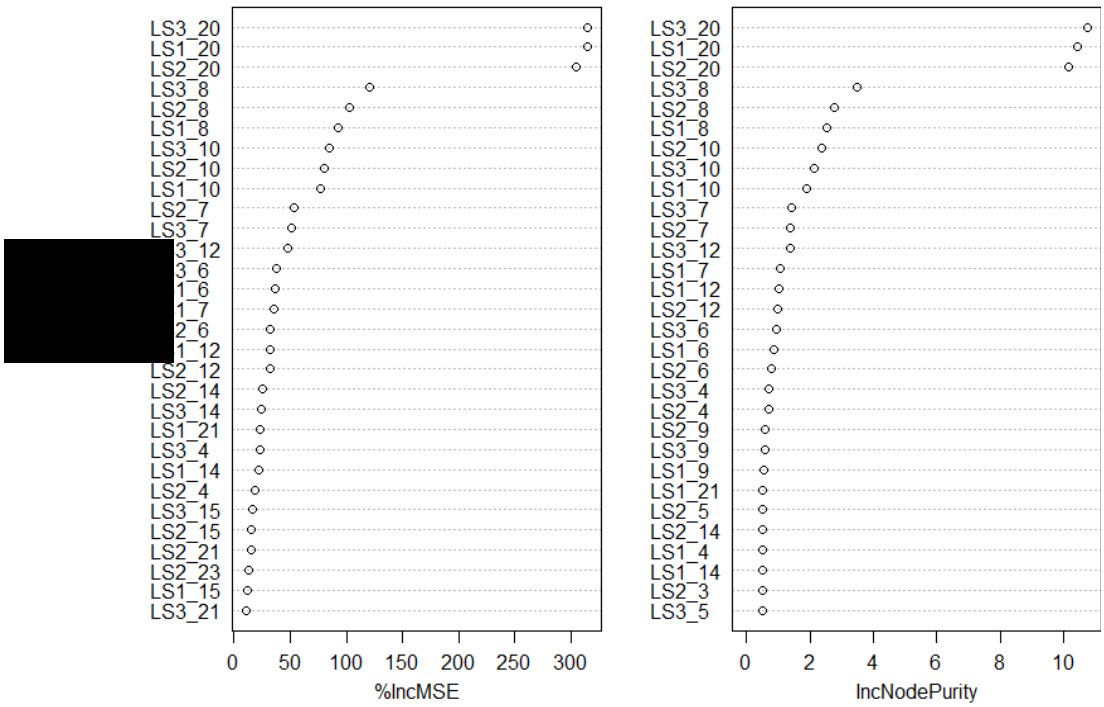


Figure 6-32 – Graphs to show the % increase in MSE and increase in node purity for the average fitters objective function

Figure 6-32 shows the variable importance plots for the random forest results. It can be seen that the task 20 has the greatest importance for the average fitter values; this is a similar result to the ‘’average assembly time’ random forest results. The variable importance plot has been compared against the cycle times for each of the tasks. The 5 tasks with the highest cycle times for the assembly test case are (in order): 20, 8, 10, 6 and 7. From the variable importance plots it can be seen that the first three variables with the highest cycle times are identified as being the most important. This suggests that the random forest is capable of accurately identifying the most vital variables.

The random forest has been used to make predictions for the test data. Figure 6-33 shows the predicted against observed results for the predictions, the $y = x$ line has also been plotted on the graph. The graph shows that the results predominantly lie within the region close to the line; however, there is slight underestimation of lower values and overestimation of higher values. The standard deviation of the residuals was found to be 0.0801 and the average of residuals was -0.000517.

In addition, 68.32% of the results were found to fall within 1 standard deviation of the mean, hence the random forest has been deemed reasonable for predicting new data points.

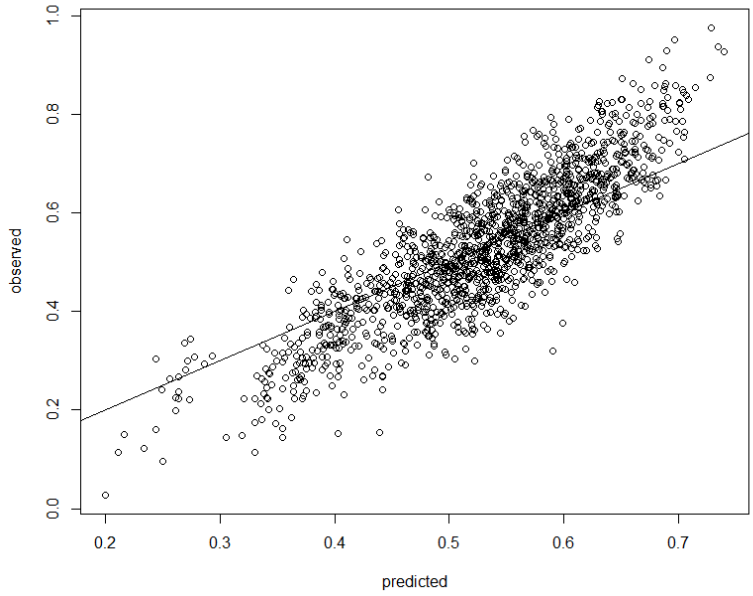


Figure 6-33 – A graph to show the observed against predicted values for the 'average fitters' objective function

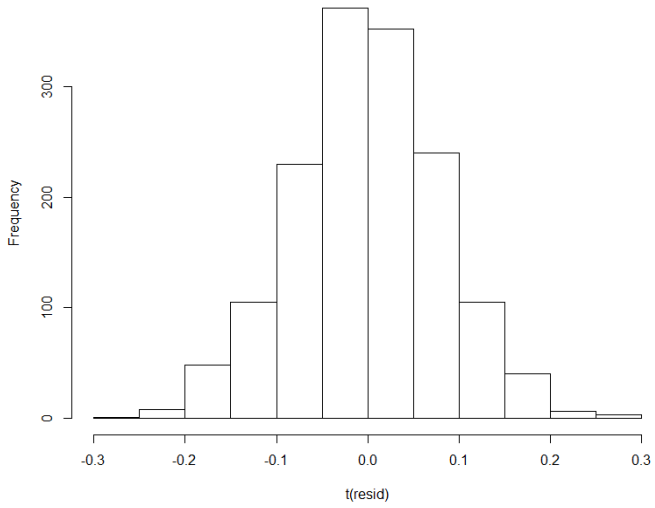


Figure 6-34 - A histogram of the residuals for predictions using the random forest on the 'average fitters' objective function

6.5.5 Summary of Results from Test Case 3

A number of interesting observations have been made through the application of regression trees and random forests to the assembly test case for both conflicting objective functions.

Figures 6-24 and 6-26 show the generated regression trees using the methodology which has been outlined in Figure 6-1. The regression trees are capable of demonstrating the general trends for both objective functions across the various nodes within the trees. This information is clearly visible within the graphs and users can also visualise the distribution of points amongst the tree nodes and leaves. This breakdown of information can be highly valuable for decision makers as it can assist them in visualising complex interactions between input and outputs.

A main drawback of the regression tree is found to be the prediction capabilities of the tree. For the previous test cases the largest percentage differences were found to be around 20%, however for the assembly test case a few predictions had percentage differences of over 100%. This method does not provide sufficient accuracy to enable users to predict outcomes from given data points. In order to alleviate this problem, random forests have been created for each of the conflicting objective functions. The random forests have been generated with training data and a separate set of test data points has been used to carry out predictions. For both of the generated random forests, although there is some correlation between the predicted value and the residuals, the predictions have largely fallen within 1 standard deviation of the mean. For both random forests the average of the residuals was found to be within 1% of 0 and the majority of results have been found to fall within 1 standard deviation of the mean. Hence both random forests have been deemed to be reasonable to use for predicting new data points. The random forest was also found to be a quicker method for the creation of prediction for new data points.

In future research, the regression trees could also be overlaid with information such as confidence intervals to assist users in understanding the range of expected values and assist users in understanding robustness of their solutions.

6.6 Discussion of Results

The chapter has presented a framework using regression trees for visualising the relationships between multiple KPIs and decision variables. In addition, random forests are proposed to increase the prediction accuracy of new data points compared to regression trees. Two different case studies have been generated to enable framework demonstration and validation. The assembly test case from Chapter 3 has also been used to validate the framework.

The first case study is a small scale test case of a manufacturing facility which includes cranes, variable labour, three workstations, two different part types and a combination of automated and manual operations. The second case study is a medium size test case of eight workstations and variable buffer sizes, variable labour pool and variable inter-arrival times. In addition, the framework is demonstrated and validated using the assembly test case as it is a larger test case of a real-life system.

From the results in the first case study it was found that output variables with a correlation coefficient of -1 had regression trees which are mirror images of each other; however, regression trees for pairs of variables with other correlation coefficients varied considerably. A regression tree with overlaid values for an alternative output variable has been generated; the tree has been used to carry out prediction for new data points. The predictions were found to fall within 6% of the actual values.

The second case study has been used to generate a regression tree which demonstrates the relationship between input variables and two conflicting objectives. The regression tree is capable of demonstrating the general trend in a simple, easy to interpret manner. However, when new data points were tested with the regression tree, the predictions were found to have a percentage difference of up to 20% from the actual values. Hence random forests have been proposed to assist with creating higher accuracy predictions.

For the second test case two random forests have been generated. The random forest created for the throughput objective used regression trees due to the

continuous nature of the output objective function. The random forest produced a mean square error of ~0.066 on the training data. Upon validation using a set of test data the mean square error was found to be ~0.081. This method is shown to reduce the prediction error of the original regression tree. A classification random forest for the ‘man hours per day’ objective function has been generated. The problem is considered to be a classification problem due to the discrete nature of the outputs. This random forest produced an out-of-the-bag estimate of the error rate of 0%. This result echoes the results from the original regression tree validation.

For the assembly test case, the regression trees are capable of demonstrating the general trends for both objective functions across the various nodes within the trees. This information is clearly visible within the graphs and users can also visualise the distribution of points amongst the tree nodes and leaves. A main drawback of the regression tree is found to be the prediction capabilities of the tree. For the previous two case studies the largest percentage differences were found to be around 20%, however for the assembly test case a few predictions had percentage differences of over 100%. Hence random forests have been generated for the objective functions. Both generated random forests had some correlation between the predicted value and the residuals. However, for both random forests, the average of the residuals was found to be within 1% of 0 and the majority of results have been found to fall within 1 standard deviation of the mean. Hence both random forests have been deemed to be reasonable to use for predicting new data points. The random forest was also found to be a quicker method for the creation of prediction for new data points.

In summary, regression trees have been demonstrated to provide users with a quick interpretable method for understanding trends and relationships between input and output variables. This breakdown of information can be highly valuable for decision makers as it can assist them in visualising complex interactions between input and outputs. Regression trees are also capable of demonstrating how many observations lie within different sections of the data; this information is useful to enable users to see the distribution of observations

under varying conditions. Random forests have been proposed as they are capable of providing similar or better prediction accuracy. Random forests have been used to produce a detailed breakdown of variable importance for input variables; this is not available using the regression or classification tree method and is highly valuable for users to understand the entire problem.

It is proposed that the regression tree and random forest methods can be used together for the analysis of problems. In particular, the regression tree method can be used to visualise main trends and the random forest method can be used to make more accurate predictions on given data points.

In future work, confidence intervals can be overlaid onto the nodes within the regression trees to enable users to understand the level of certainty of output values for the tree nodes and leaves.

6.7 Chapter Summary

This chapter has presented a framework to enable DES users to visualise key relationships between input and output variables for manufacturing process optimisation problems. The framework includes the use of regression trees for visualising the impact of decisions on multiple output variables and the use of random forests to enable prediction for new data points.

The framework has been demonstrated on two different generated case studies. Regression trees for output variables with a correlation coefficient of -1 are shown to be mirror images of each other. The generated regression trees are shown to provide simple visualisation of trends and relationships among the input and output variables. The random forests are shown to produce a detailed breakdown of variable importance for input variables. Random forests are also shown to provide similar or better prediction for output variable values for new data points.

In addition, the framework has been demonstrated on the assembly test case, as introduced in Chapter 3. The regression trees demonstrate the main trends within the data and the random forest is found to predict the key variables within the real life assembly test case to a reasonable level of accuracy.

CHAPTER 7 - DISCUSSION AND CONCLUSIONS

This chapter outlines the key findings from this research. A number of areas are discussed including: research trends and gaps, observations from test cases, novel contributions to knowledge and research limitations. The conclusions are provided thereafter.

7.1 Literature Review

The literature review has been carried out in two main sections covering: assembly optimisation using DES and visualisation of manufacturing processes. The key observations, research gaps and trends are summarised in the next subsections.

7.1.1 Observations

The literature review which has been carried out has generated a range of interesting insights. This section summarises the key observations.

The first section of the literature review has covered DES modelling and optimisation of assembly systems. The main application domains for DES and optimisation of assembly systems were systems with high variability. These applications are mostly automotive, electronics and flexible manufacturing systems. These applications predominantly aim to optimise time based objective functions such as lead time and throughput; the systems are mostly modelled in commercial software packages. Such applications largely use what-if scenario analysis to optimise with respect to their objective functions, although there is a growing use of Artificial Intelligence methods to optimise these objectives. The high computational effort required to optimise these DES

models has been identified amongst application papers. To deal with this, a number of papers have proposed analytical models as an alternative to DES models and compared the results between both models. Hybrid modelling methods have been used to generate models including heuristic rules, analytical approaches and visualisation. Multiple objectives are often combined into a single objective function to optimise simultaneously; however, these studies have not explicitly applied multi-objective optimisation algorithms, possibly due to the high number of evaluations required for convergence.

The second part of the literature review covers the visualisation of manufacturing processes with simulation. This is noted to be a relatively new field of research; hence there is not a large amount of literature in this area. It is noted that visualisation and DES can be used to detect collisions for construction of assemblies. Virtual commissioning can be used for modelling facilities and this can be used to reduce issues when the system is implemented. In addition, it has been identified that the generation of animation for worker kinematics is highly time consuming.

7.1.2 Research Trends and Research Gaps

A number of key research trends and research gaps have been identified through the literature review. This subsection outlines the key trends and research gaps, these areas have been used to generate the aim and objectives for this research.

The trends which have been identified are as follows:

- There has been a growing number of papers in the area of DES optimisation of assembly systems
- Application domains of DES optimisation for assembly systems are predominantly assembly lines with high levels of variability
- An increasing number of papers are using hybrid modelling methods

- Various authors have utilised analytical models to compare to DES models, in order to attempt to reduce the computational time associated with DES

- The majority of papers studied use what-if scenario analysis to assist with optimising models and decision making for configurations
- Within recent years, the use of Artificial Intelligence methods to optimise assembly systems has increased
- Multiple objectives have been considered within studies, however these objectives are often aggregated into one objective function.
- The use of immersive visualisation for DES models is a relatively new area of research which is growing.

The research gaps are summarised as follows:

- Although many authors have linked DES models with multi objective optimisation algorithms for a range of specific test cases, the literature does not contain discussion and explanation on the process of creating models of assembly systems which are ready-to-optimise. There are a few challenges associated with creating a generic process to create ready-to-optimise DES models which could be explored.

- The literature shows a growing interest in hybrid optimisation and AI methods. There is also growing interest in the use of multi-objective optimisation; many authors aggregate their multiple objectives into a single objective. A few authors explicitly apply multi objective optimisation. However, there is no literature to relieve the computational cost of running multi objective optimisation on DES models of assembly systems.

- Within the area of immersive visualisation, there is a growing interest in the use of DES with immersive visualisation. Although there is research in the area of using immersive visualisation and DES for layout planning; there is a lack of research in scenario analysis from within the 3D

environment. Creating a closed feedback loop to visualise and modify DES models within an immersive environment can assist decision makers.

- The literature search has revealed papers which perform optimisation on DES models; however, after the DES results are obtained there is a lack of research into use of data mining techniques to assist users with understanding insights from the results. In particular decision trees can be used to link input and output variables from DES models visually and assist decision making.

7.2 Development of ready-to-optimise DES models

Chapter 3 presents a framework to enable the development of DES models which are ready-to-optimise. The framework includes selecting input and output variables, generating interfaces for input and output variables, selection of simulation run parameters, linking the model to the optimiser and adjustments to the model after initial testing. The framework is then applied to create two DES models for two different real life test cases: the turbine assembly test case and a repair facility test case. Through the development of both simulation models and optimisation links, a variety of important observations have been made regarding both test cases.

Both models demonstrated the requirement to determine a compromise between computation time and convergence of results. The repair facility model was found to converge to stable results faster than the assembly test case. This is expected to be due to the amount of variability within the assembly test case, as there are no probabilistic parameters within the repair facility test case.

For both test cases, the simulation run parameters must be selected prior to optimisation; namely simulation run length, simulation run window and number of replications. For the repair facility test case, as there were no probabilistic parameters, hence no replications were required. For the assembly test case,

an interdependency was identified between the input and output parameters. This interdependency led to skewed results when the optimiser was applied; hence the formulation of this variable had to be modified.

It was found that, for the repair facility test case, layout optimisation was not possible by simply applying an optimiser to the locations of workstations; this would require the development of geometrical constraints. Hence optimisation of a set of predefined layout solutions was proposed. For the resource allocation optimisation, an optimisation algorithm can be directly applied to the model without addition of further constraints.

For both test cases, the number of potential solutions had to be reduced as much as possible in order to relieve the computational cost of running the simulation models.

7.3 Framework for Multi-Objective Optimisation of DES Models

A framework is proposed in Chapter 4 to enable multi-objective optimisation of DES models for an assembly test case. This framework development directly addresses the research gap in multi-objective optimisation of DES models of assembly processes. The developed framework includes the generation of regression models for multiple objectives and simultaneously optimising each regression model using multi-objective optimisation algorithms, such as NSGAII. This is also a novel application area as it extends work which was previously carried out in McHaney & Douglas (1997) by creating meta-models for multiple objective functions.

The proposed framework is demonstrated on three different experiments with varying simulation run parameters and with an increase in the number of stochastic variables in the model formulation for the final experiment.

The framework involves the use of design of experiments, the generation of two meta-models, multi-objective optimisation of the meta-models and re-running the Pareto front points with the actual DES model. The first experiment was

found to lead to a large visible difference between the predicted and actual results; hence a second experiment was carried out with a longer run length for each observation. The results show visible improvement in the distance between predicted and actual objective values. A last experiment was carried out to assess the effect of adding probabilistic cycle times to the simulation model; results showed a number of overlapping points in the objective space. Upon further inspection it was found that these points did not reflect identical designs.

In general, the experimental results showed that the ‘average assembly time’ meta-models for experiments 1 and 3 were not able to predict the actual simulation results to a reasonable level of accuracy. The meta-models for ‘average fitters’ had low residual standard error values for all experiments, suggesting a good fit. This is expected as the ‘average fitters’ objective is essentially a time weighted average of the input variables. The meta-models for the ‘average fitters’ objective account for a higher proportion of variability than for the ‘average assembly time’ objective.

The presented optimisation methodology can be used in place of multi-objective optimisation on the DES model, in order to reduce computation time required for optimisation. It is proposed that this method will enable users to drive towards the Pareto front region of the objective space with significantly less computation time than explicit multi-objective optimisation on the DES model. It is recommended that users subsequently choose single solutions and explore the nearby regions using single objective optimisation methods.

It is noted that interpreting the results from the developed framework may be difficult for manufacturing and assembly experts. The Pareto front clearly demonstrates the trade-off between the two objectives; however, it does not assist with choosing the final optimal results and corresponding decision variable values. The framework also does not assist users with understanding the relationship between input and output variables; hence another framework has been developed in Chapter 6 to enable this.

7.4 Framework for Interactive Immersive Visualisation of DES Models

Chapter 5 presents a framework to enable interactive immersive visualisation of 3D DES models. The framework includes a DES model, an immersive visualisation environment and a prototype user interface. The prototype user interface has two-way interaction capabilities with the DES model. The DES model has capability to create and interact with the immersive visualisation environment. The prototype user interface enables users to change DES model configurations and visualise the impact of changes on the model variables. The framework has been developed for the repair facility test case, however many of the features will be universally applicable to other DES models such as the run toolbar and utilisation statistics. For the repair facility test case, the production managers are particularly interested in reconfiguring the layout for future repair requirements.

Meetings with production managers are used to generate requirements for an interactive interface for the repair facility test case; the requirements are split into three categories: visualisation, interaction and specific tasks. The user interface has been developed in Matlab as a prototype; at a later stage this can be incorporated into the main immersive environment.

It was found that a number of specific functions had to be added within the original DES model to enable features within the developed user interface. This additional functionality conflicts with the requirement to streamline the model for immersive visualisation, the resulting model was found to be slower to run.

The developed framework has been validated using two experiments; the experiments demonstrate the two-way interaction between the interface and the simulation model. The two experiments include a layout configuration experiment and a labour utilisation experiment.

A number of observations have been made as a result of the two experiments. For the factory layout reconfiguration experiment, a number of inconsistencies were found to occur in the immersive environment when the factory layout was

reconfigured, for instance the locations of the buffers were found to be incorrect which could lead to collisions. For the labour utilisation test case, it was found that the labour utilisation adjustment did not have a clear impact on the 3D immersive model. The results have been clearly reflected in the user interface; however, there is no available method to visualise which workstations have been impacted by the change.

7.5 Framework for Visualisation and Decision Making using DES Models

Chapter 6 presents a framework for visualising interactions between variables within DES models to assist with decision making. The presented framework uses regression trees for multiple key performance indicators to visualise trends within the data. Random forests are also proposed to improve the prediction accuracy for new scenarios compared to regression trees.

The developed framework is validated using three different case studies. The first two case studies are generated DES problems; the third case study is the turbine assembly test case as introduced in Chapter 3.

The first case study is a small scale test case of a manufacturing facility. For this case study, it was found that output variables with a correlation coefficient of -1 had regression trees which are mirror images of each other; however regression trees for pairs of variables with other correlation coefficients varied considerably. A regression tree with overlaid values for an alternative output variable has been generated; the tree has been used to carry out prediction for new data points. The predictions were found to fall within 6% of the actual values. This was deemed to be an acceptable result.

The second case study is a medium size test case with eight workstations and two conflicting objective functions. The regression tree is capable of demonstrating the general trend in a simple, easy to interpret manner. However, when new data points were tested with the regression tree, the predictions were found to have a percentage difference of up to 20% from the actual values.

Hence random forests have been applied. The random forest produced a mean square error of ~0.066 on the training data. Upon validation using a set of test data the mean square error was found to be ~0.081. This method is shown to reduce the prediction error compared to the original regression tree and provide accurate prediction capabilities.

In addition, the framework is demonstrated and validated using the assembly test case as it is a larger test case of a real life system. The regression trees are capable of demonstrating the general trends for both objective functions across the various nodes within the trees. Again, the main drawback of the regression tree is found to be the prediction capabilities, a few predictions had percentage differences of over 100%. Hence random forests have been generated for the objective functions. The random forest for the ‘average assembly time’ objective function produced a mean square error of 0.006503 and the model was found to explain 73.4% of the variation of the results; hence the model has been deemed reasonable for predicting new data points. The random forest for the ‘average fitters’ objective function had a mean square error of 0.006974 and was found to account for 69.24% of the variation in the results, hence the random forest is found to be reasonable for predicting new data points. The random forests were also found to be a quicker method for the predicting new data points.

In summary, regression trees have been demonstrated to provide users with a quick interpretable method for understanding trends and relationships between input and output variables. Random forests have been proposed as they are capable of providing similar or better prediction accuracy. Random forests have also been used to produce a detailed breakdown of importance for different input variables.

It is proposed that the regression tree and random forest methods can be used together for the analysis of problems. In particular, the regression tree method can be used to visualise main trends and the random forest method can be used to make more accurate predictions on given data points.

7.6 Research Limitations

A variety of research limitations have been identified through the research. The limitations are summarised as follows.

The requirements for ready-to-optimise DES models have been generalised based on two different examples: the repair facility test case and the turbine assembly test case. These two particular test cases have specific requirements. The repair facility test case is a layout optimisation problem and the turbine assembly test case is a resource allocation problem. It is expected that for other types of problems, there may be a requirement for specific modifications to enable a link with optimisation algorithms. For instance, a scheduling optimisation problem may require a method to enable perturbation in the order of parts.

The framework for multi-objective optimisation is only tested on the turbine assembly test case. This test case has two output objectives which can be modelled as linear regression models. This is due to the proportional reduction in time required when labourers are added onto the assembly tasks. This framework should also be tested by replacing the regression models with other meta-modelling methods such as random forests to deal with non-linear relationships.

The interactive immersive framework developed is specifically for layout optimisation. Hence, only features which are relevant to this problem have been built into the framework and tested. These features include layout reconfiguration capabilities and labour modification. Some of the functionalities demonstrated can be used for general immersive DES models, however further test cases should be developed to enable a range of key functionalities to be identified.

The regression tree visualisation can only be used to visualise trends if they are linear. The regression tree will not be able to assist users to identify non-linear relationships between input and output variables. The generation of a single regression tree is highly dependent on the location of the first binary split due to

the sequential nature of the algorithm. Hence, random forests have been proposed to enable more accurate prediction capabilities. It is expected that for cases where the regression trees are not capable of providing reasonable results due to non-linearity, random forests can still be used.

7.7 Future Research

A number of key areas for research in the future are proposed as follows:

- The multi-objective optimisation framework presented in Chapter 4 can be augmented with other meta-modelling methods to improve prediction accuracy. These methods could include Artificial Neural Networks or Random Forests.
- The interactive immersive visualisation framework could be further developed by applying geometrical constraints to enable a direct link between the model and an optimisation algorithm.
- The 3D interactive immersive visualisation functionality could be extended to include an indication of the workstation status, in a similar manner to 2D DES models. This will enable a visual indication of results of the reduction in labour in the 3D immersive visualisation. This could be carried out using 3D coloured lighting.
- For the immersive interaction framework, further testing should be carried out with workstations as the union of the workstation and the associated input and output buffers in order to resolve any issues with buffers clashing.
- For the immersive interactive framework, the interface can be extended to include options for sections for visualisation allowing users to choose appropriate combinations of information to stop overload of information.
- The visualisation framework for DES results could be tested with non-linear functions. In addition, confidence intervals can be overlaid onto

the nodes within the regression trees to enable users to understand the level of uncertainty of output values.

7.8 Conclusions

A number of key findings have been generated from this work; these are listed as follows:

- It is possible to create an optimisation ready DES model for resource allocation optimisation and labour utilisation optimisation. However, there may be unique requirements for constraints for layout optimisation and other more complex optimisation problems.
- Multi-objective optimisation is possible on meta-models of a DES model. The prediction capabilities of labour allocation meta-models are reasonably accurate, however the meta-models for average lead time do not predict the simulation model well due to the stochastic nature of the underlying simulation model parameters.
- It is possible to generate an immersive interactive interface with a DES model to assist with layout planning. There are many features of 2D DES models which could be built into the interface to assist users with decision making. However, often there are conflicting requirements for a 3D immersive model and a 2D DES model. For instance, the 3D immersive model requires smooth visualisation, hence many features of the 2D model were removed to streamline the model.
- Regression trees can be used to assist users with visualising the relationships between input and output variables for DES models. Multiple output values can be overlaid on the nodes of the regression trees to visualise the impact of decisions on multiple objectives. Random forests can be used to improve the prediction accuracy for new data points.

REFERENCES

Aguirre, A., Müller, E., Seffino, S., & Méndez, C. A. (2008). Applying a simulation-based tool to productivity management in an automotive-parts industry. *Winter Simulation Conference* (WSC '08), December 7-10, Miami, Florida, 1838–1846.

Alrabghi, A., & Tiwari, A. (2015). State of the art in simulation-based optimisation for maintenance systems. *Computers and Industrial Engineering,* 82*,* 167-182.

Asil Bulgak, A. (1992). Impact of quality improvement on optimal buffer designs and productivity in automatic assembly systems. *Journal of Manufacturing Systems*, *11*(2), 124–136.

Babiceanu, R.F. & Seker, R. (2015) Manufacturing operations, internet of things, and big data: towards predictive manufacturing systems, in: T. Borangiu, A. Thomas, D. Trentesaux (Eds.), Service Orientation in Holonic and Multi-agent Manufacturing, 594, Springer, Berlin, Germany, pp. 157–164

Baines, T., Hadfield, L., Mason, S., & Ladbrook, J. (2003). Using empirical evidence of variations in worker performance to extend the capabilities of discrete event simulations in manufacturing. *Winter Simulation Conference* (WSC '03), 7-10 December, New Orleans, Louisiana, 1210–1216.

Behdad, S., Berg, L. P., Thurston, D., & Vance, J. (2014). Leveraging virtual reality experiences with mixed-integer nonlinear programming visualization of disassembly sequence planning under uncertainty. *Journal of Mechanical Design, Transactions of the ASME*, *136*(4).

Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653–1669.

Blanchet, M. Rinn, T., von Thaden, G. de Thieulloy, G. (2014) “Industry 4.0: The New Industrial Revolution. How Europe will succeed.”, Roland Berger Strategy Consultants GmbH.

Brennan, R. W., & Rogers, P. (1995). Stochastic optimization applied to a manufacturing system operation problem. *Winter Simulation Conference* (WSC'95), 3-6 December, Arlington, Virginia, 857–864.

Bulgak, A. A., & Sanders, J. L. (1991). Modeling and design optimization of asynchronous flexible assembly systems with statistical process control and repair. *International Journal of Flexible Manufacturing Systems*, *3*(3-4), 251–274.

Camci, F. (2009). Comparison of genetic and binary particle swarm optimization algorithms on system maintenance scheduling using prognostics information. *Engineering Optimization*, *41*(2), 119–136.

Cao, J., Bell, K. R. W., Kockar, I., & San Martin, L. A. S. (2010). Generation maintenance scheduling in a liberalised electricity market. *Universities Power Engineering Conference* (UPEC 2010), 31 August - 3 September, Cardiff, U.K.

Cave, A. P., Nahavandi, S., & Creighton, D. (2006). Benchmark generation algorithm for stochastic mixed model assembly shop simulation and optimization. *International Journal of Production Research*, *44*(6), 1193–1216.

Chandra, V., Huang, Z., & Kumar, R. (2003). Automated control synthesis for an assembly line using discrete event system control theory. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *33*(2), 284–289.

Choi, S., Cheung, H., (2008), A versatile virtual prototyping system for rapid

product development, *Computers in Industry, 59*(1)*,* 447-488

Cochran, J. K., & Kaylani, H. A. (2008). Optimal design of a hybrid push/pull serial manufacturing system with multiple part types. *International Journal of Production Research*, *46*(4), 949–965.

Dehghanimohammadabadi, M., Keyser, T., (2017), Intelligent simulation: Integration of SIMIO and MATLAB to deploy decision support systems to simulation environment, *Simulation Modelling Practice and Theory,* 71, 45-60

de Kok, T. G., & Visschers, J. W. C. H. (1999). Analysis of assembly systems with service level constraints. *International Journal of Production Economics*, *59*(1), 313–326.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithm*. England: John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

DeJong, C. D. (2001). Simulating test program methods in semiconductor assembly test factories. *Winter Simulation Conference* (WSC '01), 9-12 December, Arlington, Virginia, 1157–1162.

Desmet, B., Aghezzaf, E.-H., & Vanmaele, H. (2010). Safety stock optimisation in two-echelon assembly systems: Normal approximation models. *International Journal of Production Research*, *48*(19), 5767–5781.

Durillo, J. J., & Nebro, A. J. (2011). JMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*, *42*(10), 760–771.

Ferreira, L. P., Ares, E., Peláez, G., Resano, A., Luis, C. J., & Tjahjono, B. (2012). Evaluation of the changes in working limits in an automobile assembly line using simulation. *Manufacturing Engineering Society International Conference* (MESIC 2011), 21-23 September, Cadiz, Spain,

617–624.

Ghani, U., Monfared, R. P., & Harrison, R. (2012). Energy optimisation in manufacturing systems using virtual engineering-driven discrete event simulation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *226*(11), 1914–1929.

Han, Y.-H., Zhou, C., Bras, B., McGinnis, L., Carmichael, C., & Newcomb, P. J. (2003). Paint line color change reduction in automobile assembly through simulation. *Winter Simulation Conference* (WSC '03), 7-10 December, New Orleans, Louisiana, 1204–1209.

Hao, Q., & Shen, W. (2007). Simulating materials handling processes using both discrete event and agent-based technologies. *Transactions of the North American Manufacturing Research Institution of SME* (Vol. 35, pp. 441–448).

Hao, Q., & Shen, W. (2008). Implementing a hybrid simulation model for a Kanban-based material handling system. *Robotics and Computer-Integrated Manufacturing*, *24*(5), 635–646.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *Elements of Statistical Learning*. Springer.

Hellinger, A. and Seeger, H., (2011) Cyber-Physical Systems. Driving Force for Innovation in Mobility, Health, Energy and Production, Acatech Position Paper, National Academy of Science and Engineering.

Hugan, J. C. (2001). Using simulation to evaluate site traffic at an automobile truck plant. *Winter Simulation Conference* (WSC '01), 9-12 December, Arlington, Virginia, 1013–1016.

Ismail, H. S., Tey, V. S., Wang, L., & Poolton, J. (2011). A UML approach for the design of reconfigurable manufacturing simulation models. *IEEE International Conference on Industrial Engineering and Engineering Management* (IEEM 2011), 6-9 December, Singapore, Singapore, 1690–1694.

Iwata, K., & Oba, F. (1984). Simulation for Design and Operation of Manufacturing Systems. *CIRP Annals - Manufacturing Technology*, *33*(1), 335–339.

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research*, *203*(1), 1–13.

Kellner, M.I., Madachy, R.J., Raffo, D.M., (1999) Software process simulation modeling: Why? what? how?, Journal of Systems and Software 46 (2) 91–105

Khalil, R. A., & Stockton, D. J. (2010). Predicting the effects of cycle time variability on the efficiency of electronics assembly mixed-model, zero-buffer flow processing lines. *International Journal of Computer Integrated Manufacturing*, *23*(12), 1149–1157.

Kibira, D., & McLean, C. (2002). Virtual reality simulation of a mechanical assembly production line. *Winter Simulation Conference* (WSC '02), 8-11 December, San Diego, California, 1130–1137.

Koo, L.-J., Park, C. M., Lee, C. H., Park, S., & Wang, G.-N. (2011). Simulation framework for the verification of PLC programs in automobile industries. *International Journal of Production Research*, *49*(16), 4925–4943.

Korytkowski, P., Wisniewski, T., & Rymaszewski, S. (2013). An evolutionary simulation-based optimization approach for dispatching scheduling. *Simulation Modelling Practice and Theory*, *35*, 69–85.

Kouikoglou, V. S. (2000). Optimal rate allocation in unreliable, assembly/disassembly production networks with blocking. *IEEE Transactions on Robotics and Automation*, *16*(4), 429–434.

Koulouriotis, D. E., Xanthopoulos, A. S., & Tourassis, V. D. (2010). Simulation optimisation of pull control policies for serial manufacturing lines and assembly manufacturing systems using genetic algorithms. *International Journal of Production Research*, *48*(10), 2887–2912.

Kulwiec, R. A., of Mechanical Engineers (ASME), A. S., & (IMMS), I. M. M. S. (1985). Materials Handling Handbook (pp. 79–100). John Wiley & Sons.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., Hoffmann, M. (2014) Industry 4.0, Business and Information Systems Engineering 6(4), pp. 239-242

Law, A. M. (2014). *Simulation Modelling and Analysis*, McGraw Hill Higher Education.

Lee, H. F., & Stecke, K. E. (1996). An integrated design support method for flexible assembly systems. *Journal of Manufacturing Systems*, *15*(1), 13–32.

Lee, K., Shin, J. G., & Ryu, C. (2009). Development of simulation-based production execution system in a shipyard: A case study for a panel block assembly shop. *Production Planning and Control*, *20*(8), 750–768.

Lee, K.-G., Koo, L.-J., Park, S.-C., & Wang, G.-N. (2009). A suggestion of simulation framework for verifing PLC program & application in automotive production line. *AIP Conference Proceedings*, 1-3 June, Bali, Indonesia, 163–169.

Lee, J., Lapira, E., Bagheri, B., Kao, H., (2013). Recent advances and trends in predictive manufacturing systems in big data environment, In Manufacturing Letters, Volume 1, Issue 1, Pages 38-41, ISSN 2213-8463,

Lee, Kao and Yang (2014). Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems - Procedia CIRP 16, 3 – 8

Lei, D. (2009). Multi-objective production scheduling: A survey. *International Journal of Advanced Manufacturing Technology*, *43*(9-10), 925–938.

Li, G., Yao, Z., Yuan, Q., & Fang, M. (2006). Optimization methods of the product assembly line system. *IET Conference Publications,* (ITIC 2006), 6-7 November, Hangzhou, China, 19–22.

Li, J., Nagarur, N., & Srihari, K. (2011). Modeling PCB assembly lines in EMS

provider's environment: Integrating product design into simulation models. I *Winter Simulation Conference* (WSC '11), 11-14 December, Phoenix, Arizona, 2336–2345.

Li, Y., Nikolaidis, E., & Mourelatos, Z. (2011). Efficient Probabilistic Reanalysis and Optimization of a Discrete Event System. *SAE International Journal of Materials and Manufacturing*, *4*(1), 1169–1187.

Liu, C. M., & Sanders, L. (1989). Approximate design optimization of asynchronous assembly systems. *International Journal of Computer Applications in Technology*, *2*(1), 30–37.

Liu, C.-M., & Sanders, J. L. (1988). Stochastic design optimization of asynchronous flexible assembly systems. *Annals of Operations Research*, *15*(1), 131–154.

Lu, R. F., Petersen, T. D., & Storch, R. L. (2007). Modeling customized product configuration in large assembly manufacturing with supply-chain considerations. *International Journal of Flexible Manufacturing Systems*, *19*(4), 685–712.

Lu, R. F., & Sundaram, S. (2002). Manufacturing process modeling of Boeing 747 moving line concepts. *Winter Simulation Conference* (WSC '02), 8-11 December, San Diego, California, 1041–1045.

Lu, S., Xu, C., Ray, Y., Wang, Z.L., (2017). A RFID-enabled positioning system in automated guided vehicle for smart factories, In Journal of Manufacturing Systems, Volume 44, Part 1, Pages 179-190, ISSN 0278-6125,

Maccarthy, B. L., & Liu, J. (1993). Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, *31*(1), 59–79.

Manns, M., & Elmaraghy, H. A. (2009). Inter-arrival time patterns in manufacturing systems with main and side loops. *International Journal of Production Research*, *47*(10), 2721–2744.

Marsh, R., Jonik, M., Lanham, J., Cheung, W. M., Newnes, L. B., & Mileham, A. R. (2010). Modelling an assembly process using a close coupled generative cost model and a discrete event simulation. *International Journal of Computer Integrated Manufacturing*, *23*(3), 257–269.

Martinelli, F., & Valigi, P. (1999). Sample path techniques for the control of a real assembly system. In *IEEE Symposium on Emerging Technologies and Factory Automation, (*ETFA '99), 18-21 October, Barcelona, Spain, 1547–1553.

McDonald, T., Van Aken, E., & Ellib, K. (2012). Utilizing simulation to evaluate production line performance under varying demand conditions. *International Journal of Industrial Engineering Computations*, *3*(1), 3–14.

McHaney, R. W., & Douglas, D. E. (1997). Multivariate regression metamodel: A DSS application in industry. *Decision Support Systems*, *19*(1), 43–52.

Mehrsai, A., Karimi, H. R., Rügge, I., & Scholz-Reiter, B. (2011). Application of learning pallets for real-time scheduling by use of artificial neural network. - *International Conference on Software, Knowledge Information, Industrial Management and Applications* (SKIMA 2011), 8-11 September, Benevento, Italy, 60–67.

Mehrsai, A., Karimi, H.-R., Thoben, K.-D., & Scholz-Reiter, B. (2013a). Application of learning pallets for real-time scheduling by the use of radial basis function network. *Neurocomputing*, *101*, 82–93.

Mehrsai, A., Karimi, H.-R., Thoben, K.-D., & Scholz-Reiter, B. (2013b). Using metaheuristic and fuzzy system for the optimization of material pull in a push-pull flow logistics network. *Mathematical Problems in Engineering*, 2013, 1-19.

Mehrsai, A., Thoben, K.-D., & Karimi, H. R. (2013). A fuzzy programming method for optimization of autonomous logistics objects. *IEEE International Conference on Mechatronics,* (ICM 2013), 27 February - 1 March, Vicenza, Italy, 344–349.

Mendes, A. R., Ramos, A. L., Simaria, A. S., & Vilarinho, P. M. (2005). Combining heuristic procedures and simulation models for balancing a PC camera assembly line. *Computers and Industrial Engineering*, *49*(3), 413–431.

Monostori (2014) Variety Management in Manufacturing – Proceedings of the 47th CIRP conference on Manufacturing Systems Procedia CIRP 17 9-13

Negahban, A., & Smith, J. S. (2014). Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, *33*(2), 241–261.

Neira Dueñas, B., Lamas Rodríguez, A., Duro, R. J., & García Del Valle, A. (2007). Macroevolutionary algorithms in the optimization of industrial plants. *IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications,* (IDAACS 2007), 6-8 September, Dortmund, Germany, 255–260.

Nikoukaran, J., & Paul, R. J. (1999). Software selection for simulation in manufacturing: A review. *Simulation Practice and Theory*, *7*(1), 1–14.

N.N. (2013) Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group, acatech, 78

N.N (2013) , Foresight (2013). The Future of Manufacturing: A new era of opportunity and challenge for the UK Summary Report, The Government Office for Science, London

Otamendi, F. J. (2011). Screening cell designs based on synchronous manufacturing. *European Conference on Modelling and Simulation,* (ECMS 2011), 7-10 June, Krakow, Poland, 337–341.

Oyekan, J., Hutabarat, W., Turner, C., Tiwari, A., Prajapat, N., Ince, N., Gan, X., Waller, T., A 3D Immersive Discrete Event Simulator for Enabling Prototyping of Factory Layouts (2015), *Procedia CIRP, 38*(1), 63-67

Petti. A., Hutabarat, W., Oyekan, J., Turner, C., Tiwari, A., Prajapat, N., Gan, X., Ince, N., Impact of model fidelity in factory layout assessment using immersive discrete event simulation, *Operational Research Society Simulation Workshop* (SW2016)*,* 11-13 April, Stratford, U.K.

Posada, J., Toro, C., ; Barandiaran, I., and Oyarzun D et al, (2015) “Visual Computing as a Key Enabling Technology for Industrie 4.0 and Industrial Internet” IEEE CG&A Journal, Published by the IEEE Computer Society 0272-1716/15

Pradhan, S., & Damodaran, P. (2009). Performance characterization of complex manufacturing systems with general distributions and job failures. *European Journal of Operational Research*, *197*(2), 588–598.

Prajapat, N., Vejju, K., & Tiwari, A. (2015). Comparison of multi objective algorithms for discrete event simulation based turbine assembly model optimization. *International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems,* (EUROGEN 2015), 14-16 September, Glasgow, Scotland.

Proth, J. M., Mauroy, G., Wardi, Y., Chu, C., & Xie, X. (1997). Supply management for cost minimization in assembly systems with random component yield times. *Journal of Intelligent Manufacturing*, *8*(5), 385–403.

Radziwon, A., Bilberg, A., Bogers, M., Madsen, E.S., (2014). The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions, In Procedia Engineering, Volume 69, Pages 1184-1190, ISSN 1877-7058

Sahu, A., Pradhan, S., (2017). Quantitative analysis and optimization of production line based on multiple evaluation criteria using discrete event simulation: a review. *Automatic Control and Dynamic Optimization Techniques* (ICACDOT 2017), 9-10 September, Pune, India

Smith, J. S. (2003). Survey on the use of simulation for manufacturing system design and operation. *Journal of Manufacturing Systems*, *22*(2), 157–171.

Spedding, T. A., De Souza, R., Lee, S., & Lee, W. (1998). Optimizing the configuration of a keyboard assembly cell. *International Journal of Production Research*, *36*(8), 2131–2144.

Steinemann, A., Taiber, J., Fadel, G., Wegener, K., & Kunz, A. (2013). Adapting discrete-event simulation tools to support tactical forecasting in the automotive industry. *CoDesign*, *9*(3), 159–177.

Stratman, J. K., Roth, A. V, & Gilland, W. G. (2004). The deployment of temporary production workers in assembly operations: A case study of the hidden costs of learning and forgetting. *Journal of Operations Management*, *21*(6), 689–707.

Taghaddos, H., Hermann, U., AbouRizk, S., & Mohamed, Y. (2010). Simulation-based scheduling of modular construction using Multi-Agent Resource Allocation. *2nd International Conference on Advances in System Simulation,* (SIMUL 2010), 22-27 August, Nice, France, 115–120.

Tiwari, A., Noriega Hoyos, P., & Hutabarat, W., Turner, C., Ince, N.Z., Gan, X., Prajapat, N., (2015). Survey on the use of computational optimisation in UK engineering companies. CIRP Journal of Manufacturing Science and Technology.

Toro, C., Barandiaran, I., Posada, J., (2015). A Perspective on Knowledge Based and Intelligent Systems Implementation in Industrie 4.0, In Procedia Computer Science, Volume 60, Pages 362-370, ISSN 1877-0509

Wang, Y., & Handschin, E. (2000). New genetic algorithm for preventive unit maintenance scheduling of power systems. *International Journal of Electrical Power and Energy System*, *22*(5), 343–348.

Watson, J. D., Cross, I. J., & Albiston, J. N. (2010). Exploring and interpreting the solution space of a simulation. *Operational Research Society Simulation Workshop,* (SW 2010), (pp. 263–265).

Wischnewski, R., & Freund, E. (2004). COSIMIR® transport: Modeling, simulation and emulation of modular carrier based transport systems. *IEEE*

International Conference on Robotics and Automation (ICRA '04), 26 April - 1 May, New Orleans, Louisiana, 5171–5176.

Yang, H., Bukkapatnam, S. T. S., & Barajas, L. G. (2013). Continuous flow modelling of multistage assembly line system dynamics. *International Journal of Computer Integrated Manufacturing*, *26*(5), 401–411.

Yu, Q., Duffy, V., McGinley, J., & Rowland, Z. (2006). Productivity simulation with promodel for an automotive assembly workstation involving a lift assist device. *Winter Simulation Conference* (WSC '06), 3-6 December, Monterey, California, 1935–1939.

Zankoul, E., Khoury, H., (2014). Simulation of On-Shore Wind Farm Construction Process in Lebanon, *International Symposium on Automation and Robotics in Construction and Mining* (ISARC 2014), 9-11 July, Sydney, Australia.

Zeng, X., Wong, W.-K., & Leung, S. Y.-S. (2012). An operator allocation optimization model for balancing control of the hybrid assembly lines using Pareto utility discrete differential evolution algorithm. *Computers and Operations Research*, *39*(5), 1145–1159.

Zhao, Y., Yan, C.-B., Zhao, Q., Huang, N., Li, J., & Guan, X. (2010). Efficient simulation method for general assembly systems with material handling based on aggregated event-scheduling. *IEEE Transactions on Automation Science and Engineering*, *7*(4), 762–775.

Zhong, R.Y., Dai, Q.Y., Qu, T., Hu, G.J., Huang, G.Q., (2013) RFID-enabled real-time manufacturing execution system for mass-customization production, *Robotics and Computer-Integrated Manufacturing*, Volume 29, Issue 2, 2013, Pages 283-292, ISSN 0736-5845,