

**Minimisation of Energy Consumption Variance in
Manufacturing through Production Schedule Manipulation**

by

Christopher Duerden

A thesis submitted in partial fulfilment for the requirements for the degree of Doctor
of Philosophy at the University of Central Lancashire

June 2016

Declaration

I declare that while registered as a candidate for the research degree, I have not been a registered candidate or enrolled student for another award of the University or other academic or professional institution.

I declare that no material contained in the thesis has been used in any other submission for an academic award and is solely my own work.

Signature of Candidate:

Type of Award: Doctor of Philosophy

School: School of Engineering

Abstract

In the manufacturing sector, despite the vital role it plays, the consumption of energy is rarely considered as a manufacturing process variable during the scheduling of production jobs. Due to both physical and contractual limits, the local power infrastructure can only deliver a finite amount of electrical energy at any one time. As a consequence of not considering the energy usage during the scheduling process, this limited capacity can be inefficiently utilised or exceeded, potentially resulting in damage to the infrastructure. To address this, this thesis presents a novel schedule optimisation system. Here, a Genetic Algorithm is used to optimise the start times of manufacturing jobs such that the variance in production line energy consumption is minimised, while ensuring that typical hard and soft schedule constraints are maintained.

Prediction accuracy is assured through the use of a novel library-based system which is able to provide historical energy data at a high temporal granularity, while accounting for the influence of machine conditions on the energy consumption. In cases where there is insufficient historical data for a particular manufacturing job, the library-based system is able to analyse the available energy data and utilise machine learning to generate temporary synthetic profiles compensated for probable machine conditions.

The performance of the entire proposed system is optimised through significant experimentation and analysis, which allows for an optimised schedule to be produced within an acceptable amount of time. Testing in a lab-based production line demonstrates that the optimised schedule is able to significantly reduce the energy consumption variance produced by a production schedule, while providing a highly accurate prediction as to the energy consumption during the schedules execution.

The proposed system is also demonstrated to be easily expandable, allowing it to consider local renewable energy generation and energy storage, along with objectives such as the minimisation of peak energy consumption, and energy drawn from the National Grid.

Contents

Declaration.....	i
Abstract.....	ii
Contents	iii
List of Tables	vi
List of Figures.....	ix
Glossary of Terminology	xv
List of Abbreviations and Notations.....	xvi
Acknowledgements	xxi
1. Introduction	1
1.1. Project background.....	1
1.2. Motivation for Research.....	2
1.3. Aims of Research.....	5
1.4. Research Approach.....	5
1.5. Contribution to Knowledge	7
1.6. Thesis Structure	8
2. Electrical Theory and Energy Monitoring.....	11
2.1. Attributes of Electricity	11
2.2. Energy Monitoring Principles.....	15
2.3. Energy Monitoring System.....	17
2.3.1. Custom Energy Monitoring System	18
2.4. Summary.....	23
3. Modelling Manufacturing Job Energy Consumption	24
3.1. Energy Modelling Literature Review	24
3.2. Energy Modelling Literature Review Conclusions	31
3.3. Intelligent Historical Library for Manufacturing Energy Prediction.....	32
3.3.1. Profile Analysis	34
3.3.2. Model Generation	45
3.3.3. Probability based Selection of Metadata.....	52
3.3.4. Exact Nearest Neighbour Search Algorithm.....	55
3.4. Summary.....	57
4. Schedule Optimisation System – Background	58

4.1. Review of Energy Optimisation in Manufacturing.....	58
4.1.1. Machine Focused Approaches	59
4.1.2. Schedule Focused Approaches	62
4.2. Machine or Schedule Centred Approach?	66
4.3. Proposed Approach.....	67
4.4. Selection of Heuristic Optimisation Algorithm	68
4.5. Summary.....	70
5. Schedule Optimisation System – Implementation.....	71
5.1. Heuristic Algorithm Encoding Scheme	71
5.2. Genetic Operators	73
5.2.1. Population Generation	73
5.2.2. Selection	75
5.2.3. Crossover and Mutation.....	76
5.2.4. Partial Validation of Schedule	78
5.3. Fitness Functions	79
5.3.1. Probable Energy Consumption Variance Prediction Engine	81
5.3.2. Best Case Energy Consumption Variance Prediction Engine	83
5.4. Overall Algorithm Design and Implementation	86
6. Schedule Optimisation System – Performance Enhancements.....	92
6.1. Improvements to Genetic Algorithm Search Efficiency.....	92
6.2. Improvements to Runtime	96
6.3. Coarse Prediction.....	99
6.3.1. Reducing Historical Energy Profile Length.....	100
6.3.2. Additional Methods for Producing a Coarse Prediction	112
7. Experimentation Results and Analysis.....	114
7.1. Experiment Setup	114
7.2. Production Schedule Optimisation – Overview of Results	118
7.3. Production Schedule Optimisation – Performance Analysis.....	130
7.4. Comparison of Most Probable and Best Case Results.....	135
7.5. Analysis of Dynamic Machine Reassignment Influence	136
7.6. Experimental Implementation	139
7.6.1. Implementation Setup	139
7.6.2. Implementation Results	140
7.7. Coarse Prediction – Results and Analysis	143
7.8. Summary.....	150
8. Optimisation System Expandability	151
8.1. Optimisation of Minimum Peak Energy Consumption	151
8.2. Demand Response for Renewable Energy Generation.....	152

8.3. Other Potential Expansions	157
9. Conclusion.....	159
9.1. Addressing the Research Aim and Objectives.....	160
9.2. Future Work Recommendations	164
9.3. Applications in Real World Manufacturing Production Lines	165
9.4. Final Word.....	167
References.....	168
Appendix A – Custom Energy Monitoring Schematic	180
Appendix B – Parameters for Evaluated Machine Learning Algorithms	181
Appendix C – Full Results from Evaluating Machine Learning Algorithms to Generate Synthetic Energy Profiles	182
Appendix D – Waveforms used during Evaluation of Time Series Compression Algorithms	185
Appendix E – Details of Test Schedules	190
Appendix F – Schedule Gantt Charts	204
Appendix G – Test Implementation Schedule.....	216

List of Tables

Table 2.1 - Accuracy of custom energy monitoring system compared against the calibration standard.....	21
Table 3.1 - Results of Mann-Whitney test to determine common groups	41
Table 3.2 - Results from the Mann-Whitney test to determine if either of the two influence periods is not actually influenced by a metadata variable.....	42
Table 3.3 - Processed results from different algorithms trialled to model metadata influence on energy profiles	49
Table 3.4 - Error values for predicted profiles shown in figure 3.10.....	51
Table 3.5 - Rules for generating synthetic profiles for each metadata space plot.....	53
Table 3.6 - Average search times for Exact Nearest Neighbour Search Algorithms on test sets. (Results averaged over 100 consecutive runs. Generated on an Intel i7 1.7 GHz computer with 4GB of memory running Windows 7)	56
Table 5.1 - Comparison results of uniform and single-point crossover strategies (Results are averaged over five consecutive runs. Ran on an Intel i5 Windows 7 PC).....	77
Table 6.1 - Optimisation system runtime throughout different stages of runtime improvements. Results averaged from ten consecutive runs	99
Table 6.2 - Table showing compression induced error from four compression techniques. Compressed waveforms are expanded using 'last known value' interpolation to compare against original. The compression ratio is 10:1	110
Table 6.3 - Table showing compression induced error from four compression techniques. Compressed waveforms are expanded using 'last known value' interpolation to compare against original. The compression ratio is 20:1	111

Table 7.1 - Details of original schedules and specified production line used to test the schedule optimisation system. Schedules optimised for minimal makespan only.....	115
Table 7.2a – Most probable results of the schedule optimisation system (Results produced over ten runs. Percentage of optimisation is relative to the original variance)	118
Table 7.2b - Best case results of the schedule optimisation system (Results produced over ten runs. Percentage of optimisation is relative to the original variance).....	118
Table 7.3 – Difference in makespan between the original and most probable schedules.....	119
Table 7.4 - Table showing the range (max. – min.) of results returned by the optimisation system in 5 consecutive runs.....	130
Table 7.5 - Differences in fitness function evaluation runtime between valid and invalid candidate schedules for the range of test schedules (Results averaged over 100 iterations; Ran on an intel i7 8GB Windows 8.1 computer).....	133
Table 7.6 - Table comparing the job start times and energy profiles used in the most probable and best case results for test schedule #3. Changes are shown in bold.....	136
Table 7.7 - Table showing the job - machine assignments for the original, most probable and best case schedules of test schedule #6. Changes to machine assignment are shown in bold	138
Table 7.8 - Comparison between predicted and recorded variance values for the evaluated schedule	141
Table 7.9 - Error values between the predicted and recorded energy profiles.....	141
Table 7.10 - Table discussing the influence on accuracy and runtime of the coarse prediction methodologies.....	143
Table 7.11 - Differences in fitness function evaluation runtime between valid and invalid candidate schedules using reduced energy profiles. (Results averaged over 100 iterations; Ran on an Intel i7 8GB Windows 8.1 computer).....	143

Table 7.12 - Table showing empirical reduction factors in optimisation system runtime measured for the three coarse prediction methods on all test schedules with most probable result (Results averaged over ten consecutive runs)	144
Table 8.1 - Table demonstrating the level of peak energy consumption reduction compared to the peak consumption when a schedule is optimised for energy consumption variance	152
Table 8.2 - Results of reduction to forecast-demand breach experiment.....	154

List of Figures

Figure 1.1 - Top-level diagram showing the disconnected relationship between production scheduling and the energy monitoring infrastructure.....	4
Figure 1.2 - Top-level diagram showing an overview of the novel aspect with its connection with the energy monitoring infrastructure and scheduling system.....	6
Figure 1.3 – Roadmap of thesis chapters	10
Figure 2.1 - Instantaneous voltage, current and power when the former two measures are in-phase.....	12
Figure 2.2 - Instantaneous voltage, current and power when the former two measures are out of phase.....	12
Figure 2.3 - Example of three phase voltages	13
Figure 2.4 - Examples of resistive, inductive and capacitive loads and their phase influences on voltage and current.....	14
Figure 2.5 - Example of power factor correction	15
Figure 2.6 - Screen capture of the inrush current from an air compressor. Captured using a Fluke 43B Power Quality Analyser	17
Figure 2.7 - Images of the expanded energy monitoring system	20
Figure 2.8 - Diagram of energy monitoring software architecture.....	20
Figure 2.9 - Example of an energy profile for a bench top pillar drill boring a single hole.....	22
Figure 3.1 - Top-level diagram of the intelligent library-based system.....	34
Figure 3.2 - Graph showing the metadata influence on empirical energy profiles	36

Figure 3.3 - Time series differences between all possible combinations of the three available energy profiles.....	38
Figure 3.4 - Cross profile variance.....	39
Figure 3.5 - Normalised cross-profile moving difference.....	40
Figure 3.6 - A difference profile segregated into three groups based on the findings from figure 3.5. Each group is further subdivided and the outer sub-groups (grey) are discarded	41
Figure 3.7 - Recorded energy profiles segregated into influence periods.....	42
Figure 3.8 - Original progenitor profile on which all artificial profiles are based	44
Figure 3.9 - Energy profiles segregated into influence periods. Period 1 is unaffected by any metadata	44
Figure 3.10 - Comparison between a known profile and predictions based on different sized training sets	51
Figure 3.11 - Frequency distribution of empirical metadata	53
Figure 3.12 - Frequency distribution of energy consumption profiles after synthetic profile generation rules have been applied	54
Figure 3.13 - Example of metadata space with merged plots	55
Figure 5.1a - Example of a two process four job production schedule	72
Figure 5.1b - Encoded schedule (from figure 5.1a) with $T = 00:01:00$ and $s_e = 07:00$ (release date for process B)	72
Figure 5.2 - Diagram showing the act of uniform crossover.....	77
Figure 5.3 - Image showing how the fitness function simulates the job influence on a machine. In this example, tool wear is used as a singular influence parameter	82
Figure 5.4- Diagram showing the chromosome compositions when the optimisation algorithm is set to find the most probable and best case predictions. In the case of the most probable	

result, the chromosome does in fact contain a section for the profile ID's (greyed out). However these are ignored by the most probable fitness function as the profiles are selected via exact nearest neighbour	85
Figure 5.5- Example of a single process, two job schedule in the format accepted by the optimisation system.....	87
Figure 5.6- Example of the production line information in the format accepted by the optimisation system.....	88
Figure 5.7- Example of a historical job energy profile in the format accepted by the optimisation system.....	88
Figure 5.8 - Overview of Optimisation System	91
Figure 6.1a - Graph showing the influence crossover and mutation rates along with tournament selection size have on the most optimal found solution with an eight job schedule. Results averaged over ten consecutive runs.....	94
Figure 6.1b - Graph showing the influence crossover and mutation rates along with tournament selection size have on the most optimal found solution with a ten job schedule. Results averaged over ten consecutive runs.....	94
Figure 6.2- Graph showing how the termination conditions affect the most optimal solution found in a 50 job schedule. Optimal pairing shown by stem. Results averaged over ten consecutive runs	96
Figure 6.3 - Overlay of original and mean aggregated energy profile for a turning process (10:1 compression). Original profile is an empirical recording from a metalworking lathe ..	103
Figure 6.4 – Example original 4000 data point energy profile waveform	105
Figure 6.5 - Graph showing the absolute moving difference of data shown in figure 6.4	105
Figure 6.6 - Original waveform overlaid with identification of upper 90% of informative points..	106

Figure 6.7 - Sample of waveform divided into blocks (green dashed lines) and waveform features (red dot & dashed lines)	106
Figure 6.8 - Output waveform compressed from the original by a factor of 10.....	107
Figure 6.9 - Comparison of standard mean aggregation and the custom compression algorithm on empirical data. Compressed waveforms interpolated for comparison	108
Figure 7.1 - Diagram showing experimental implementation of the intelligent library-based system.....	116
Figure 7.2 - Breakdown of the artificial energy profile generation software	117
Figure 7.3 - Example of an artificially generated energy profile	117
Figure 7.4 - Graphs of individual and combined predicted production line profiles for an 8 job schedule (Test schedule #1)	120
Figure 7.5 - Gantt chart for 8 job schedule (Test schedule #1) showing the job layouts for the original, most probable and best case results	121
Figure 7.6 - Graphs of individual and combined predicted production line profiles for a 12 job schedule (Test schedule #2)	123
Figure 7.7a - Gantt chart for 12 job schedule (Test schedule #2) showing the job layouts for the original and most probable results	124
Figure 7.7b - Gantt chart for 12 job schedule (Test schedule #2) showing the job layouts for the best case results.	125
Figure 7.8 - Graphs of individual and combined predicted production line profiles for a 20 job schedule (Test schedule #3)	126
Figure 7.9 - Graphs of individual and combined predicted production line profiles for a 30 job schedule (Test schedule #4)	127

Figure 7.10 - Graphs of individual and combined predicted production line profiles for a 40 job schedule (Test schedule #5)	128
Figure 7.11 - Graphs of individual and combined predicted production line profiles for a 50 job schedule (Test schedule #6)	129
Figure 7.12 - Graph showing the percentage ratio between valid and invalid schedules in the Genetic Algorithm's population during the optimisation of different schedules (Results produced using most probable prediction engine over six consecutive runs).....	132
Figure 7.13 - Graph showing the runtime of the optimisation system. (Results averaged over 5 consecutive runs).....	132
Figure 7.14 - Graph showing the GA restart iteration where the best found value was located (Results averaged and maximum value located through ten consecutive runs).....	133
Figure 7.15 - Graph showing influence of DMR on the most probable and best case results (Results averaged over ten consecutive runs)	138
Figure 7.16 - Comparison between predicted and recorded energy profiles for the original and optimised evaluation schedule	140
Figure 7.17 - Comparison of power factor when using a floodlight and AC motor independently and simultaneously. (Note: When no devices are using power, the meter is unable to measure power factor and hence produce erroneous results around zero).....	142
Figure 7.18 - Graph comparing most probable results from original and coarse prediction methods	145
Figure 7.19 - Graph comparing best case results from original and coarse prediction methods.....	145
Figure 7.20 - Graph showing most probable coarse prediction methods error % compared to original results.....	146

Figure 7.21 - Graph showing best case coarse prediction methods error % compared to original results	146
Figure 7.22 - Graph comparing most probable runtimes for original and coarse prediction methods.	147
Figure 7.23 - Graph comparing best case runtimes for original and coarse prediction methods.	147
Figure 8.1 – Example of production line energy profile which is optimised for demand response with renewable energy sources	153
Figure 8.2 - Second example of production line energy profile which is optimised for demand response with renewable energy sources	153
Figure 8.3 - Example prediction of production line powered from a PV array with battery storage. A backup connection to the National Grid is available for exporting excess power and demanding power when demand exceeds local availability. Note: Battery percentage shown is measured between the fully charged and the acceptable discharge cut-off level.	156
Figure 8.4 - Optimised version of data shown in figure 8.3.....	157

Glossary of Terminology

Deadline

The time at which all jobs in the process are to be complete.

Job (Manufacturing job)

A manufacturing task, typically comprised of elemental manufacturing operations and undertaken on a single machine with material handling operations before and after.

Makespan

The total processing time of a process, ranging from the beginning of the first job to the completion of the last.

Process

A collection of manufacturing jobs which are connected either sequentially or in parallel.

Release date

The time at which the jobs of a process are ready (released) for execution.

Specific energy consumption

The energy required to process a single unit of material.

Tardiness

The positive time between the scheduled completion of a process and its deadline.

List of Abbreviations and Notations

Abbreviations

AC	Alternating Current
Ah	Ampere-hour
DAQ	Data Acquisition device
DMR	Dynamic Machine Reassignment
DSM	Demand Side Management
MLP	Multilayer Perceptron Neural Network
PV	Photovoltaic (Solar panel)
RMS	Root Mean Squared
SVM	Support Vector Machine
SVR	Support Vector Regression
UK	United Kingdom

Symbols

A	Electrical amperage
$abs\Delta profile$	Absolute moving difference between neighbouring points in a <i>profile</i>
$abs\Delta profile_{Normalised}$	$abs\Delta profile$ normalised between zero and one
B	Set of uniformly sized blocks used to divide an energy profile
B_{Cap}	Battery maximum storage capacity
$B_{lv}(t)$	Battery charge level at time t
B^{Size}	Size of blocks in B
$Cross\sigma^2(t)$	Cross profile variance – The variance of all difference profile values at time index t

$d_{u,i}$	Deadline for process u , of which job i is a part
D_{Max}	Latest deadline in the schedule (chromosome domain)
$\Delta Cross\sigma^2$	Moving difference between $Cross\sigma^2(t)$ and $Cross\sigma^2(t-1)$
$\Delta Cross\sigma^2_{Normalised}$	$\Delta Cross\sigma^2$ normalised between 0 and 1
$\Delta profile^{largest}$	The difference profile with the largest numerical range
$\Delta profiles$	A collection of difference profiles
$\Delta profile_{xy}(t)$	Difference profile - The difference between $profile_x$ and $profile_y$ at each time index t
ε'	Combined prediction error from all influence period errors in a profile
ε_{ip_i}	A prediction error value for the model responsible for influence period i
$Emp(y)$	The number of empirical energy profiles within the bounds of hypercube plot y in metadata space
E_{Var}	Predicted production line energy consumption variance
F	Set of neighbouring data points in a time-series energy profile which belong to the same profile feature
f_s	Sampling frequency
f_t	Electrical transmission frequency (50Hz in UK)
g_i	Start time of job i (chromosome domain)
h	Prerequisite job
I_{ms}	Instantaneous current reading
$Hist$	Collection of historical energy profiles for a single manufacturing job
$influence_group$	A collection of neighbouring data points within a profile, which are

affected by machine conditions in a statistically similar way

$ IP $	Number of influence periods in a profile
ip	Collection of <i>influence_groups</i> which share statistically similar influences from metadata
$ ip_i $	Number of data points within influence period i
I_{RMS}	RMS current
J	Number of jobs in the schedule
K	Number of machines available for use on the production line
k_{ip_i}	Value of data point at the centre of influence period i
$MaxSTD$	The metadata's largest standard deviation in any dimension.
$MetaDataA /$ $MetaDataB$	Machine conditions which influence the energy profile. Selected by a domain expert
$MetaData^{Range}[i]$	Either the numerical or operating range of the metadata's i^{th} dimension. Decided by domain expert
$MetaData^{\sigma}[i]$	Standard deviation of the metadata's i^{th} dimension
M_k	Semaphore-based in-use/available profile for machine k
mp_{ip_i}	Model prediction for the value of the data point at the centre of influence period i
N	Size of the Genetic Algorithms population
$N_{Tournament}$	Tournament selection size
$P_{Crossover}$	Probability of crossover
P_i^h	Processing time of job h , which is a prerequisite of job i

$P_{Hit}(y)$	The probability of hypercube plot y in metadata space being hit with a query request for a profile
p_i	Processing time of job i
$P^{Influence}$	Collection of points which specify the boundary points between influence groups
$P_{Mutation}$	Probability of mutation
$profile$	Energy profile for a manufacturing job
$profile_{Demand}(t)$	Production line demand at time t
$profile_{Excess}(t)$	Excess energy not consumed by the production line at time t
$profile_{Export}(t)$	Energy exported to the National Grid at time t
$profile_{Forecast}(t)$	Renewable energy output forecast at time t
$profile_{Import}(t)$	Energy imported from the National Grid at time t
$P^{Informative}$	Collection of points which specify the boundary points between energy profile features
$profile_{P}^{M_k}_{Predicted}$	Time series predicted energy consumption profile for machine M_k
$profile_{P}^{Prod}_{Predicted}$	Time series predicted production line energy consumption profile
$profile_{ref}$	A historical profile used as a reference template for constructing the synthetic profile
$profile_{Required}(t)$	Additional energy required to run the production line after the energy provided by the renewable source has been considered at time t
$profile_{syn}$	A synthetic profile
$Progenitor$	An artificially generated profile used as a template for generating a series of copies influenced by machine conditions

$r_{u,i}$	Release date for process u , of which job i is a part
s_e	Earliest time in the schedule
s_i^h	Start time of job h , which is a prerequisite of job i
s_i	Start time of job i (time domain)
T	Size of discrete time step in the schedule
t	Time index in a time-series
φ	Phase difference between voltage and current waveforms
V_{Ins}	Instantaneous voltage reading
V_{RMS}	RMS voltage

Acknowledgements

None of this work would be possible without the tireless efforts of my supervisors - Lik-Kwan Shark, Geoff Hall and Joe Howe, along with my industrial supervisors Damian Adams and Stuart Barker. I could not ask for a better team of people. The same can be said for those at the Advanced Digital Manufacturing Technology research centre. Carl, Anthony, Xiao, Aftab, Richard, Ben, Mike and Joanne, thank you.

I would also like to thank BAE Systems (Operations) Ltd. and the Engineering and Physical Sciences Research Council for funding all of this work.

Finally, Mum and Dad, thank you for everything, for reading through reports, for checking presentations, for always being there.

This thesis is dedicated to my parents – Colin and Elaine Duerden

CHAPTER 1

INTRODUCTION

This thesis investigates the incorporation of two historically disjointed domains – energy consumption information, and manufacturing production line scheduling. Through the assimilation of these two, energy-related objectives can be considered during the scheduling process, allowing for significant optimisation within the energy domain while satisfying the constraints typical of production scheduling. This thesis documents background research, subsequent investigation, and a proof-of-concept implementation, for the purpose of ascertaining whether energy related information can be reliability and accurately used during the scheduling process.

1.1 Project Background

In modern manufacturing production lines, the scheduling of production jobs is one of the primary factors which bring about the ability to produce a large volume of high quality products at a high rate while minimising cost and overall wastage. This is achieved by scheduling systems accounting for numerous factors and constraints such as the jobs to be executed, their machine and material requirements, the resources of the production line, shipping and delivery dates, etc. All this is considered, and the resulting output is a production schedule which specifies precisely when jobs need to begin execution, along which what resources and material are required. Typically, this is organised such that total makespan – the time between the start and end of a sequence of jobs for a product is minimised with the projected completion date aligning as close as possible to products delivery date. During this process, energy, which can be considered as one of the staple inputs for operating a production line is seldom considered as a process variable despite its consumption typically being monitored and recorded.

The industrial sector is one of the largest consumers of energy, with manufacturing being the highest consuming sub-sector. Globally in 2007, manufacturing was responsible for 90% of industry energy consumption along with 84% of its energy related CO₂ emissions (Duflou et al, 2012). While the consideration of energy-related goals is by no means a new prospect, it has primarily been undertaken by energy intensive industries such as primary metal, oil, and paper processing. On the contrary, similar considerations for the discrete part and product manufacturing sectors have only majorly come to light at the turn of the century, despite potential. In a report on manufacturing competitiveness, Deloitte attributes the importance of ‘Energy costs and policies’ to be almost equal to labour costs and materials (Roth et al, 2012, pp. 7). This shows that in the same way companies have historically moved their production to areas of reduced labour costs, similar savings can be achieved through the increased consideration of energy. Due to numerous international and governmental legislations, namely the Kyoto Protocol, Climate Change Levy and Climate Change Act (United Nations Framework Convention on Climate Change, No date; Environmental Taxes Reliefs and Schemes for Businesses, 2015; Committee on Climate Change, No date), the discrete and production manufacturing sector is being coerced into considering more energy-based objectives during normal operations.

1.2 Motivation for Research

Individually as manufacturing jobs, hereby referred to as jobs, are being executed, the machine performing the task demands and consumes a different amount of energy depending on the elemental manufacturing operation currently being conducted. Over time, this results in the job’s energy consumption profile being highly volatile with large peaks and troughs, the amplitude and timespan of which are dependent upon the machine and job. When this is not accounted for, and multiple jobs are executed concurrently in a manufacturing production line, this volatility is translated through and amplified as the sum result of concurrently running jobs. For electricity, the infrastructure which consists of both transmission cables and power control

equipment, and connects the production line to the National Grid has a finite capacity. Should the production lines instantaneous demand for electricity ever exceed this limit, damage at a significant financial cost can occur.

To ensure against this, current power infrastructures are designed with extra capacity to accommodate unforeseen peaks in energy consumption. If the problem still persists, naive power cutters can be installed to shutdown specific processes should the power consumption ever exceed a specified level. While these techniques are proven, they do come with limitations of varying severity. Not all processes are able to withstand arbitrary shutdowns and this can negatively impact production rates. Also, while the power infrastructure can be suitably overdesigned, one key question is what to do when the production line is to be expanded – Does the power infrastructure need expanding with it, or can it be used more efficiently?

Manufacturing is a highly competitive industry where costs need to be minimal and stringent deadlines must be met. Therefore, minimising infrastructure and/or expansion cost is considerably appealing. Optimising energy consumption at the process level is seen to be inefficient. Because not all processes and machines are alike, the effectiveness of generic strategies can be limited, and there can be a high time requirement for developing machine/process specific ones. Also manufacturing production lines are highly sensitive environments and manufacturers are hesitant to adjust proven process parameters. Approaching this problem at the schedule level is seen as a suitable alternative. It requires no changes to the individual jobs or machines, and although dependent upon the schedule constraints, can have a significant amount of flexibility. However historically, as seen in figure 1.1, energy consumption data has been disjoined from the scheduling process and has only been considered from a financial perspective. Research into predicting and modelling energy consumption within manufacturing production lines is also limited to single-value statistics such as average and maximum energy demand, as opposed to time-variant predictions. Furthermore with scheduling being an NP-hard problem, accounting for additional factors only acts to increase the problem's complexity (Pach et al, 2014; Fang et al, 2011). Schedule-based approaches have historically been used for energy optimisation. At the National Grid level, load scheduling, also

known as load shifting or Demand Side Management (DSM), is used to govern the level of peak consumption. Here industry is encouraged to run energy intensive processes during off-peak times via financial incentives (Brown et al, 2012). Further research also expands this for use with variable-supply renewable energy sources (Emec et al, 2013), and there is interest in applying this in domestic environments (Sung and Ko, 2015; Dang and Ringland, 2012).

These works show the potential for schedule-based energy optimisation, however currently used methods do not approach this from an energy perspective. Cases where this is, are so far mainly confined to research and in certain cases, scheduling constraints are breached in favour of a better result (Mouzon et al, 2007). Additionally, a common energy-based objective – the minimisation of peak energy consumption, does not take advantage of a power infrastructure’s ability to be overloaded for a short, defined period of time. This is thanks to the use of protection devices such as slow-blow fuses and time delay circuit breakers. Furthermore, while load scheduling is utilised, given the volatile nature of job energy profiles, it may be possible for certain jobs to run at a degree of concurrency while keeping overall energy consumption below a predefined level. However given the high level of abstraction used in current energy modelling methods, its potential is yet to be realised.

One potential alternative would be the consideration of the energy’s consumption variance. In minimising the variance, the production lines energy consumption would be stabilised as sufficiently as allowed by the constraints, while still allowing for a small volume of large, short lived peaks where/if necessary. This would maintain flexibility during the scheduling process while reducing the overreliance on the over-design of the power infrastructure.

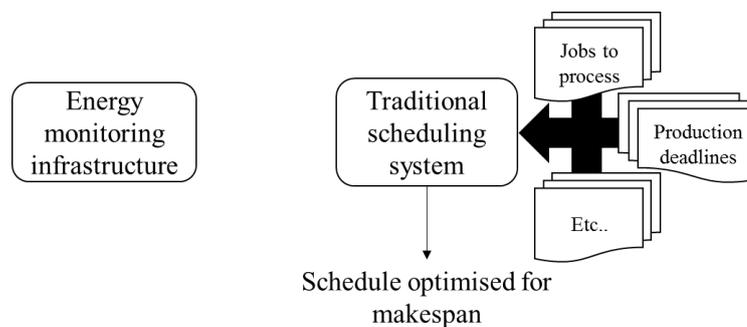


Figure 1.1 - Top-level diagram showing the disconnected relationship between production scheduling and the energy monitoring infrastructure.

1.3 Aims of Research

In light of the situation discussed in the previous section, this thesis aims to develop a methodology which is able to optimise a production schedule such that it produces a minimal variance in the production line's energy consumption when executed. To achieve this, the overall work is segmented into three accompanying objectives, each addressing a field of research which needs to be considered. The overall aim and subsequent objectives are stated below.

Aim of Research

Develop an original methodology for optimising manufacturing production schedules so that when executed on a known production line, they produce a minimal variance in the production lines energy consumption.

Research objectives

1. Investigate and develop industry based energy monitoring methods.
2. Investigate and develop state-of-the-art methods for modelling energy consumption in manufacturing production line environments.
3. Investigate and develop current and state-of-the-art methods for optimising production schedules for energy-based objectives.

1.4 Research Approach

To address the research aim, this thesis presents an expansion to the original framework seen in figure 1.1, such that energy is a heavily influential factor during the scheduling process. Figure 1.2 visualises this, where a schedule optimisation system is used to bridge the gap between production scheduling and energy monitoring. The end result is a schedule which is optimised to produce a minimal variance in the production line's energy consumption when executed. To achieve this, a production schedule produced by a traditional production planning system is input into the optimisation system, which acts on the individual job start times. To overcome the

NP-hard nature of scheduling problems, the core of the schedule optimisation system comprises of a Genetic Algorithm. This is customised for solving production scheduling problems and references historical energy data to determine how to modify the schedule in order to attain the desired outcome.

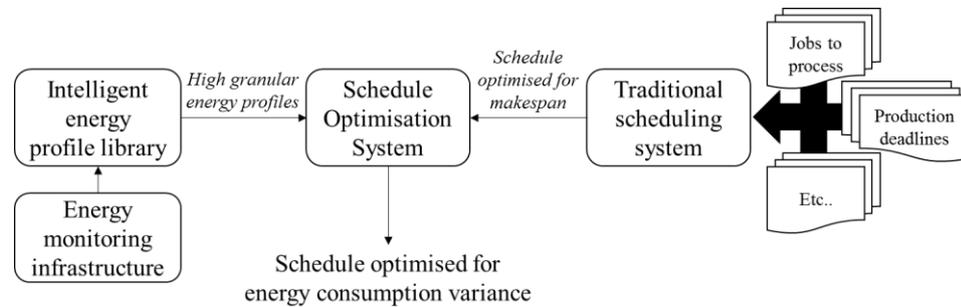


Figure 1.2 - Top-level diagram showing an overview of the novel aspect with its connection with the energy monitoring infrastructure and scheduling system.

Job-specific time-variant energy consumption information is provided by an intelligent energy profile library which is connected to the production lines energy monitoring infrastructure. This library is able to store multiple historical energy profiles for each job, along with select information regarding the mechanical conditions of the machine at the time of recording. This, in collaboration with the schedule optimisation system, allows the energy consumption predictions to compensate for changes due to varying machine mechanical factors such as age and tool wear. In the case where historical data is lacking for a particular job, the library is able to analyse how these recorded mechanical factors influence the energy consumption profiles. Synthetic profiles are then generated to temporarily increase data density.

As this particular application focuses on energy consumption variance, highly granular energy profiles are used, allowing the schedule optimisation system to produce an optimised schedule with a high accuracy but at a cost of optimisation runtime. Where this becomes an issue, three methods for producing a coarse accuracy prediction at a significantly reduced runtime are proposed and evaluated to determine which produces the best cost-to-benefit ratio.

Once an implementation was complete, a comprehensive test scheme was used to evaluate the performance of the entire system. This involved the use of production schedules, containing a

different number of jobs and originally constructed using a traditional production planning system generating the schedules for minimal makespan.

It is important to note that this methodology of reducing the variance in production line energy consumption does not reduce the sum energy consumed. It instead redistributes it evenly over the time allowed by the schedule constraints, namely the earliest start times and deadlines of each job's parent process.

As energy is used via a variety of mediums in the manufacturing sector, to give this work focus, all future references to energy within this report will refer to electrical energy. More specifically the variable of energy which will be the main focus for optimising the consumption variance will be electrical current.

1.5 Contributions to Knowledge

This work provides several original contributions to knowledge within the field of engineering. The following points breakdown this contribution.

- A Genetic Algorithm is customised for modifying a schedule's job start times while maintaining all schedule constraints, with the goal to minimise the variance in the associated production line's energy consumption.
- Highly granular energy consumption profiles are used for all predictions permitting high temporal accuracy.
- To compensate for machine related changes to energy consumption, multiple profiles are stored for each job along with select machine information. In collaboration with the optimisation system predicting machine conditions when jobs are to be executed, predictions are made using the most probable historical energy profiles.
- Where historical energy data is lacking, synthetic profiles can be generated for particular machine conditions based on the probability of those conditions being encountered.

- The implementation is found to be easily expandable, with investigations into its applications in existing energy based optimisation schemes, such as the minimisation of peak energy consumption and Demand Side Management.

Selected work has been disseminated into the research community via a journal paper, a book chapter, public presentations and conference papers along with associated presentations (Duerden et al, 2014a; Duerden et al, 2014b; Duerden et al, 2015; Duerden et al, *In print*; Duerden et al, *Accepted for publication*). The importance of the work has also been acknowledged through the award of best paper (Duerden et al, 2014a) and second prize in the IET's Present Around the World competition.

1.6 Thesis Structure

A brief overview of the chapters within this thesis follows, with a roadmap of how they interrelate shown in figure 1.3.

Chapter One: Introduces the problem investigated by this work and gives an overview of the thesis.

Chapter Two: This chapter provides the reader with an introduction to electrical theory, the components of electrical energy and the methods used in monitoring them. Details regarding the custom energy monitoring system developed for this work are also provided.

Chapter Three: In this chapter, the current state-of-the-art strategies for energy consumption modelling in manufacturing environments are discussed. To address the shortcomings found, the Intelligent Historical Library for Manufacturing Energy Prediction is proposed to provide time-series energy profiles at a high temporal granularity, sourced directly or indirectly from historical recorded data.

Chapter Four: This is the first chapter of three which discusses the developed schedule optimisation system. Here, the literature review outlines the current state-of-the-art methodologies for optimising manufacturing production lines for energy-related objectives.

Two umbrella strategies for achieving this are observed, necessitating a discussion into which strategy is best fitted for solving the problem of energy consumption variance. Following this, using the literature review findings as reference, the core algorithm for implementing the schedule optimisation system is selected.

Chapter Five: Following the background research, this chapter provides a detailed description of the schedule optimisation system. This includes the integration of the main optimisation algorithm and the choices made with regards to the various sub-level algorithms, along with the additional considerations required for this algorithm to solve scheduling-based problems. Finally, the numerous features of the optimisation system are introduced which act to provide either additional information to the manufacturer, or to aid in the system locating a better result. The systems prediction engines used for determining the energy consumption variance produced were a particular schedule executed are also detailed, along with their connection to the intelligent energy profile library.

Chapter Six: Concluding the chapter trilogy, this chapter discusses the experimentation conducted on the schedule optimisation system to optimise its various internal parameters, maximising result optimality while minimising runtime. For cases where the optimised runtime is unsuitable, this chapter introduces three separate methodologies for operating the schedule optimisation system at a coarse level of accuracy.

Chapter Seven: This chapter discusses the testing scheme used to evaluate the performance and result quality of the developed system. Following testing of the core components, the individual features and abilities of the system are investigated along with a detailed look into the three proposed methods for producing a coarse accuracy prediction.

Chapter Eight: This short chapter details two potential alternative applications of the developed system. One focuses on purely minimising peak energy consumption and results are compared against those produced for minimal energy consumption variance. The second application is designed to fit the production lines energy consumption to the predicted response

of renewable energy resources, such that a production line could potentially operate while drawing minimal energy from the National Grid.

Chapter Nine: Concluding the work, this chapter summarises all the findings and evaluates them in the context of the original aim and objectives of research. Shortcomings and issues are addressed along with a discussion on the plausibility of the developed system being implemented within an actual manufacturing production line.

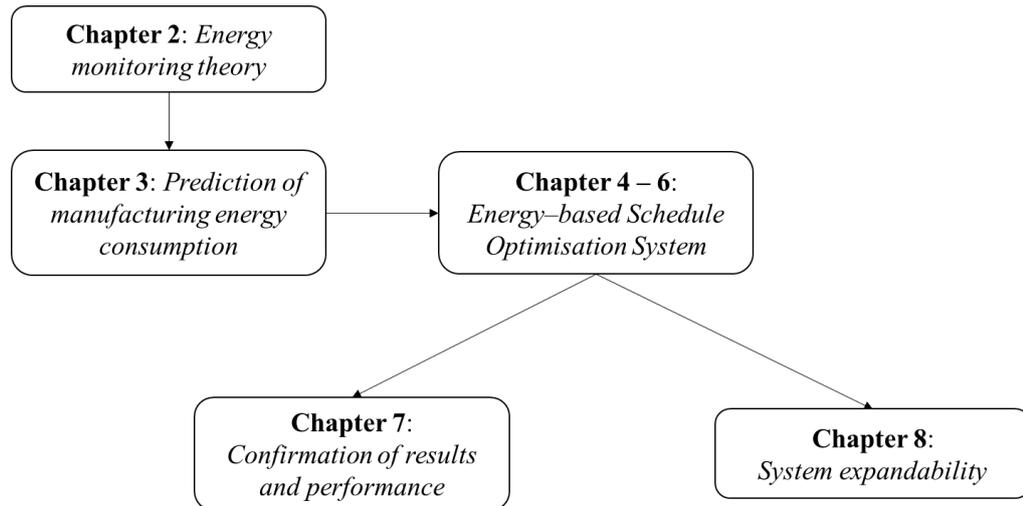


Figure 1.3 – Roadmap of thesis chapters.

CHAPTER 2

ELECTRICAL THEORY AND ENERGY MONITORING

Electricity can be considered one of the most popular forms of energy used in manufacturing, with practically all production machines utilising it to some capacity. In this chapter, the attributes and monitoring principles for electricity are documented, followed by a description of the high granularity custom energy monitoring system used for this work. It is important to note that while power and energy are two different but related measures, these terms are used interchangeably in literature and by industry. Both energy and power monitoring devices can be purchased which solely measure power consumption. To align with this, this work will refer to energy monitoring and energy consumption with relation to power.

2.1 Attributes of Electricity

Alternating Current (AC) electricity is a multivariate form of energy with each parameter sharing some form of interrelationship. These are

- Voltage (V)
- Current (A)
- Apparent power (VA)
- Active power (W)
- Reactive power (Var)
- Voltage and current phase difference (ϕ)

Voltage and current are the two elemental measures from which all other parameters are calculated. In the United Kingdom, government regulation states that the end use voltage levels must remain within the range 216.2V to 253V with a mean value of 230V (Standards & Technical Regulations Directorate, 2005), with a frequency range of 50Hz \pm 0.5Hz (National

Grid, No date). Similar regulations apply for three-phase electricity and compliance with all these regulations is the responsibility of the network operators.

For the power measures, unlike apparent power, active and reactive power account for the phase difference between voltage and current, which is an important consideration for both energy suppliers and manufacturers.

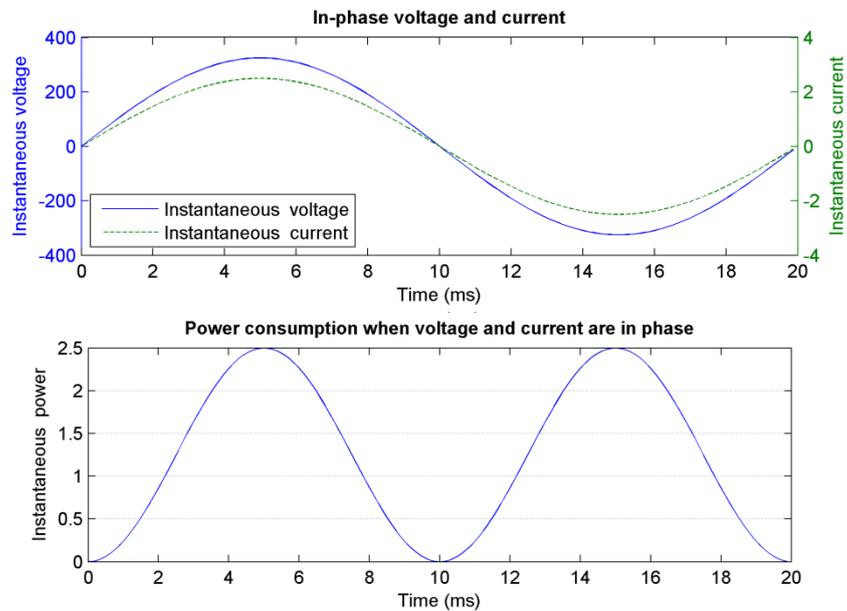


Figure 2.1 - Instantaneous voltage, current and power when the former two measures are in-phase.

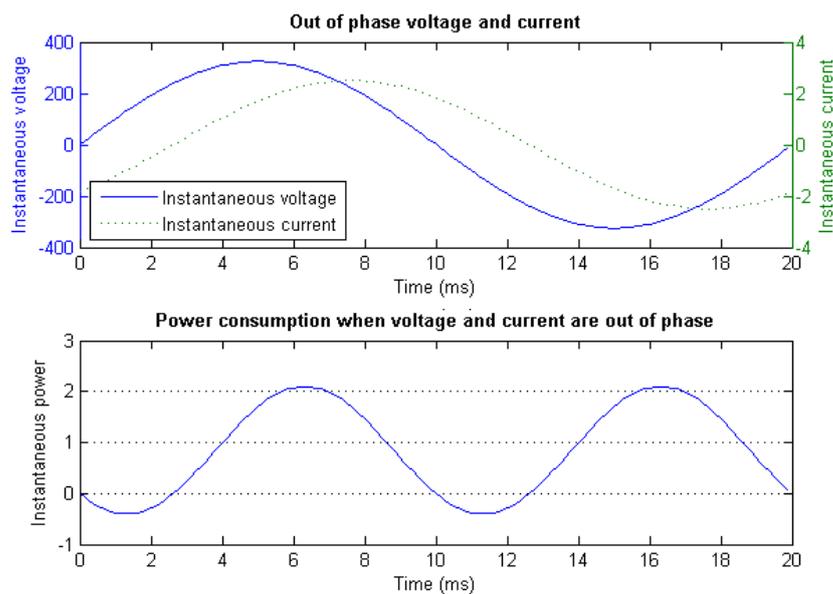


Figure 2.2 - Instantaneous voltage, current and power when the former two measures are out of phase.

Electrical loads can be classified as resistive, capacitive, inductive, or a combination of all three. From milling machines to lathe, drills and robotic arms, the most common components of manufacturing machines are AC motors. As a consequence, manufacturing lines can be electrically modelled as predominantly inductive loads with smaller resistive and capacitive elements. In all but resistive loads, a phase shift occurs between the instantaneous current and voltage with inductive loads causing the voltage to lag behind current and vice versa for capacitive loads. A comparison of figures 2.1 and 2.2, demonstrates that when a phase shift is present, the instantaneous product power has a negative component. This is referred to as reactive power and is considered ‘useless power’ as it does not contribute to useful work (nPower, No date). As a consequence, the overall power level, referred to as apparent power is diminished. This forces machines to increase current consumption to increase the amplitude of the positive component, known as active power. The presence of reactive power is therefore highly undesirable as the end user is paying for power they are not using and for energy suppliers, not only must they output more apparent power but in addition, the return cabling of the National Grid is not designed to transmit significant amounts of power.

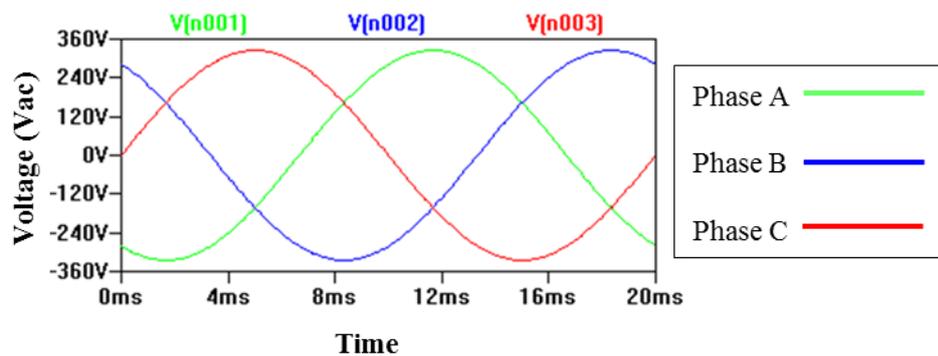


Figure 2.3 - Example of three phase voltages.

For most of its route from source to end use site, electricity is transmitted as three-phase with each of the three cables conducting its own voltage and current $\pm 120^\circ$ out of phase with the others (see figure 2.3). From an electrical perspective, operating goals aim to ensure that the load on each phase is balanced. When this occurs, each of the three currents negate each other removing the requirement for a return cable, however in practice as loads may never be

perfectly balanced a small capacity single return cable is present. As a result transmission networks are typically not configured for returning large power levels. The measure for the phase difference ϕ , is known as Power Factor and is calculated using (2.1) to ascertain a number between zero and one. To protect the National Grid, most transmission network operators will financially penalise end users with a power factor below either 0.95 or 0.9 (Ware, 2006).

$$\text{Power Factor} = \frac{\text{Active Power (W)}}{\text{Apparent Power (VA)}} = \cos(\phi) \quad (2.1)$$

To combat poor power factor, power factor correction techniques can be employed to realign the voltage and current sinusoids. As inductive and capacitive loads have complimentary effects on the phase as seen in figure 2.4, for inductive loads, banks of capacitors connected to the power systems can be employed to realign phase shifts. Figure 2.5 shows an example of all three load types where the inductive and capacitive effects negate each other. The correction of power factor is the responsibility of the end user.

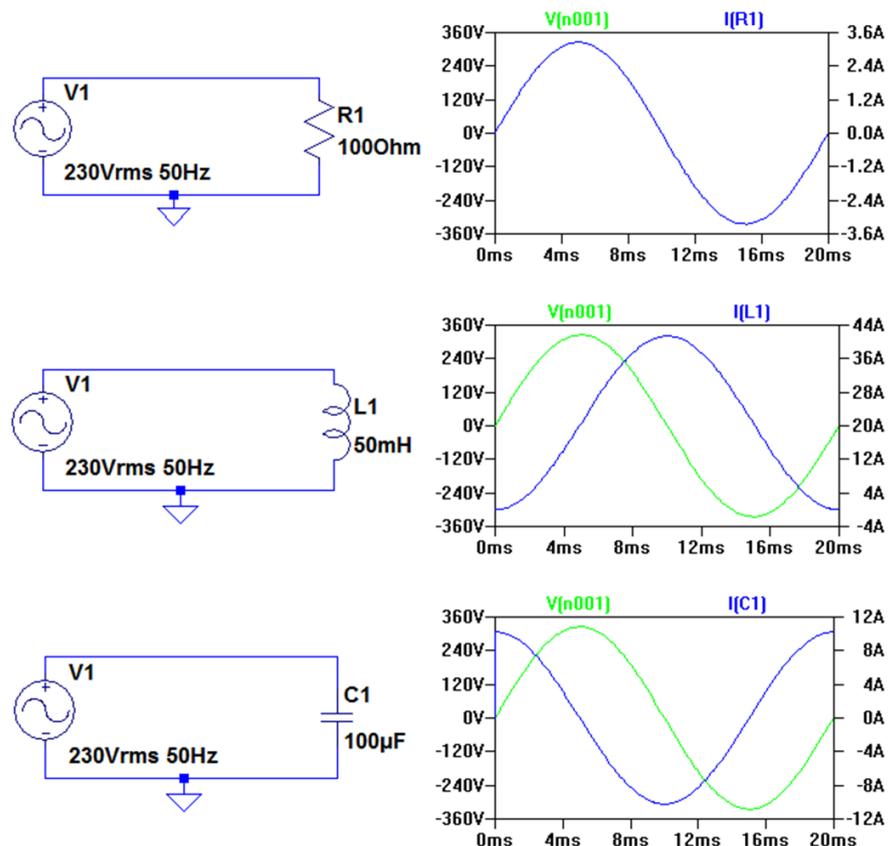


Figure 2.4 - Examples of resistive, inductive and capacitive loads and their phase influences on voltage and current.

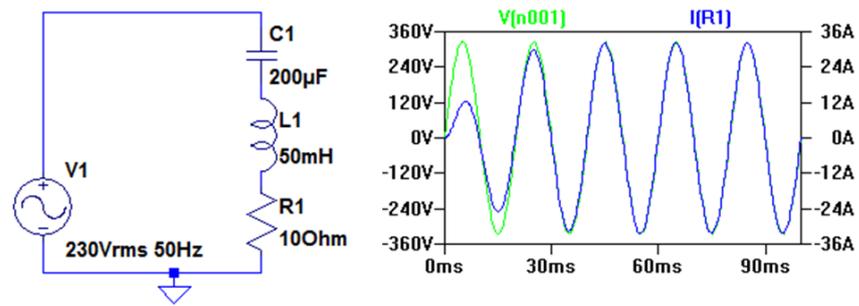


Figure 2.5 - Example of power factor correction.

2.2 Energy Monitoring Principles

One of the key prerequisites for any energy-based optimisation in the manufacturing sector is to understand how energy is distributed and consumed in production lines. Numerous off-the-shelf devices are available for this with single and three phase applications.

As explained in the previous section, voltage and current are the two elemental measures from which all power parameters can be calculated. Both AC voltage and current are sinusoidal waves, meaning that can be easily digitised by discrete electronics for measuring and analysis. Prior to this, conditioning circuitry is required to ratio-metrically reduce the amplitudes to levels acceptable for the electronics. For voltage, the most popular choice is to use a voltage transformer, while for current, more options are available. A shunt resistor can be placed in series with the electrical system and the voltage drop across it can be measured. Ohm's law can then be used to calculate the current through the resistor. While a possible method, this is not significantly popular due to its intrusive nature. Non-intrusive alternatives include split-core current transformers which are simply clamped around the cable to be measured. The current flowing through the cable induces a radio-metrically reduced current into the transformer which can then be measured via a shunt resistor. Similarly, a Rogowski coil can also be used for non-intrusive measuring. The main reason these two methods are popular is due to their non-intrusive nature, which facilitates quick, easy and safe installation and removal.

Standard UK electricity is transmitted at a frequency of 50Hz, necessitating a minimum sampling frequency of 100Hz according to Nyquist theorem (2.2).

$$\begin{aligned}
f_s &\geq 2 \times f_t \\
&\geq 2 \times 50 \pm 0.5 \text{Hz} \\
&\geq 100 \pm 1 \text{Hz}
\end{aligned} \tag{2.2}$$

where f_s is the sampling frequency and f_t is the transmission frequency.

The features and functionality of a wide range of industry grade energy monitoring systems were initially investigated during this work. In every case, the monitor sampled the voltage and current waveforms at a high rate, typically in the low kHz range. Values are recorded for a defined measurement interval, after which the data is computationally amplified to negate the original amplitude reductions, and the values for RMS voltage and current are calculated. Any phase difference ϕ , can be measured by determining the time difference between the zero-crossings of the voltage and current waveforms. With these three measures, the power parameters introduced in the previous section can be calculated using (2.1), and (2.3) to (2.5) (Moulin, 2002).

$$\text{Apparent Power (VA)} = V_{RMS} \times I_{RMS} \tag{2.3}$$

$$\text{Active Power (W)} = V_{RMS} \times I_{RMS} \times \cos(\phi) \tag{2.4}$$

$$\text{Reactive Power (Var)} = \sqrt{\text{Apparent Power}^2 - \text{Active Power}^2} \tag{2.5}$$

The principles described here are based around single phase electricity, however they are applicable to three-phase electricity. While single phase systems involve a live and neutral cable only, three-phase systems typically involve four cables – three live cables, one per phase, and a single neutral (Tekronix, No date). As a result, each phase will have its own voltage, current and power parameters. It should be noted that for three-phase systems, voltage can be measured between a phase and neutral, known as phase voltage, or between one phase and another, known as line voltage. Phase voltage is used in power calculations, although methods such as the two wattmeter method (Newtons4th Ltd., 2012) are available for measuring power on all phases without requiring three individual voltage and current measurements.

Once all the parameters have been calculated, they are either output or internally recorded at a fixed or user defined rate. This reporting rate governs how often the monitor measures the electrical consumption and calculates parameters, but is independent of the sampling rate and measurement interval. In all of the off-the-shelf monitors investigated during this work, it was found that they reported back data at a slow interval, with most devices having reporting rates in the minutes range. As this work focuses on energy consumption variance, it is necessary to understand the energy consumption at a much finer granularity. Through the use of a power quality analyser, it was deduced that the time period of certain waveform features such as inrush currents can potentially only be a few hundreds of milliseconds long as seen in figure 2.6.

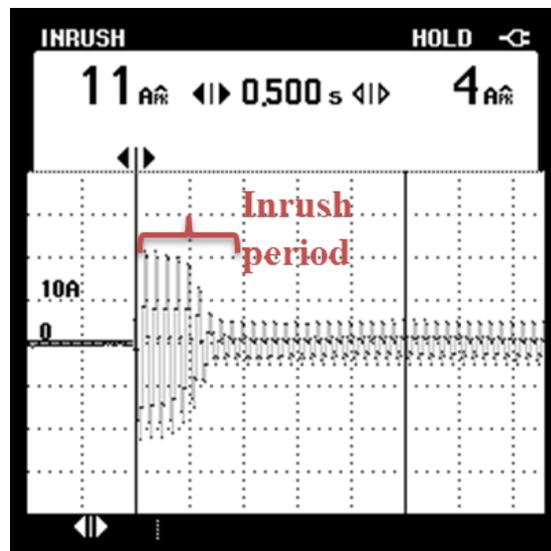


Figure 2.6 - Screen capture of the inrush current from an air compressor. Captured using a Fluke 43B Power Quality Analyser.

Due to this, in comparison with available monitoring capabilities, it was determined that a custom high granularity monitoring system would be required for this work. In the following section, the development of this energy monitor is discussed along with the background research.

2.3 Energy Monitoring System

As the overall goal of this work was to minimise energy consumption variance, one of the key requirements for this project's energy monitoring system was for a high, preferably sub-second

reporting rate. This would allow for rapid changes and short-lived features of the energy profile such as the inrush period seen in figure 2.6, to be recorded at a suitable temporal resolution. As the preliminary review of industry grade monitoring systems did not come close to that level, the initial requirement specification seen below, simply called for a meter with the fastest reporting rate.

- Measurements –
 - RMS voltage and current
 - Active power
 - Apparent power
 - Reactive power
 - Power factor
- Stream data using open protocols or save directly to an accessible file.
- Fastest reporting rate within an acceptable price range.

With industry grade solutions having poor reporting rates, two ‘off-the-shelf’ energy monitoring solutions were investigated. The first was a home appliance monitoring system with a six second reporting interval (Current Cost, No date). While fast in comparison to other products, its reporting capabilities were unreliable, with readings being randomly missed during the periodic recording. The second was an open-source energy monitoring development kit (Microchip, No date). The firmware was streamlined to minimise computational time, but hardware limitations resulted in a minimum achievable reporting interval of four seconds.

2.3.1 Custom Energy Monitoring System

Ultimately, it was deduced that the current level of energy monitoring equipment did not support an acceptable reporting interval required for generating highly granular historical data. This is further supported by the literature review in section 3.1, where researchers who use empirical data and specify their experiment setup, utilise expensive power quality analysers or

high speed data acquisition equipment. To this end, it was realised that a custom energy monitoring system would be required.

A LabVIEW based system was initially developed with software based around the ‘myPowerMonitor’ project to achieve a single voltage and current channel monitor with a 200ms reporting rate (S, 2012). In order to attain this short reporting interval, the initial specification was pruned to the following:

- Measure RMS voltage and current
- Stream data using open protocols or save directly to an accessible file.

This simplification was justified by the fact that the power factor of the electrical supply could not be regulated during experimentation. Therefore there would be little correlation between it and the machine being operated. Furthermore, the measured current is what is actually being demanded by the equipment and therefore is dependent on the power factor. With this new specification, apparent power could still be calculated offline according to (2.3), however power factor, active and reactive power were all removed.

Eventually this implementation was heavily modified, adding six additional 30A current channels along with software rewritten in C# to achieve a 150ms reporting rate on all channels. Images and diagrams of these can be seen in figures 2.7 and 2.8, with hardware schematics in appendix A. In terms of the internal hardware, a connection with the mains electricity line was made using a 30Aac - 1Vac split-core current transformer, with voltage being measured via a 230Vac to 6Vac transformer. Both these interfaces operate to reduce the amplitude of the mains voltage and current signals so they can be measured by discrete electronics. In the case of this custom system, these reduced amplitude signals are read into one of two USB-1208FS data acquisition devices (DAQ’s) via their on-board 12-bit analogue to digital converter. With a differential input range of $\pm 10V$, this would result in a quantisation error of $\pm 2.44mV$. All inputs are then multiplied by their respective channels re-scale value which corrects for the amplitude reduction. For every reported reading, 50 instantaneous values are read in at a sampling rate of 1 kHz per channel. RMS value can then be generated for the voltage and current according to

(2.6) and (2.7) respectively. At a UK mains electricity frequency of 50Hz, this would allow for 2.5 complete cycles to be read in prior to an RMS value being calculated. Post calculation, these values are written to a file for later analysis.



Figure 2.7 - Images of the expanded energy monitoring system.

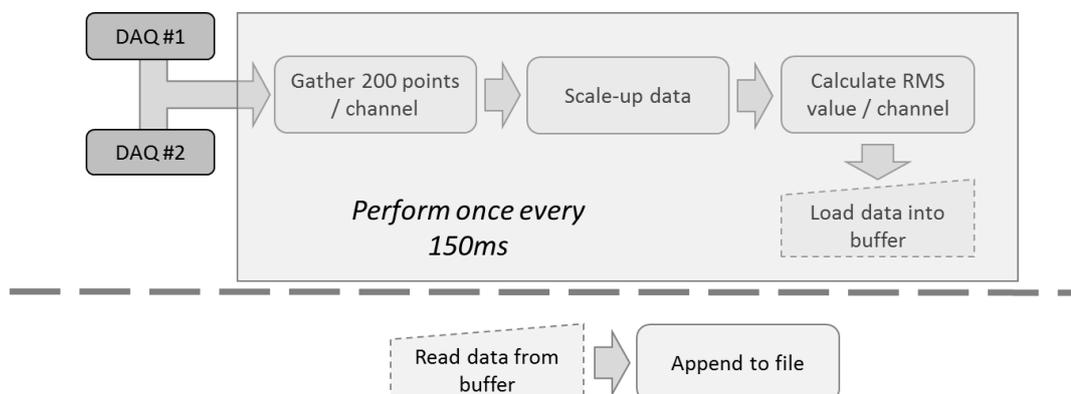


Figure 2.8 - Diagram of energy monitoring software architecture.

$$V_{RMS} = \sqrt{\frac{1}{n}(V_{Ins,1}^2 + V_{Ins,2}^2 + \dots + V_{Ins,n}^2)} \quad (2.6)$$

where a set of n values $\{V_{Ins,1}, \dots, V_{Ins,n}\}$ represent the instantaneous voltage samples

$$I_{RMS} = \sqrt{\frac{1}{n}(I_{Ins,1}^2 + I_{Ins,2}^2 + \dots + I_{Ins,n}^2)} \quad (2.7)$$

where a set of n values $\{I_{Ins,1}, \dots, I_{Ins,n}\}$ represent the instantaneous current samples

For both developed systems, a Fluke 113 true RMS multimeter and a UNI-T UT58C multimeter were used as calibration standards for the voltage and current measurements respectively. Due to its suitably constant current consumption, a 2kW fan heater was used for the calibration process. During this, both the custom energy monitor and the calibration standards were used to measure the voltage and current consumption of the fan heater. The re-scale values for each input channel within the custom energy monitoring program were then updated based on the difference in readings between the custom energy monitor and the calibration standards. This was repeated until the difference was minimal. The final calibration results for the custom energy monitor can be found in table 2.1.

Table 2.1 - Accuracy of custom energy monitoring system compared against the calibration standard.

Channel	Calibration standard reading (RMS)	Energy monitor reading (RMS)	Error (%)
Voltage	251.00V	248.9V	-0.84
Current #1	10.00A	10.12A	1.20
Current #2	9.93A	9.82A	-1.11
Current #3	9.94A	9.97A	0.30
Current #4	9.98A	10.12A	1.40
Current #5	10.03A	9.97A	-0.60
Current #6	9.94A	10.12A	1.81
Current #7	10.01A	10.27A	2.60

Table 2.1 shows that overall, the system is highly accurate with a maximum error of 2.6% (current channel #7). In the case of the current channels, this is due to custom cradles which are not only used to hold the split-core current transformers in place, but also ensure the cable passes through the centre of the core at a perpendicular angle. This maximises the efficiency of the transformer. Equations 2.8 to 2.11 present the resolution calculations. In the case of the current channels, an average resolution between all seven current channels is presented.

$$\text{Samples per 50Hz cycle} = \frac{1000\text{Hz}}{50\text{Hz}} = 20 \quad (2.8)$$

$$\begin{aligned} \text{Input voltage resolution} &= \text{Input voltage range} / 2^{\# \text{ of ADC bits}} \\ &= 20 / 2^{12} \\ &= 4.88\text{mV} \end{aligned} \quad (2.9)$$

$$\begin{aligned} \text{Processed voltage resolution} &= \text{Input resolution} \times \text{Voltage rescale value} \\ &= 4.88\text{mV} \times 40.7 \\ &= 198.6\text{mV} \end{aligned} \quad (2.10)$$

$$\begin{aligned} \text{Average processed current resolution} &= \text{Input resolution} \times \text{Average current rescale value} \\ &= 4.88\text{mV} \times 30.9\text{V/A} \\ &= 0.151\text{A} \end{aligned} \quad (2.11)$$

Once calibrated the custom energy monitor was used to measure the energy consumption of a metalworking lathe and pillar drill (see figure 6.3 and 2.9 respectfully), in an effort to review the waveform features and how they relate to the elemental machining operations currently being performed. The recorded profile from the bench-top pillar drill boring a single hole can be seen in figure 2.9.

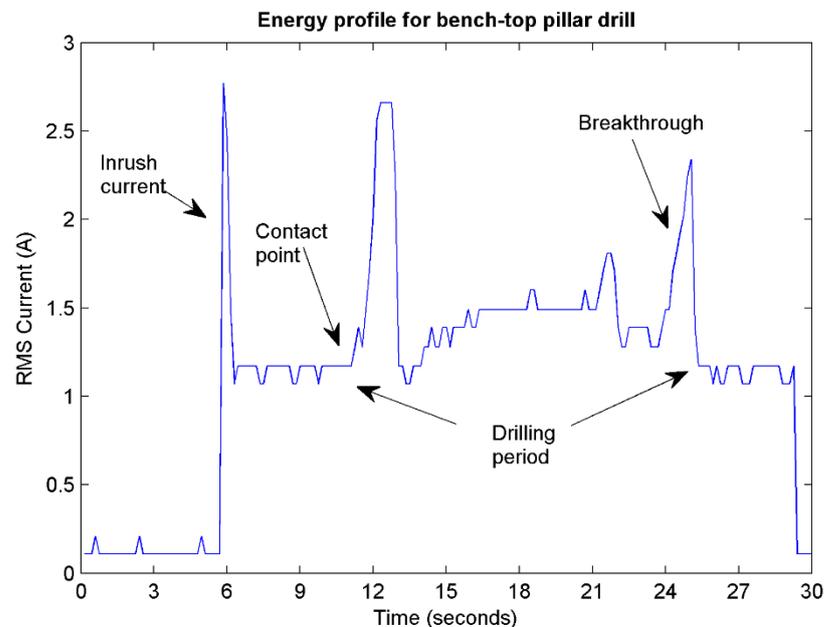


Figure 2.9 - Example of an energy profile for a bench top pillar drill boring a single hole.

The profile in figure 2.9 was seen as a typical energy consumption profile for a manufacturing machine due to the drills primary component – An electric motor. Other reductive manufacturing process machines such as lathes and mills which are heavily used in industry share the same primary component and are ultimately performing the same action as the pillar drill, just along a different axis. Therefore energy profiles from these machines will share similar characteristics and features.

With the described system in operation, the core components of the work could now be developed.

2.4 Summary

This chapter serves as a vital prerequisite for the consideration of energy-related objectives in manufacturing production lines. Not only does it establish the principles of energy monitoring, but it also establishes the current level of monitoring equipment available to manufacturers, and sets the scene for potential optimisation. With monitoring systems recording consumption at a low temporal granularity, the subsequent data will have a low information content. This immediately impedes the usefulness and applicability of any predictions and predictive models generated from that data. As a result of that, the amount of potential energy-based optimisation will be limited. The custom monitoring system attempts to solve this by producing data at a significantly higher granularity. This catalyses a chain reaction which produces higher accuracy models and finally, increases the potential for energy related optimisation. However, the technique used for modelling will also play a vital role, and the modelling methodology may need to be altered in order to take advantage of this new data.

CHAPTER 3

MODELLING MANUFACTURING JOB ENERGY CONSUMPTION

A precursor to any energy-based optimisation strategy is an understanding of how energy is used within the job and how its demand is influenced by the process configuration. Energy modelling in manufacturing is therefore the understanding of how a process and its parameters relate to the energy consumption. This step holds significant importance as the accuracy of the optimisation process is directly dependent upon it – an optimised solution is only as accurate as the models upon which it was based. Additionally, the overall nature and representation of the models must be selected with care as they can influence the model generation time and requirements, the runtime of the optimisation process, and the applicability of the models.

3.1 Energy Modelling Literature Review

A large volume of research concerning energy modelling within manufacturing environments has been carried out, with publications either entirely focusing on it, or developing models to support optimisation strategies. Despite the volume of publications available, the overall diversity of the research is limited. Works by Gutowski et al (2006), Dietmair et al (2009), and Kara and Li (2011) are treated as the progenitors. While they are not the originators of their individual energy modelling paradigms, their work has been heavily referenced and built upon. Gutowski et al (2006) work on modelling energy consumption via the thermodynamic concept of exergy – a measure of the amount of workable energy a component or material has. They propose two equations for calculating the total power of a manufacturing process and the specific electrical exergy per unit of material processed.

$$P = P_0 + k \quad (3.1)$$

$$B_{elect} = \frac{P_0}{\dot{v}} + k \quad (3.2)$$

Where P is the total power in kW required for processing, P_0 is the idle power of the machine in use, \dot{v} is the rate of material processing in cm^3/sec , B_{elect} is the specific electrical exergy per unit of material processed in kJ/cm^3 , and k is a constant with units of kJ/cm^3 .

An analysis of (3.1) and (3.2) shows that Gutowski et al define the total power consumption of a process to be the product of the energy required for material removal k , and the material removal rate \dot{v} , offset from the idling power consumption P_0 . They go on to validate the relationships established in (3.2) by demonstrating that the specific energy consumption - the energy required to process one unit of material, for a range of manufacturing processes is indirectly proportionate to the processing rate. As such, they conclude that process rate is a strong indicator of the specific exergy of the manufacturing process.

Another dominant set of publications is the work by Dietmair and Verl (2009), who introduce an energy model requirement specification. It specifies that a model must:

- Permit generic fitting to production machines and be scalable to appropriately represent the machine at the required level of detail,
- Be accurate to a predefined degree while being easy to parameterise and compute,
- Allow for the considerations of other objectives and be action focused,
- Allow for the prediction of alternative strategies.

They give this specification as they state “*it is virtually impossible to design an optimal product, select the best process chain or ensure maximum utilisation efficiency without having detailed previous knowledge of process and machine related energy consumption*” (Dietmair and Verl, 2009, pp. 63). To address this, they present a state-based modelling strategy. Nine discrete states are defined for a simple milling operation, which can be parameterised from only a few simple empirical energy measurements, combined with the predicted or measured transition times for each state. The end result is an easily adaptable model with low computations requirements. During validation, the model was able to predict the total energy consumption to within 5%.

With this type of modelling strategy, it is easy to determine how the total energy consumption is distributed between the various states and therefore, identify potential areas for energy efficiency improvements. Despite the high accuracy however, energy profile features such as in-rush peaks are not considered substantially important despite the fact that they classify as the peak power usage. This is due to the fact that they only contribute to a small percentage of the total energy consumption. For machines where in-rush currents dominate the energy profile, they state that kinematic, mechanical and motor models can be sufficient to calculate the instantaneous energy consumption. In later work, they expand their modelling criterion to include additional energy carriers such as compressed air, hydraulic fluid, coolant, etc. (Dietmair et al, 2009). While the resulting model can easily be adapted to suit the nature of the manufacturing process, Dietmair et al state that for suitable optimisation, work piece quality, along with machine and tool wear must be considered.

The final progenitor publication is the work by Kara and Li (2011). They initially criticise the work by Gutowski et al (2006), stating that modelling for idle power and specific process energy are lacking definition and quantification. Additionally, the modelling process is infeasible without precise values for the equation variables and coefficients. With all things considered, they doubt the usefulness of theoretical models. To that end they define an empirical modelling approach to statistically model the relationship between specific energy consumption (*SEC*) and the process parameters for a milling operation. Data was gathered from eight different CNC turning and milling machines where the operations were carried out on different materials and with varying process parameters, such as different cutting speeds, feed rates, cut depths, cut widths, and cut environments. Statistical analysis of the data demonstrated an inverse model best describes the relationship between the material removal rate (*MRR*) and the specific energy consumption. They define the following equation, where C_0 and C_1 are machine specific coefficients.

$$SEC = C_0 + \frac{C_1}{MRR} \quad (3.3)$$

It can be seen that (3.3) is very similar to the specific exergy equation, (3.2), specified by Gutowski et al (2006). However Kara and Li state that the coefficient C_1 does not equal the individual machines idle power, as this is highly dependent upon process parameters which are not considered in (3.2). It is shown that the statistical models produced by Kara and Li are highly accurate when applied to new data, with an accuracy of between 91.95% and 97.63%. It should be noted however that this model does not account for the machine's start-up, stand-by and material preparation periods. Kara and Li defend this by stating that the energy consumption of these periods is less than 10% of the total energy consumption. A later expansion of their work focuses on applying their methodology to an injection forming machine (Qureshi et al, 2012). Their result validation shows that the original model for milling and turning processes (3.3) is not entirely suitable and requires slight modification to accurately model an injection moulding process.

As is detailed above, it is seen that the three main energy modelling paradigms are theoretical-based mathematical modelling, experiment-based mathematical modelling, and state-based modelling. The latter is typically used as a vessel for the implementation of the first two but on a more segregated level to increase model accuracy and succinctness. Because of this, treating each paradigm as independent is difficult and there is a lot of commonality between them.

Amongst all the three main paradigms, state-base modelling is most commonly used in the literature. Modern manufacturing machinery can potentially contain a large amount of ancillary equipment which can be active when the machine is not operating. As a result, simply calculating the energy required for machining may not provide an accurate representation of the total process energy. This is verified by Peng et al (2014) who develop a modular modelling approach. Each manufacturing process is modelled as a series of blocks which take in the process parameters and calculates the energy consumption for that particular section of the process. They generate each modular block from empirical data which is segregated based on the manufacturing operations (i.e. milling, drilling, idling, travelling, etc.), and then further segregated into the energy consumptions for individual components such as spindle power,

pumps, etc. This approach allows for a very dynamic modelling environment which is easily scalable and reusable.

The original equation by Gutowski et al (2006), (3.1), is expanded upon by Balogun and Mativenga (2013). They incorporate the energy consumed by the machine's ancillary devices such as pumps, cooling and control units, what they refer to as 'Basic' state energy, with the energy consumed while the machine is idling and preparing to execute a job, the 'Ready' state energy. When incorporated with the 'Cutting' state energy described by Gutowski et al (2006), this can fully model the energy consumption for a manufacturing process. Additionally, they also model the energy consumed during automated tool change and incorporate tool life into the final energy equation. These models were subsequently validated with empirical data from two milling machines with a maximum prediction error of 3% for the total energy demand.

Similarly, the work by Liu et al (2015) segregates a turning process into start-up, idle and cutting states, with each state consisting of their own mathematical models to predict the energy consumption of the main driving system. Thanks to the presence of empirical data, the prediction accuracy for the start-up and idle states is suitably high. However due to its increased complexity, the same cannot be said for the cutting state. While potentially accurate, this is reliant on two coefficients which are dependent upon the particular main driving system and transmission chain. This leads to a low cutting state accuracy. Despite this, the end prediction error for the total energy consumption is found to be 7.8% as the low accuracy 'cutting' state is only active for, on average 30% of the processing time. The accuracy of the other states is not specified.

Similar to state-based modelling, the Discrete Event Simulation model developed by Herrmann et al (2011) allows for highly scalable energy-orientated manufacturing simulations which account for all relevant energy flows. This is done by integrating energy-based objectives with traditional manufacturing considerations, along with realistic energy-based costs. The results from these simulations can subsequently be evaluated to identify areas of potential efficiency improvement. While they produce accurate and promising results in two case examples, a

review of current manufacturing simulation tools indicates that they do not currently support energy considerations.

The work reviewed above focuses primarily on the modelling of milling and turning processes. While these can be considered the most prominent manufacturing operations, other types of process are heavily carried out in different manufacturing industries. Shrivastava et al (2015) model the energy consumption of friction stir welding using the state-based approach. As this process is typically carried out on a milling machine, or a machine with similar kinematics, milling machine efficiency concepts can be used. Empirical data for the energy consumption of elementary operations and machine movements are used to calculate the energy consumption for each stage. For predicting the actual friction stir weld energy, the calculated standby and idle power can be subtracted from the measured power, or the spindle torque measurements can be related to power consumption. Shrivastava et al propose an equation for calculating the latter to determine the total process energy consumption. It should be noted that while they predict the consumption levels for the idle, standby and welding stages, they state that they are only concerned with the energy consumption of the welding stage.

The energy consumption for single point incremental forming process is modelled by Ingarao et al (2014). The process itself is modelled on three different platforms – a CNC mill, a 6-axis robot arm, and a dedicated incremental forming machine tool. The energy consumption for each machine is segregated into different states to reflect the unique methods each machine uses to perform the operation. They conclude that a generalised process specific model is insufficient to accurately represent a process being performed on different machine platforms. An example of this is given by the fact that when performed on a 3-axis mill, the energy consumption is not dependent upon the material and is simply a time multiple of the process energy consumption. However, the model for the 6-axis robot is material sensitive, requiring a custom equation. This demonstrates that energy models need to be machine-specific in order to produce accurate predictions. Peng et al (2014) furthers this by saying that models need to be both machine and process specific. This is indirectly supported by the fact that while the other state-based modelling methods discussed are proven to be accurate to a suitable degree, they are validated

on processes of a very similar nature. Furthermore the applicability and accuracy of their modelling approaches on other manufacturing processes is not considered.

While state-based modelling aims to model the entire energy consumption of the manufacturing job, additional work has also been conducted, which focuses specifically on modelling the energy consumption for the actual process. Based on a review of other modelling methods, Velchev et al (2014) derive an empirical approach to modelling the specific energy consumption of a CNC lathe. Models for tool changing and cutting are devised which consider cutting parameters along with the tool life through the extended Taylor's equation. Model validation shows a 4.85% error in predicted specific energy consumption. Additionally, they review the different modelling approaches detailed by other researchers, including Kara and Li (2011), and Gutowski (2006). This literature review highlights the fact that many researchers have proposed many different equations for calculating the specific energy consumption. In a number of cases these include machine specific coefficients which further reinforce the fact that these models need to be machine-specific.

It is evident that machine and process specific models may be required depending upon the process being executed. This presents a problem as the modelling time can be significant, especially for large-scale and/or diverse production lines. This issue can potentially be elevated with the work by Gilani et al (2013). They implement an automated energy model learning scheme for the identification of energy anomalies. While the focus of their results is the accurate detection of anomalies, they are successfully able to predict process energy consumption based around machine timing signals and empirical energy measurements. Using historical energy and process data, Shin et al (2014) is able to use a back propagation neural network to predict total power consumption. Input data is automatically extracted from G-Code and additional machine control files, and training data sets are constructed based on the theoretical total power, however this can easily be replaced by measured values. While their case study demonstrates the abilities of this form of modelling, generating a suitably large training set may be difficult for some manufacturers. However with the introduction of 'Big Data' concepts, there is a move to save all gathered data for future use and processing.

Additional methods, centring on case-based reasoning for energy consumption estimation have also been proposed. Gong and Ma (2011) specify an energy-based similarity measure which investigates similarities between product name, machining operations, processing times, etc. to determine a set of similar cases associated with an up-and-coming job. Following an analysis of the energy consumptions for these similar cases, they are able to determine a set of influential features and map these to the energy consumption. This mapping can subsequently be used to predict the energy consumption for the new job. When tested, this implementation was able to predict the energy consumption with a maximum error of 7.1%. Their use of historical data is supported by their statement that generating models mathematically is comparatively difficult.

3.2 Energy Modelling Literature Review Conclusions

Research into production line energy modelling has seen increasing interest in recent years; however the overall diversity in the approaches is limited. Theoretical modelling requires machine and/or process specific coefficients which can be difficult or impractical to determine in a modern production line. Despite the high accuracy of the prediction results, most research into energy consumption modelling aims to predict a singular value, such as total energy consumption. While beneficial for determining the energy consumed in producing a product, this does not give an accurate representation as to how the energy consumption varies over the processing time. This fact alone immediately makes these methods incompatible with this project. While state-based approaches aim to model the entire profile from start to finish, they too only refer to fixed levels of energy consumption throughout each state. As figure 2.9 shows, the energy profile for a manufacturing job is highly volatile. Representing each manufacturing state as a single value will result in significant information loss. Furthermore, many manufacturing production machines are fitted with energy management devices such as motor soft-starters, which may produce a dynamic relationship between the process parameters and the energy consumption. This is especially the case for industrial robot arms where the inverse

kinematics problem means the production line manager may not be able to predict how the robot arm will travel from point to point, and how the individual joint motors will be used.

While empirical modelling can be seen as an appropriate alternative, it too may require additional energy monitoring equipment to be installed, and the accuracy and granularity of the final model will be directly related to the monitoring equipment used. Other issues raised by only a few of the researchers (Dietmair et al (2009), Balogun et al(2013) Velchev et al (2014)), is the fact that the energy consumption of a machine will change over time due a varying number of factors, these include:

- Machine age,
- Machine maintenance schedules,
- The life of the machine tool,
- Environmental attributes such as temperature, humidity, etc.

Furthermore, even when regulated, power factor and fluctuations in the supply voltage as a result of multiple machines in operation can influence the energy consumption. Because of these, a fixed model may soon become inaccurate if it does not account for the above considerations. To that end, a new approach was designed.

3.3 Intelligent Historical Library for Manufacturing Energy Prediction

While mathematical based models are able to predict process energy consumption to a high degree of accuracy, the direct use of historical empirical time series energy data is seen to be a more attractive alternative. This is further supported by the following:

- i. Rather than singular values, predictions can be a multi-dimensional time series, making them suitable for determining energy consumption variance.
- ii. With the prevalence of low cost computing storage combined with increasing international and governmental pressures relating to energy efficiency, the recording

and storing of energy consumption data at a high granularity is becoming increasingly prudent for manufacturers.

- iii. Environmental machine-related factors such as machine temperature, machine age, tool wear, etc. along with supply voltage and power factor levels can be easily related to individual profiles permitting analysis into how these influence job energy consumption.

To realise all this, the Intelligent Historical Library for Manufacturing Energy Prediction is proposed, to provide time variant energy predictions. In a production line, time series energy consumption information can be assembled into energy profiles for individual manufacturing jobs. Furthermore machine or production line related factors such as tool wear, machine degradation and supply voltage levels can be calculated / estimated and associated with each profile as metadata. From this, when energy data is required for a prediction, the querying system can provide information regarding the machine conditions. Exact nearest neighbour can then be used to locate the most appropriate historical profile for that particular job, maintaining prediction accuracy over time.

Applicability of the library is maximised by the fact that the historical energy profiles can be independent from one another, allowing for profiles of differing resolutions and granularities to be used in tandem. This is advantageous given the variety of energy monitoring devices employed throughout modern manufacturing lines. The only item which demands compatibility is the metadata.

Figure 3.1 shows a top-level diagram of the proposed intelligent library-based system. Here, the historical energy profile library sources its data from two locations – a collection of actual energy profiles recorded from the production line, and a collection of synthetic energy profiles. The latter is designed to compensate for when actual historical profiles are lacking for a particular job. When this is the case, the sparse historical energy profiles are analysed to determine how the metadata influences the profiles over time. From this, models are created which allows the library to generate synthetic profiles for any metadata values. To ensure efficient generation and use of the synthetic profiles, they are only generated for certain

metadata values, based on the probability of the library being queried for them. The energy models and probable metadata values are then fed into the synthetic profile collection where they are generated prior to being stored. When a query is received, the historical energy profile library utilising exact nearest neighbour, will search the actual and synthetic energy profile libraries, treating them as one conjoined collection. The profile, synthetic or actual, whose metadata lies closest to that of the queries, will be output. These methods are elaborated on in the following sections. Following the discussions in chapter 2, only electrical current is considered as the electrical measure in this particular implementation of the work. However the library is theoretically able to support all electrical parameters discussed in chapter 2.

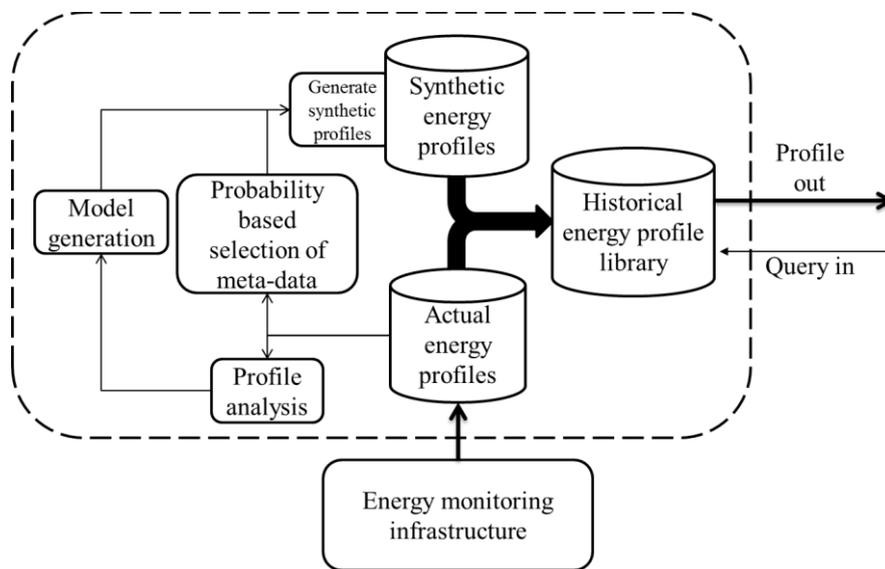


Figure 3.1 - Top-level diagram of the intelligent library-based system.

3.3.1 Profile Analysis

For the library-based system, assuming that the historical profiles have been collected accurately, the accuracy of the predictions will be dependent upon two main elements:

- The overall distribution of the profiles in the context of their metadata, relative to the acceptable range of machine and tool conditions,
- The level of variability within the library’s individual profiles over the acceptable range of conditions.

The two aforementioned points can be considered dependent upon the individual jobs and the machines on which they are executed on. As a generalised rule, the more profiles per job are collected, the more accurate the prediction will be. Unfortunately this directly relates to the frequency of the jobs execution. For more frequently executed jobs, their libraries will be more highly populated, with a smaller average distance between metadata values. Thereby the prediction is likely to be more accurate than a job which is executed less often. To compensate for large gaps within the historical profiles metadata range, synthetic energy profiles could be generated to ‘fill the gaps’. These synthetic profiles could be used along with the empirical profiles temporarily, until being replaced with a newly recorded empirical profile whose metadata values are within a suitable predefined Euclidean range.

Generating these synthetic profiles necessitates the analysis of how the metadata influences the energy profiles. To garner an accurate understanding of this for each machine in the production line ideally requires two elements:

- i. The rate of change for each metadata factor.
- ii. The influence periods of each factor – This is the temporal window for when the influence factor is acting upon the energy profile. An example of this could be the influence from an ageing spindle motor which is only active for set periods of the jobs execution.

With these two elements, it is possible to generate synthetic profiles for any combination of metadata values. However, in reality knowing these beforehand is unlikely. As a result, these elements will have to be discovered retrospectively.

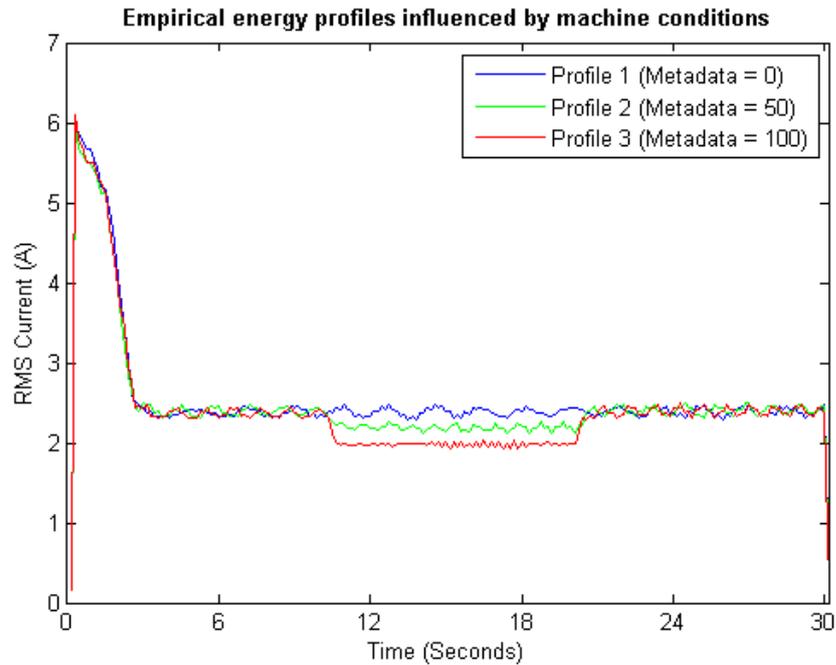


Figure 3.2 - Graph showing the metadata influence on empirical energy profiles.

Figure 3.2 presents three empirical profiles recorded from a dust extraction system using the custom seven channel energy monitor. Each profile has been influenced by the machine conditions. It is presumed that prior to this the various profiles will have been aligned with respect to time via methods such as cross-correlation, least squares, etc. The initial step to analysing the metadata's influence is to determine the influence periods. It is assumed in this work that for each individual influence period, the level of change in the energy consumption will be consistent. For example, a motor with poorly lubricated bearings may result in additional amperage being drawn while it is in use. However this current offset will be consistent so long as the motors input parameters such as speed or torque are maintained. Any changes to these will be identified as a different influence period.

Distinguishing the influence periods in a set of historical profiles is achieved by maximising the difference between the profiles. This, by consequence, maximises the difference between the influence periods allowing their boundary points to be identified, and the data to be segregated. The final stage is to statistically compare these segregated groups to determine if they are similar, and therefore are part of the same influence period. This methodology is described in further detail in the following steps, where *Hist* represents a set of historical profiles.

Step 1

Calculate the time series difference between every combination of available recorded profiles in *Hist*, using (3.4). Exclude mirrored combinations. See figure 3.3.

$$\Delta profile_{xy}(t) = profile_x(t) - profile_y(t) \quad \text{where } x \neq y, profile_x \& profile_y \in Hist \quad (3.4)$$

Step 2

Determine the cross-profile variance of all available $\Delta profile$, by calculating the variance of every t data point in each $\Delta profile$ using (3.5), where $\Delta profiles$ is the collection of $\Delta profile$. See figure 3.4.

$$Cross\sigma^2(t) = \frac{1}{N-1} \sum_{xy} \left(\Delta profile_{xy}(t) - \overline{\Delta profiles(t)} \right)^2 \quad \text{where } N = (|Hist|-1) \times |Hist| \quad (3.5)$$

Step 3

Filter $Cross\sigma^2$ using a median filter to reduce large fluctuations. See figure 3.4.

Step 4

Compute the moving difference of $Cross\sigma^2$ to reveal magnitude changes using (3.6), and normalise the results between 0 and 1 to produce $\Delta Cross\sigma^2_{Normalised}$. See figure 3.5.

$$\Delta Cross\sigma^2(t) = Cross\sigma^2(t-1) - Cross\sigma^2(t) \quad (3.6)$$

Step 5

Apply a thresholding operation to identify large step changes as the boundary points of the influence periods, (3.7).

$$t = P_i^{Influence} \quad \text{where } \Delta Cross\sigma^2_{Normalised}(t) > 0.6 \quad (3.7)$$

Step 6

Locate the $\Delta profile$ with the largest numerical range $\Delta profile^{largest}$, and divide it into influence groups based on the boundary points specified in $P^{Influence}$, using (3.8).

$$influence_group_i(t) = \Delta profile^{largest}(t) \quad \text{where } P_{i-1}^{Influence} \leq t < P_i^{Influence} \quad (3.8)$$

Step 7

Equally sub-divide each influence group into three sub-groups. Consider the middle sub-group as the most representative of the influence group as this is not affected by transitions from one influence group to another. See figure 3.6.

Step 8

With the most representative sub-group of each influence group, utilise the Mann-Whitney U test to identify statistically similar groups. Classify these respective influence groups as belonging to the same influence period.

Step 9

Utilising the Mann-Whitney U test again, compare the most representative sub-group of each influence group with a zero array to identify whether an influence period is affected by metadata.

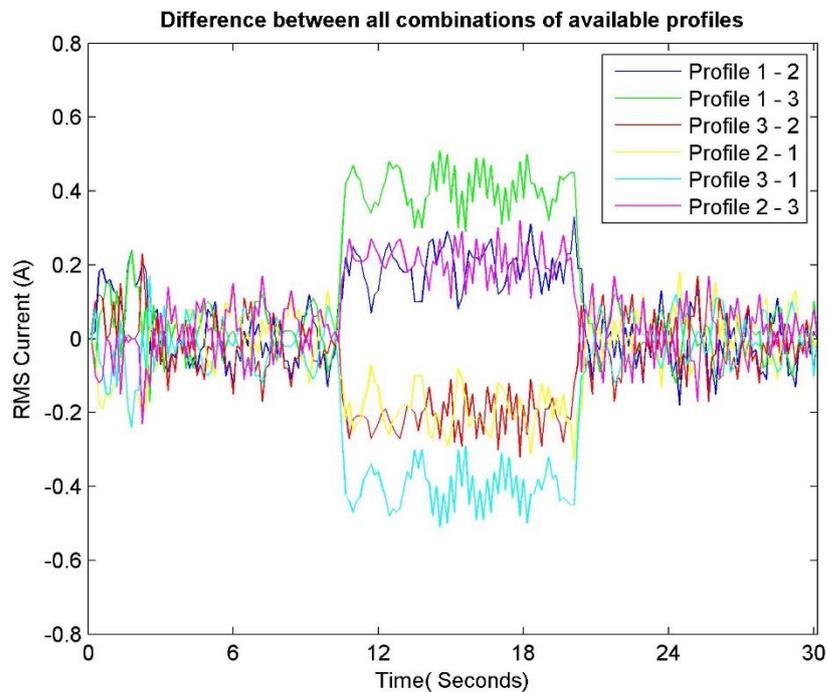


Figure 3.3 - Time series differences between all possible combinations of the three available energy profiles.

Figure 3.3 demonstrates the initial step of the above methodology, in which the three historical profiles are compared against each other in all possible combinations. The purpose of this is to maximise the variance in data points for all discrete time values, as this is used to distinguish the step changes.

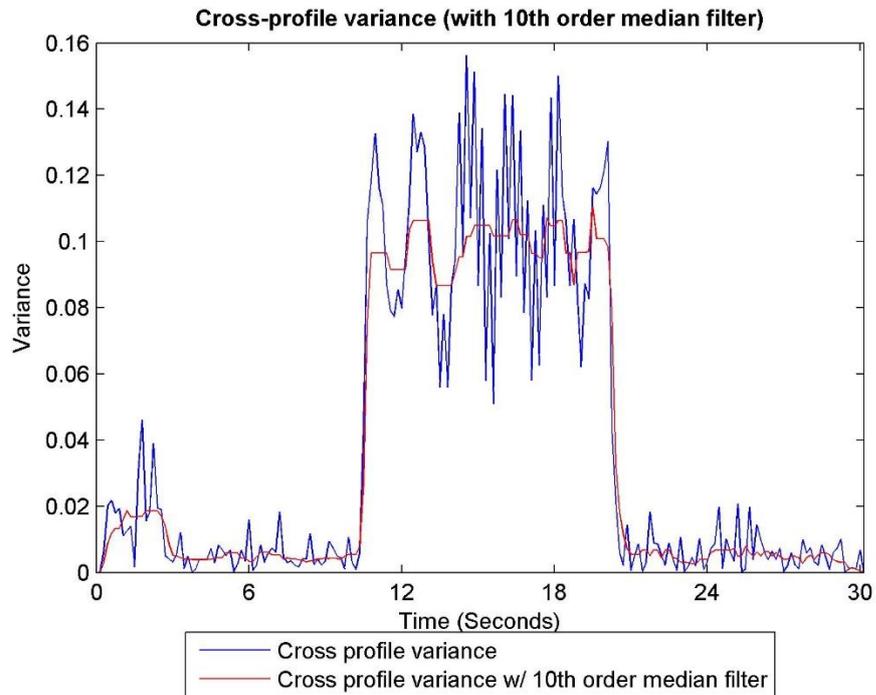


Figure 3.4 - Cross profile variance.

In figure 3.4, a 10th order median filter is applied to reduce the large fluctuations and increase the step distinction as seen in figure 3.5.

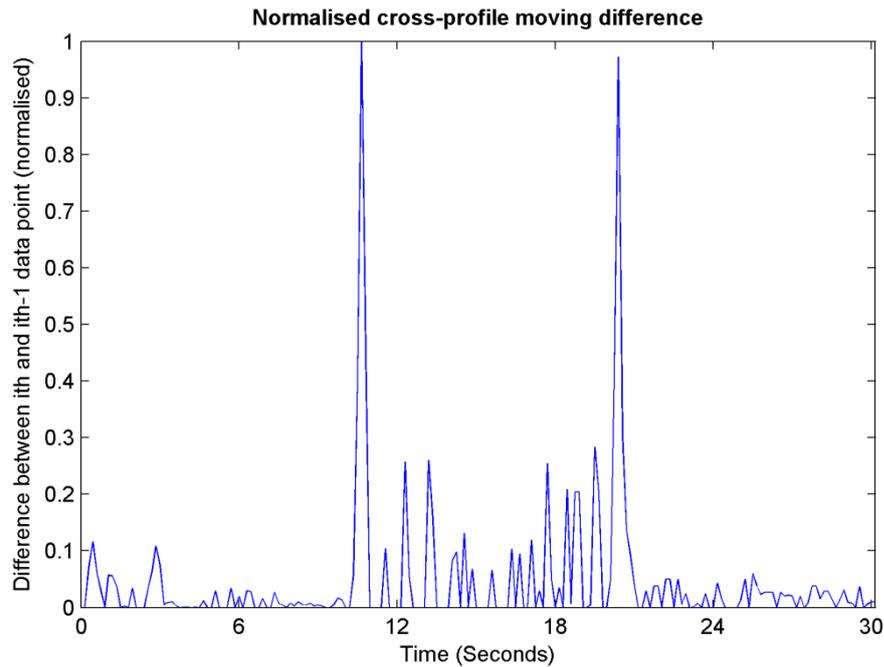


Figure 3.5 - Normalised cross-profile moving difference.

The information provided by figure 3.5 and extracted in accordance with step 5 of the method, allows for the historical profiles to be divided into a precursor form of the influence periods, known as influence groups. See figure 3.6. Depending on the dataset under analysis, the threshold limit may need to be adjusted in order for step 5 to return the correct result. The final two tasks involve compiling the influence periods by identifying influence groups which are statistically similar, and disregarding influence periods which are not actually influenced by the metadata. This is all performed using a $\Delta profile$ which shows the profile changes as a result of the influence. To maximise the contrast between none similar influence groups, the $\Delta profile$ with the largest numerical range is used. One issue with comparing the influence groups is that from a process timing viewpoint, the time when a particular influence takes effect may differ slightly between profiles. This will be the case, especially when the process involves a human operator or some form of human intervention. An example of this can be seen in both figures 3.2 and 3.3, where the changes in amplitude do not coincide at precisely the same time. As a result, the step detection will not be 100% accurate for all profiles, potentially being several data points out. To compensate and ensure an accurate comparison, only the middle third of each influence group is considered during this stage as figure 3.6 shows.

Comparison of the different groups is performed by the Mann-Whitney U test. As the statistical distribution of the data is unknown, this particular null hypothesis test was selected due to its nonparametric nature (Corder and Foreman, 2014, pp. 1 – 3, 69 – 80). The central sub-groups of each influence group are tested, and common groups are classified as belonging to the same influence period. The results of the Mann-Whitney test can be found in table 3.1.

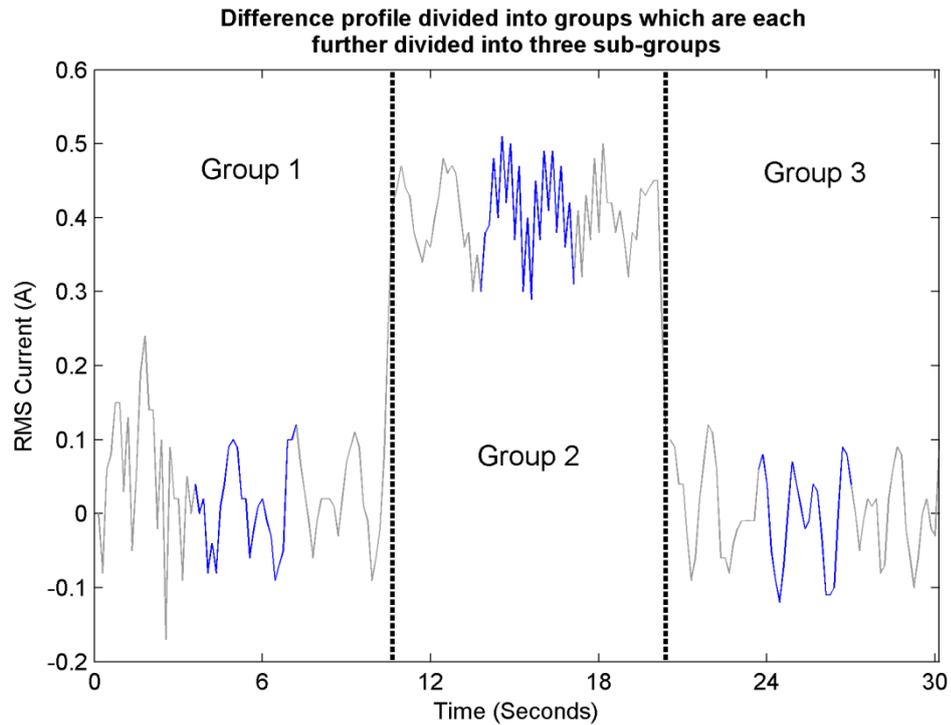


Figure 3.6 - A difference profile segregated into three groups based on the findings from figure 3.5. Each group is further subdivided and the outer sub-groups (grey) are discarded.

Table 3.1 - Results of Mann-Whitney test to determine common groups.

Group combination	Mann-Whitney test result
Group 1 & 2	0 (rejected)
Group 1 & 3	0.3852
Group 2 & 3	0 (rejected)

The results shown in table 3.1 demonstrate that the Mann – Whitney test has successfully identified groups one and three as statistically similar, while distinguishing group two as different. This is evident by the fact that the null hypothesis is rejected whenever group two is considered. Therefore, the system will identify groups one and three as belonging to the same

influence period, while group 2 belongs to its own influence period. Figure 3.7 shows these influence periods.

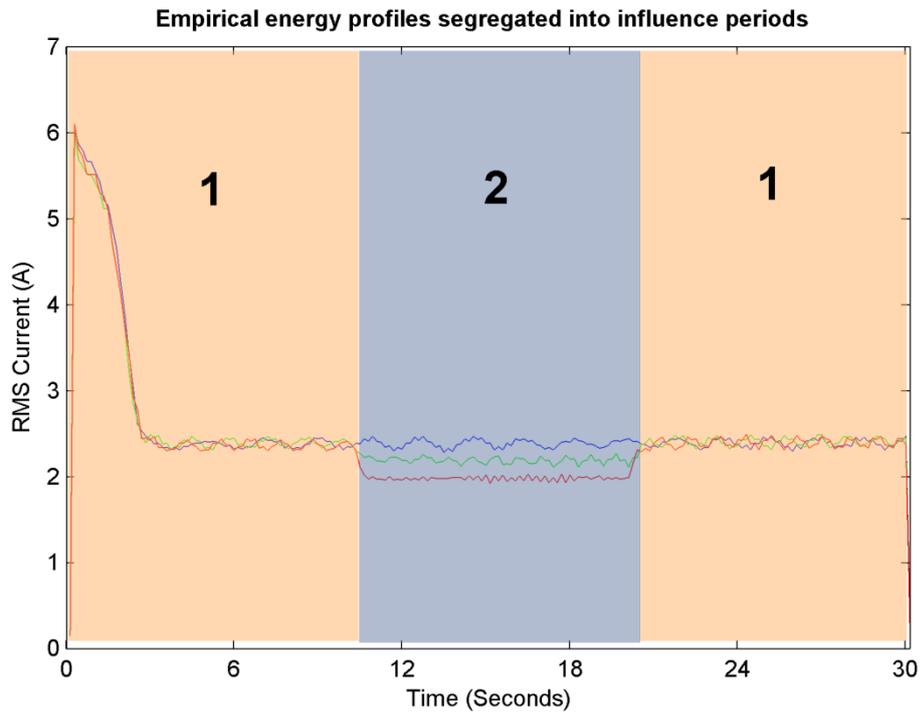


Figure 3.7 - Recorded energy profiles segregated into influence periods.

The final step is to identify influence periods which are not actually affected by machine conditions. The Mann-Whitney U test is again utilised for this by evaluating the sub-grouped data (as seen in figure 3.6) against a zero array of equal length. If a group is not influenced by any metadata, then in the difference profile ($\Delta profile$), the data within that group should centre on zero. Table 3.2 shows the results, where the test has rejected influence period two but not influence period one, indicating that this is not influenced by the metadata. As a result, influence period one is discarded from further evaluation.

Table 3.2 - Results from the Mann-Whitney test to determine if either of the two influence periods is not actually influenced by a metadata variable.

Influence period	Mann-Whitney test result
1	0.3
2	0 (rejected)

At this stage in the analysis, the methodology has identified the boundaries for the influence periods, and determined which are influenced by the profiles metadata and which can be

discarded. Using this information, any energy profile for the same job can easily be segregated into a set of influence periods. The above analysis requires a minimum of two profiles in order to complete, however it can be generalised that the accuracy of the method is related to the total number of profiles used. While all three profiles were used in this method, in a case where the number of profiles results in computational difficulties, a subset of profiles can be used.

The next stage, as discussed in the following section, is to understand how the metadata affects the influence periods. To allow for a thorough evaluation, necessitated the need for a large set of profiles for the same job but recorded with different metadata. Within the scope of this project, these could not be sourced empirically. Therefore artificial profiles were generated based on the empirical profiles and process metadata influence characteristics like those seen in figure 3.2. A progenitor profile was originally generated (figure 3.8), representing the artificial profiles without any metadata influence. A series of 100 profiles were generated from this, being adjusted by two randomly generated metadata values – A and B. A low level of random noise was also applied to each profile. To maintain consistency with the profiles recorded using the custom energy monitoring system, all artificial profiles were generated at a granularity of 150ms. These profiles were then input into the above methodology to identify the influence periods. Figure 3.9 shows three of the 100 profiles segregated into influence periods with (3.9) denoting the equations used to alter them from the progenitor.

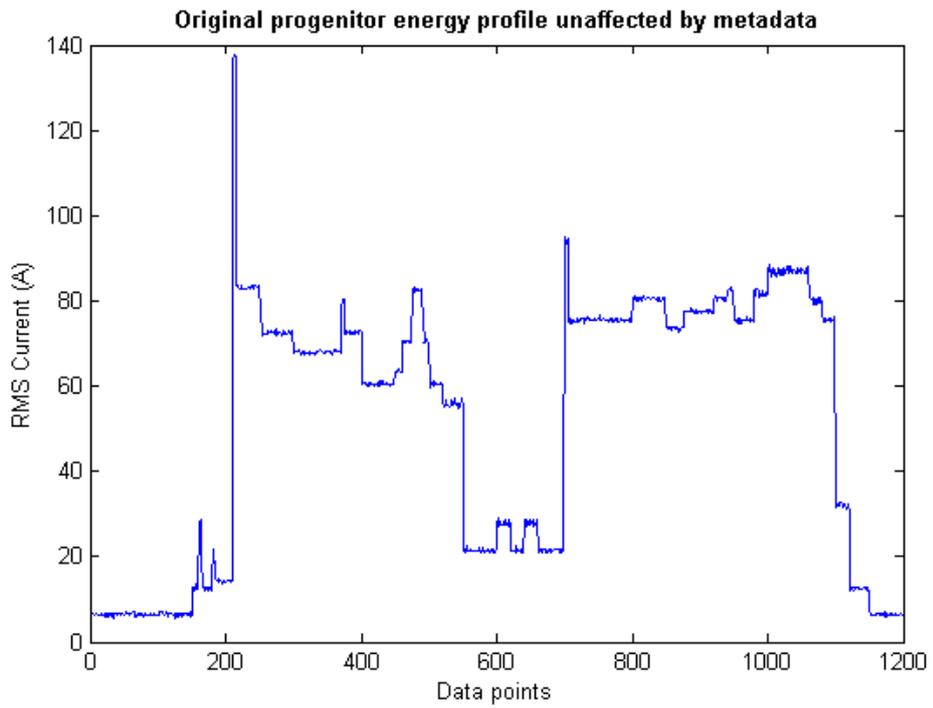


Figure 3.8 - Original progenitor profile on which all artificial profiles are based.

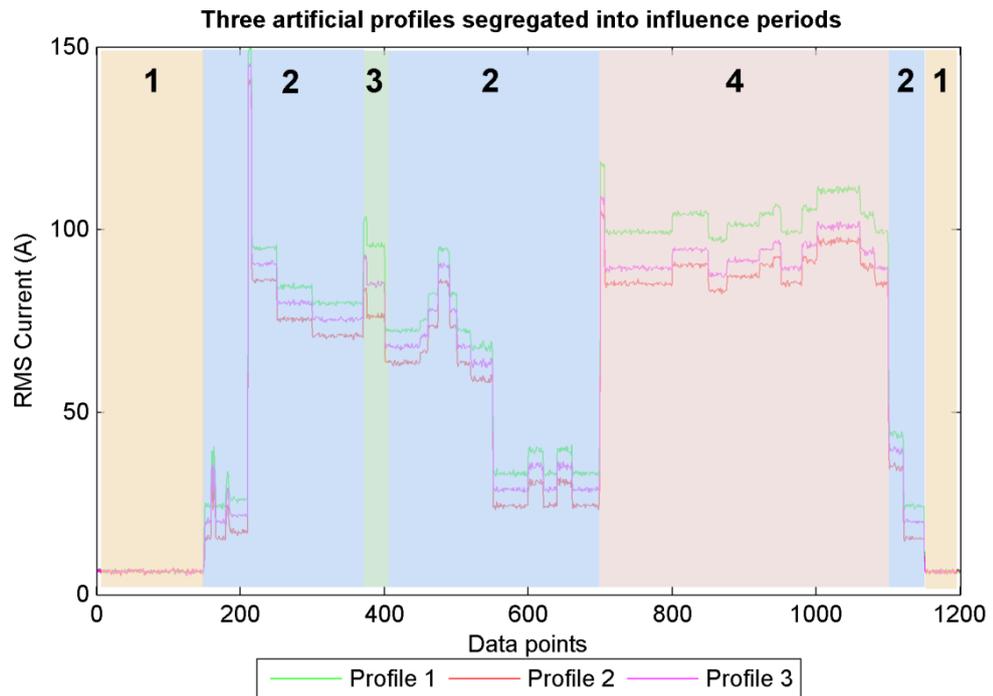


Figure 3.9 - Energy profiles segregated into influence periods. Period 1 is unaffected by any metadata.

$$\begin{array}{l|l}
Progenitor(t) & Influence\ period\ 1\ effect \\
Progenitor(t) + (MetaDataA \times 1.48) & Influence\ period\ 2\ effect \\
Progenitor(t) + (MetaDataA^{1.32} \times 1.48) & Influence\ period\ 3\ effect \\
Progenitor(t) + (MetaDataA \times 1.48) + (MetaDataB \times 2.17) & Influence\ period\ 4\ effect
\end{array} \quad (3.9)$$

The artificial profiles contain four influence periods, with period one being discarded for not being influenced by the metadata values. The remaining three periods have either a linear or exponential influence on the progenitor. This provides a sufficient spectrum of different rates of change to evaluate the next stage of the synthetic profile generation – the model generation.

The data from the 100 artificial profiles will be used throughout the remainder of this chapter.

3.3.2 Model Generation

As the rate of change for each metadata influence value is assumed to be consistent throughout an influence period, it is only necessary to consider a single data point for each influence period i , denoted by k_{ip_i} . These particular points can be selected at random; however they must remain consistent throughout all historical profiles. As there is the potential for an anomalous reading to be selected, an alternative would be to take a number of neighbouring samples from each influence period and take the average.

As the nature of the rate of change is unknown, a regression-based approach was taken for the purpose of modelling the relationship between the metadata values and the value of the common data point for each influence period. As the aim of this methodology is to generate synthetic profiles when historical ones are sparse, the algorithm selected must be able to work well with limited training data. Using their implementation in Weka 3, twelve machine learning algorithms were evaluated with an emphasis on ensemble learners (Weka 3: Data Mining Software in Java, 2015). In these, multiple models are trained on subsets of the training data, with their individual results combined to produce a single output. In cases where the training data is small or difficult to learn, ensemble learners have shown to produce more accurate predictions compared to standalone classifiers, as the final result is derived from multiple

models, each an expert in a specific partition of the decision space (Dietterich, 2000; Polikar, 2006).

All algorithms were trained on a series of four data sets generated from the 100 artificial profiles discussed in the previous section. These contained influence information on five, ten, twenty and 100 profiles. To maximise prediction accuracy, a model was generated for each influence period, meaning that a training set would be required for each training set size and influence period. For each of these, each instance contained the amplitude of the randomly selected data k_{ip_i} , along with the profiles metadata. It would be the task of the model to determine a relationship between these data points and the metadata.

The twelve algorithms evaluated are briefly introduced below:

Multilayer Perceptron Neural Network (MLP)

A neural network, trained using backpropagation which contains one or more hidden layers with weighted interconnections (Fausett, 1993). The implementation used in this work utilised a single hidden layer with linear activation functions for each neuron, resulting in the network performing linear regression. The number of hidden neurons is given by (3.10) (Boger and Guterman, 1997, pp. 3031). Experimentation demonstrated that when given the largest training set of 100 samples, the learner's accuracy benefitted from a small learning rate. However the learning rate had little effect on the accuracy when given the remaining smaller training sets. To this end, a learning rate of 0.05 was selected. The remaining parameters were kept at the default values as provided by Weka 3. A full listing of these parameters along with their values can be found in appendix B.

$$\text{Number of hidden neurons} = \lceil \text{Number of attributes} \times 0.7 \rceil \quad (3.10)$$

Support Vector Regression (SVR)

Support Vector Regression is an extension of the Support Vector Machine classifier (SVM). In these, to categorise new instances, SVM generates a plane to intersect the data, such that the distance between the plane and the multiple classes is maximised. The key ability of SVM is

that in cases where the data is not linearly separable, it can be remapped into a higher dimensional feature space (Law, 2011). This is achieved through the use of a kernel function, and allows the SVM to linearly separate the data in feature space using a hyperplane. The extension of Support Vector Regression (SVR) permits linear regression, and non-linear regression through the use of non-linear kernel mapping (Gunn, 1998). Two different kernel functions were trialled during this experiment – the polynomial kernel and the Pearson VII universal kernel, to determine which gave the best results. This implementation uses the improved Sequential Minimal Optimisation algorithm proposed by Shevade et al (2000), for training the SVM. In both cases, all parameters were left at their default values. These can be seen in appendix B.

Linear Regression (LR)

This performs linear regression analysis to generate a ‘line of best fit’ for a given data set. This algorithm was selected to establish a comparative baseline with the other algorithms. As before, parameters remained at their default values and can be found in appendix B.

Ensemble Learner – Bagging

In ensemble learning, multiple occurrences of a standalone classifier, known as a base learner, are trained on a subset of the original training data. This subset is constructed for each occurrence of the base learner through sampling the original training data with replacement. The base learners are then said to become experts in a particular field of the data space. Finally, the individual results from each base learner are combined to produce a single output. It is this combination method which defines the ensemble learner. In Bagging, also known as Bootstrapped Aggregating, the individual results, when a numeric output is required, are averaged (Brown, 2010).

In this work, each of the standalone learning algorithms discussed were trialled as base learners, with the number of base learners remaining at the default value of ten. In the case of MLP, it was found that when used as a base learner, the overall accuracy did not benefit from a learning rate smaller or larger than that of the default – 0.3.

Ensemble Learner – Stacking

Stacking, also known as Stacked Generalisation combines the individual base learner results using another learning algorithm, referred to as a meta-learner. The principle behind this is to have two stages of learning algorithms. The first consists of base learners which each learn a specific subset of the dataset. Their outputs are then fed into the meta-learner which learns the mapping between the base learners output and the expected output. The output of the meta-learner is the final result of the ensemble (Polikar, 2006).

In this implementation, the same algorithm was used for both the base-learners and the meta-learner. As with bagging, ten iterations of the same base learner algorithm were used.

Ten-fold cross-validation was used for a majority of the training processes. The only times when this was not used was when the size of the dataset disallowed it. This was the case with training sets containing five and ten profiles. Due to the size of the data set, three and five-fold cross-validation was used respectively. In the cases where the five profile training sets were applied to the bagging algorithm, where SVR with a polynomial kernel was used as the base learners, the algorithm failed to train due to the lack of diversity within the training subsets. As a consequence, the training set was used directly with no folding. Post training, all models were further tested on an independent data set of three new profiles which were not part of the 100 originally generated.

Each of the twelve algorithms was tested on the multiple data sets – the training sets and the independent test set, with the mean absolute and RMS errors from both being used to determine which algorithm performed best. For each type of error, each combination of data set and algorithm produced a collection of three errors – one for each influence period, denoted by ε_{ip} . These were subsequently combined using (3.11) where $|ip_i|$ and $|IP|$ represent the number of data points within influence period i , and the number of influence periods in a profile respectively.

$$\varepsilon' = \sum_{i=\varepsilon}^{|IP|} \left(\varepsilon_{ip_i} \times |ip_i| / \sum_{i=\varepsilon}^{|IP|} |ip_i| \right) / |IP| \quad (3.11)$$

Equation (3.11) produces an average error while accounting for the relative lengths of each influence period. Thereby a period which is difficult to model, but only lasts for a short time relative to the length of the profile will not have a significant adverse effect on the overall result. Both the raw RMS and mean absolute errors were processed via (3.11). These results can be seen in appendix C.

Finally, the processed RMS and mean absolute errors for each training set size were averaged to produce a single result set for each algorithm. These can be seen in table 3.3 with the minimum values indicated in bold.

Table 3.3 - Processed results from different algorithms trialled to model metadata influence on energy profiles.

Algorithm	Training set		Test set	
	Mean absolute error (A rms)	RMS error (A rms)	Mean absolute error (A rms)	RMS error (A rms)
Multilayer Perceptron	0.636	0.963	0.521	0.609
SVR w/ Polynomial kernel	0.482	0.787	0.285	0.332
SVR w/ Pearson kernel	0.832	1.187	0.972	1.335
Linear Regression	0.934	0.894	0.388	0.461
Bagging w/ Multilayer Perceptron	0.568	0.861	0.545	0.649
Bagging w/ SVR w/ Polynomial kernel	0.522	0.884	0.406	0.465
Bagging w/ SVR w/ Pearson kernel	0.948	1.369	1.283	1.657
Bagging w/ Linear Regression	0.950	1.136	0.372	0.442
Stacking w/ Multilayer Perceptron	1.186	1.374	1.371	1.627
Stacking w/ SVR w/ Polynomial kernel	0.925	1.205	1.109	0.507
Stacking w/ SVR w/ Pearson kernel	1.232	1.570	1.252	1.601
Stacking w/ Linear Regression	1.112	1.598	1.160	1.493

Overall, the results in table 3.3 show that Support Vector Regression (SVR) with the polynomial kernel is a clear winner, producing the least erroneous results in all cases, with the bagging variant coming a close second. While ensemble learners were anticipated to perform better, it is likely that as the training focused on only using small data sets, there was simply not

enough diversity between the training sub-sets which may have lead the models to suffer from overfitting to a degree.

Generating a complete synthetic profile, $profile_{syn}$, from only a single data point for each influence period is achieved using (3.12), where for each influence period i , the difference between the models prediction mp_{ip} and the value of the known data point at the centre of the influence period in the historical profiles, k_{ip_i} is calculated. This forms an offset which is applied to a single historical energy profile which acts as a reference template, $profile_{ref}$. The offset is applied consistently to the reference profile during the relevant influence period's time interval. Areas of the synthetic profile which are not affected by the metadata are set equal to the reference profile. For influence periods which are distributed throughout the profile (such as period 2 in figure 3.9), they can share mp_{ip_i} and k_{ip_i} values.

$$profile_{syn}(t) = \begin{cases} profile_{ref}(t) + (mp_{ip_i} - k_{ip_i}) & : P_i^{Influence} \leq t \leq P_{i+1}^{Influence} \\ profile_{ref}(t) & \text{otherwise} \end{cases} \quad (3.12)$$

Using SVR with a polynomial kernel algorithm, figure 3.10 shows a comparison between a second independent test profile and prediction results from different sized training sets, with error values shown in table 3.4.

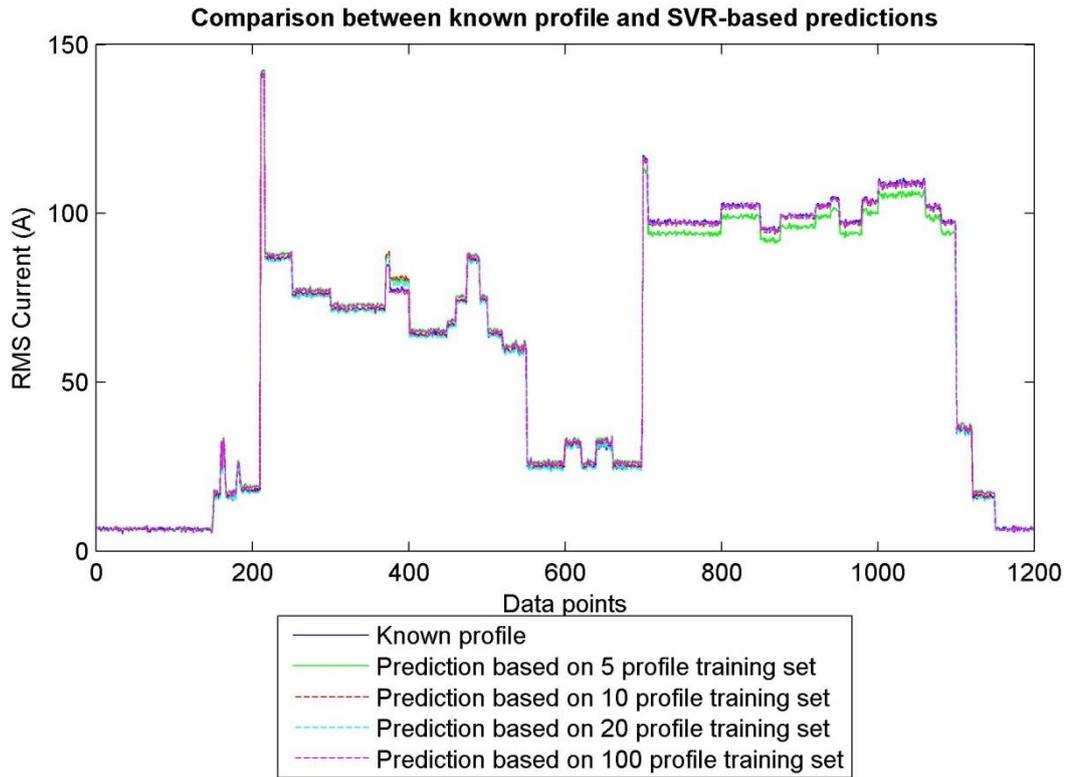


Figure 3.10 - Comparison between a known profile and predictions based on different sized training sets.

Table 3.4 - Error values for predicted profiles shown in figure 3.10.

Training set size (# of Profiles)	Mean error (A RMS)	Max error (A RMS)	RMS error (A RMS)
5	1.93	4.43	2.30
10	0.53	4.31	0.79
20	0.49	2.68	0.62
100	0.70	2.02	0.85

Table 3.4 demonstrates that SVR with the polynomial kernel can successfully be used to predict the rate of change for individual influence periods. As expected, the accuracy is proportionate to the training set size however this does decrease for the mean and RMS errors with the largest training set. This is likely due to the larger training set introducing more variance in the data. Overall the error is limited.

It is important to note that the above methodology does have certain limitations in that it cannot account for a metadata influence which results in an influence period occurring at a different point in time. As explained in the previous section, the profile analysis is able to compensate for the period transitions not aligning perfectly, however this is limited. This could be rectified by

introducing an initial step which compensates for influence period changes, with an additional model used to determine the relationship between the metadata and the influence period occurrence in the profile.

Finally, in the case of a newly introduced job where no historical data is available, depending on the nature of the job itself, it may be possible to utilise historical data from another job with similar operations. Furthermore, depending on the training error, it may be prudent to forgo generating synthetic profiles until a jobs training set increases past a predefined point. The error from poorly trained models may be in excess of the error from directly using the few empirical profiles.

3.3.3 Probability based Selection of Metadata

At this stage, for a particular job, models are now available to generate synthetic profiles at any value of metadata. Two issues remain however – how many synthetic profiles are needed and where should they be generated in metadata space? Generating too few or placing them in the wrong regions of the metadata space will limit their usefulness, while generating too many will hamper computational efficiency. To maintain high accuracy, every time a new empirical profile is received for a particular job, its models can be retrained and the synthetic profiles regenerated. This further enforces the need to generate the right amount. Synthetic profiles are therefore unevenly distributed throughout metadata space, based on the probability of that region of space being hit by a profile query request. To achieve this, metadata space is subdivided into equal sized plots according to (3.13). These plots can be considered hypercubes in metadata space.

$$\text{Number of plots for dimension } i = \left\lceil \frac{\overset{\text{Range}}{\text{MetaData}[i]}}{0.5 \times \overset{\sigma}{\text{MetaData}[i]}} \right\rceil \quad (3.13)$$

Equation (3.13) accounts for the metadata's standard deviation $\overset{\sigma}{\text{MetaData}[i]}$, in each dimension i , to maintain a suitable resolution during the subdivision. By default, $\overset{\text{Range}}{\text{MetaData}[i]}$ specifies the

allowed operating range of the associated values machine condition (e.g. tool wear operates between 100% and 0%). If this is not known, the numerical range of the metadata can be used. A discrete frequency distribution is then calculated, with figure 3.11 showing an example with ten sets of randomly generated two dimensional metadata. A probability distribution can then be generated from that. At this stage, the number of synthetic profiles assigned to each plot y , is based on the query hit probability $P_{Hit}(y)$ and is determined according to table 3.5. As before, the number of synthetic profiles is relative to the metadata's largest integer single dimension standard deviation, denoted by $MaxSTD$. To ensure plots are not overloaded with synthetic profiles, the number of empirical profiles in a plot, $Emp(y)$ are accounted for.

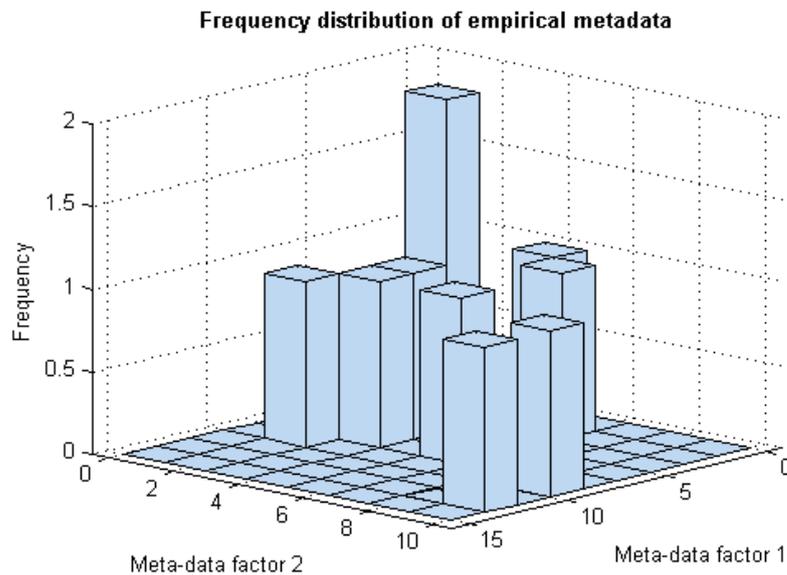


Figure 3.11 - Frequency distribution of empirical metadata.

Table 3.5 - Rules for generating synthetic profiles for each metadata space plot.

Probability per plot	Synthetic profile assignment
$P_{Hit}(y) = 0$	Number of synthetic profiles is equal to 1
$0 < P_{Hit}(y) < 0.2$	Number of synthetic profiles is equal to $\max\{0, \lceil (MaxSTD/2) - Emp(y) \rceil\}$
$P_{Hit}(y) \geq 0.5$	Number of synthetic profiles is equal to $\max\{0, \lceil MaxSTD - Emp(y) \rceil\}$

The rules in table 3.5 are designed to favour plots with a moderate probability of being queried, as it is here that the synthetic profiles can be most effective. Plots with a higher or lower

probability receive a reduced amount of synthetic profiles. As before, considering the metadata's standard deviation ensures that the number of profiles, be they synthetic or empirical, are relative to the plot size. Figure 3.12 shows how the frequency distribution of profiles in metadata space is altered after the rules have been applied.

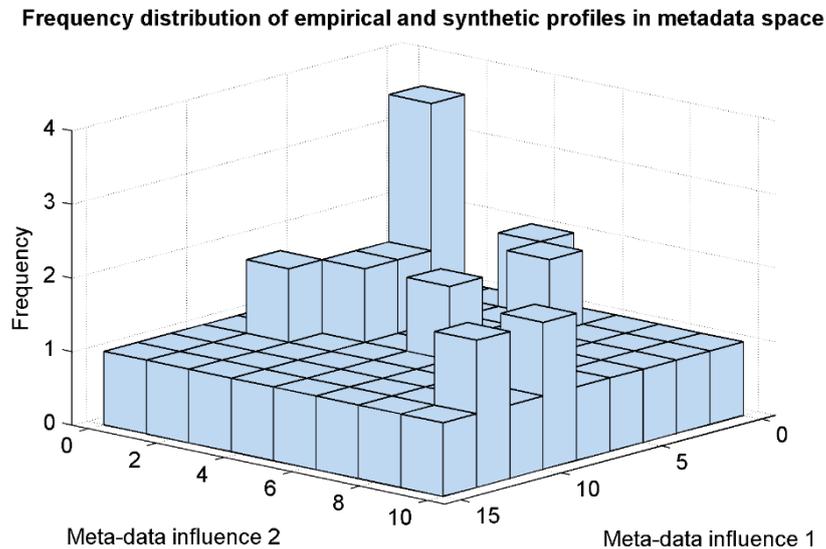


Figure 3.12 - Frequency distribution of energy consumption profiles after synthetic profile generation rules have been applied.

Once each plot has been assigned a number of synthetic profiles, these can be generated by randomly selecting metadata values within each plot.

Should there be large groups of plots with zero hit probability that lie far from plots with a probability higher than zero, it may be beneficial to merge neighbouring plots prior to applying the rules in table 3.5. Not only will this reduce profile generation time, but as these plots lie away from any empirical data, their synthetic profile prediction accuracy is likely to be reduced. In figure 3.13, a probability distribution is initially generated from the frequency distribution in figure 3.11. For the merging process, a perimeter is set around every plot with a hit probability above zero. The plots which are merged are those which do not lie within any of these perimeters. All this can be seen in figure 3.13 where plots are merged in groups of two, resulting in an 11% drop in the number of plots. For larger metadata spaces, this rule could operate iteratively; with incrementally larger numbers of plots merging the further away they are from a plot with a hit probability above zero.

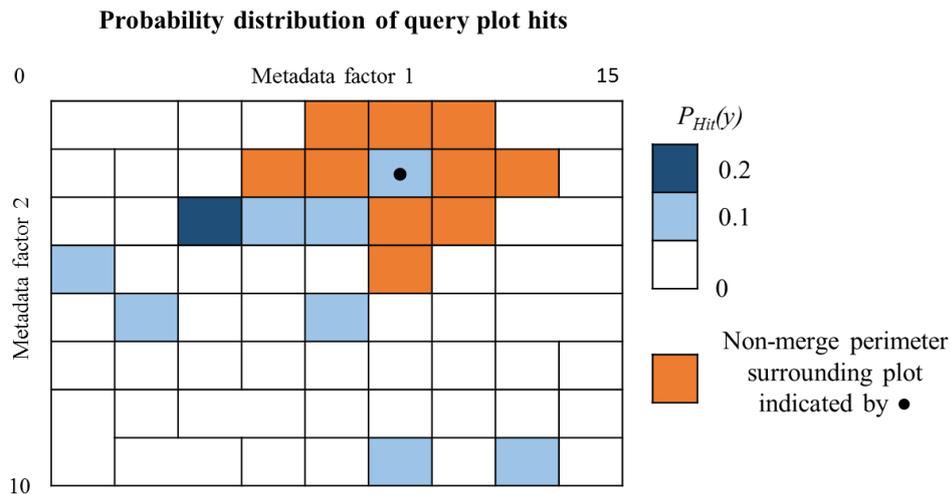


Figure 3.13 - Example of metadata space with merged plots.

Using a combination of the two systems described in both this and the previous sections, would allow for an efficient and accurate historical energy profile library to be used to compensate for machine-related changes in job energy profiles.

3.3.4 Exact Nearest Neighbour Search Algorithm

The final component of the intelligent library-based system is the algorithm which will actually locate which profile to return when the system is queried. Each query will consist of the job whose energy profile is to be predicted, and a set of machine conditions predicted for when the job is to run. The system can then search the library of actual and synthetic profiles for a profile which, a) belongs to the correct job, and b) has the smallest Euclidean distance in metadata space. As the former is solved by simply looking in the relevant jobs library, the problem can be simplified to a linear nearest neighbour search. Four separate exact nearest neighbour search algorithms were evaluated – Linear Search, Kd-Tree, Metric Trees and Cover Trees. With a supplied training set, a linear search simply calculates the distance between the test instance and every training instance, returning the one with the smallest distance. Contrariwise, Kd, Metric and Cover Tree methods all invest in partitioning the training set so the search space can be quickly reduced (Witten and Freank, 2005, Ch. 4.7). For Kd trees and Metric trees, also known as ball trees, the search space is partitioned into hyper-rectangles and hyperspheres respectively.

Individual partitions are continuously subdivided until the number of contained points reaches a certain threshold. Cover trees also use hyperspheres but utilise them in a multi-layered approach. In this, a single root hypersphere contains all the data points which are sub-partitioned into further inner hyperspheres. Multiple further hyperspheres are generated inside the inner hyperspheres; concentrating around clusters of data points until the innermost hyperspheres each contain a single data point. When a query is received, it has its own hypersphere generated around it whose radius is decreased based on the closest hypersphere centroid in the current level. The advantage of this method is the system only needs to calculate the Euclidean distance between the query hypersphere centroid and each hyperspheres centroid at each level, rather than for every data point in the search space.

Each algorithm was experimentally evaluated using their respective implementations available in the (Weka 3: Data Mining Software, 2015). Three and ten dimensional training sets of 15000 instances, along with test sets with 500, 10000 and 15000 instances were used. All this data was randomly generated with fixed ranges applied to each dimension. The results can be seen in table 3.6.

Table 3.6 - Average search times for Exact Nearest Neighbour Search Algorithms on test sets. (Results averaged over 100 consecutive runs. Generated on an Intel i7 1.7 GHz computer with 4GB of memory running Windows 7).

	Training set size	Linear search (ns)	Metric Trees (ns)	Cover Trees (ns)	Kd-Trees (ns)
Three dimensional data	500 instances	355561	98239	332632	39228
	10000 instances	1312109	111708	459705	61752
	15000 instances	1879338	108757	316407	45116
Ten dimensional data	500 instances	489545	1305668	532154	112433
	10000 instances	1656181	1897516	1897516	397989
	15000 instances	2441153	1952869	2855997	342352

The results presented in table 3.6 show that in every case, Kd-Trees have a significantly reduced search time compared with the other methods. These results are supported by the findings of Kibriya and Frank (2007) who conduct a similar survey. It should be noted that in terms of dimensional scalability, it is the linear search that outperforms all the tree search based methods.

From an applications viewpoint, table 3.6 shows that the algorithm of choice will be dependent upon the average dimensional size of the energy profile metadata. Domain expert knowledge

will be required in the feature selection process to ensure a sensible number of attributes are used. In the interest of scalability and overall universality with different sized metadata, a linear search algorithm was employed in this work, with a Euclidean distance function used to determine the exact nearest neighbour.

3.4 Summary

This chapter introduces the concepts and current techniques for energy modelling in manufacturing environments, and addresses their limitations. To compensate for these limitations, an intelligent library-based system is developed. This employs multiple historical profiles along with associated metadata to attribute the profiles to a particular state the machine was in at the time of recording, in terms of mechanical and environmental factors. As a consequence, the most appropriate profile can be selected for the prediction process should the machine and environmental factors be known or predicted beforehand. To further compensate when there is insufficient historical data, the intelligent library is able to generate synthetic profiles based on the analysis of the available empirical data and intelligently distribute them within the metadata space, based on the libraries' query hit probability.

Altogether, this presents a highly suitable prediction system for energy consumption within manufacturing environments.

CHAPTER 4

SCHEDULE OPTIMISATION SYSTEM – BACKGROUND

In the context of manufacturing, scheduling is the process of allocating limited resources to a set of manufacturing jobs over a specified time period, with the goal of optimising one or more objectives (Pinedo, 2012, pp. 1; Karger et al, 2009). Typically, these resources consist of the specialised machines and workforce within the production line, however the finite delivery rate of energy must also be considered for efficient usage. The schedulers typically used in manufacturing are normally designed to minimise the makespan of the process. This attempts to maximise job execution parallelisation, which will increase both the instantaneous energy consumption, and the overall variance in the energy consumption. With this in mind, several research focuses have been developed in the field to consider energy related objectives along with traditional time-based ones when generating schedules, with the goal to either reduce overall energy consumption, reduce peak energy consumption, or produce the products in a more energy efficient manner. These are introduced in section 4.1. Following this, an overview of the solution to the proposed schedule optimisation problem is introduced in sections 4.2 and 4.3, with a detailed discussion into algorithm selection in section 4.4.

4.1 Review of Energy Optimisation in Manufacturing

For reducing overall energy consumption in a manufacturing line, various methods have been proposed, which can all be generalised into two distinct but relatable categories – machine-centric approaches and production line centric approaches. The former concentrates on optimising the operation of individual machines, considering them in isolation to each other. The latter attempts the same but in the context of the entire production line. A review of both approaches can be found below.

4.1.1 Machine Focused Approaches

The traditional approach for energy optimisation is to operate and maintain energy efficient equipment. This has been applied to practically all products from manufacturing equipment to household white goods. However purchasing new equipment for a manufacturing production line can be impractical for financial reasons. A suitable approach is the optimisation of idle time energy consumption. When a machine is between jobs, it is typically left idling instead of being shut down or placed in an energy-saving mode (Mouzon et al, 2007). Certain equipment can be costly in terms of energy and time to turn back on again. Therefore a major question is when is it efficient to turn off a machine or place it in an energy saving mode? Current methods for improving this include turning off a machine's ancillary components or components which can be rapidly turned off and on, or placing them in an energy saving mode (Orio et al, 2013). This can be reinforced by the use of industrial energy management networks such as SERCOS Energy and Siemens PROFIenergy, which analyse energy consumption of machines and the time it takes for them to transition between operational states so that they can be shut down or placed into an energy saving mode during predictable downtime such as lunch times and holidays. However they are unable to take advantage of short-term non-executing periods in the production schedule.

Methods for overcoming this have been proposed by Di Orio et al (2013). Using the state-based modelling approach developed by Kara and Li (2011), which naturally identifies periods of non-execution, they develop an architecture for a Self-Learning Production System (SLPS). This assesses the current context of the production line and predicts when a machine will next be required to run based on the mining and machine learning of historical data. The output is a set of schedule alterations, suggesting when to shut down a machine or place it in an energy saving mode. Experiment results show that an implementation of the system results in an approximate 7% reduction in overall energy consumption while maintaining the level of machine availability. However due to its reliance on historical data, two months of training were required before these levels of reliability were attained. Similar research is discussed by Eperspächera and Veral (2013). They represent a machine as a graph of discrete operational states which

includes shutdown and energy saving modes, with the transitions between states represented with an energy and time cost. With this representation, locating the shortest path through the graph from the start to the end node subsequently produces the machine schedule with the lowest energy consumption. Once this path is generated, it is checked to determine if it complies with the schedule time constraints.

Mouzon et al (2007) contribute by proposing several dispatch rules aimed at minimising energy consumption through intelligent machine shutdown. Knowing when the next job is to arrive is of significant importance with this type of solution. When the release dates are unknown, the problem becomes NP-hard. Mouzon et al attempt to solve this problem by calculating an average idling time. A ten job schedule is tested on nine different dispatch rules, which are based on whether there is scheduled idle time and what prior information is known about the job release dates. While each of the dispatch rules was successfully able to reduce total energy consumption, they each resulted in a slight increase in production time. While they state that this may be suitable depending on the constraints, they investigate ways to enhance their results through the use of a multi-objective linear mixed integer program which additionally considers minimising production time. The production line manager is then able to determine which objective takes priority depending on the current production circumstances. A similar choice is available with implementation by Yildirim and Mouzon (2012). They use a Genetic Algorithm to determine candidate solutions, with a linear program used to determine the completion time and energy consumption. This produces solutions along a pareto front allowing the production line manager to select the most appropriate one.

One of the issues with a generalised system for intelligent machine control is the fact that the available energy saving states may be machine or supplier dependent. One machine may have several different energy saving modes while another may not have any. Shrouf et al (2014) address this issue by devising three common states a machine can be in – processing, idle and shutdown. With this they develop a model for determining when a job should start, when a machine should idle and when it should be shut down. To solve this model, they perform a comparative study using a Genetic Algorithm and a traditional analytical solver. Of the 13 cases

tested, the Genetic Algorithm provided a stable computational time which is not significantly influenced by the number of jobs or the time frame of the schedule. With the analytical solver however, the runtime increased exponentially as more jobs were considered. These results help to demonstrate that heuristic and non-deterministic approaches are best suited for solving these of problems.

In all the above cases, the strategies focus on a single machine. The work by Pach et al (2014) aims to consider multiple machines, deciding when to shutdown each one based on potential fields. With the goal to reduce both energy consumption and makespan, their system allows both jobs and production line equipment to “attract” one another via potential fields. Each machine is thus able to detect the jobs it needs to execute, and is able to decide how to execute the jobs while optimising its energy consumption. In a simulation and empirical experiments, results show that the total energy consumption was reduced by 19%, however they state that this will reduce as the number of jobs increases. While these are promising results, during the practical experiments, products had to be fitted with a Wi-Fi connection in order to implement the potential field. As such, depending on the production line, this may significantly hinder a practical implementation.

In the case of an automotive assembly line, Chen et al (2012) models sequential production lines as Bernoulli serial lines. Using Markov Chains to analyse the production line’s performance, they are able to determine the effect of machine shutdowns and start-ups on the overall manufacturing performance and the energy consumption. Unlike those detailed above where the energy consumption minimisation is second to the minimisation of production time, their strategy presents a trade-off between energy consumption and production throughput. Through the use of a greedy algorithm, two solutions are returned - one aims to produce the lowest average energy consumption per production output, while the other aims to minimise energy consumption while remaining as close to the original schedule constraints as possible.

Reducing the energy consumption of idling machines is advantageous within the concepts of reducing peak energy consumption and its overall variance. By reducing the energy consumption of non-executing machines, the average energy consumption for the entire

production line, along with the amplitude of energy peaks will be reduced. While these systems show promising results, they are only able to influence the energy consumption when the machine is not being used.

4.1.2 Schedule Focused Approaches

The idea of intelligently shutting down a machine is expanded on by Zononi et al (2014), whose system incorporates this into the scheduling of two serial machines. Their results demonstrate that the potential energy savings are heavily dependent on how the machines are governed in relation to the intelligent shutdown. When a machine is abruptly restarted to keep up with demand, the level of optimisation is reduced. However as different machines will have different start-up energy requirements, it is unsure how reliable these results are over the spectrum of manufacturing equipment. Weinert et al (2011) discuss a similar approach but do not confine it to a simple two machine problem. Utilising an energy modelling strategy similar to the one developed by Peng et al (2014), they develop a scheduling system which is able to account for energy consumption through the use of 'Energy Blocks'. Each block represents a possible machine state and contains the energy consumption for that state. Using these, the scheduler is able to construct a predicted energy consumption profile for the schedule. During testing, energy consumption was significantly improved when the workload was balanced between the machines. This is presumably due to the fact that machine idle time is reduced. These improvements can be furthered by shifting the starting times of jobs, and adding an additional machine to prevent bottlenecks in the production line.

Due to its obvious financial incentives, there has been keen interest in optimising schedules to reduce energy-based costs by scheduling jobs to run during times of low energy tariffs. Emec et al (2013) investigates this by defining three strategies for energy-aware scheduling, each with differing levels of job start time flexibility for the purpose of reducing energy costs. In testing, they show that maximising the flexibility of each job start time results in an optimal schedule for avoiding peak energy costs. Furthering their research, they also demonstrate that when

varying the speed and acceleration of robotic arms, its influence on energy consumption is non-linear, with the minimum energy consumption not located at either of the jobs processing time boundaries.

During energy-optimised scheduling, maximising the flexibility of the job constraints has a direct influence on the optimality of the result. Based on the fact that different machines can have different energy consumptions despite performing the same job, He et al (2012) develop an energy-oriented scheduling strategy to select the best job-resource assignment for minimising total energy consumption. Thanks to the limited number of assignment options in their investigated three job schedule, a deterministic dispatch rule is used based on a first-come-first-serve basis. As has come to be expected, while the range of schedules have a variation of 27.6% in the total energy consumption, they also have a 29.5% variation in processing time.

Depending on the nature of the production line and the manufacturing operations to be completed, the flexibility when devising a schedule can differ dramatically. In the cases discussed above, maximum flexibility is achieved by relaxing the job start times and job-resource assignments. This thus allows an energy-orientated scheduler to produce more optimal results but simultaneously increases the difficulty of the problem. In certain manufacturing environments, the energy delivered to a job can be adjusted over time. In their investigation of a foundry, Artigues et al (2009, 2013) they propose a generalisation of the cumulative scheduling problem - the 'Energy Scheduling Problem'. This aims to produce a schedule which specifies both the start time of jobs, and the input power to the related furnace such that energy costs and peak power are minimised by maintaining it under a specified limit. An iterative two-step approach is initially used with a constraint programming optimiser used to determine the job start times and resource assignments, followed by a mixed integer linear programming model to solve the power settings. While effective, this method does not guarantee to find the optimal solution. It is later extended to include 'energetic reasoning' – a tree search algorithm capable of early detection of infeasible solutions for cumulative scheduling problems Berthold et al (2011). During testing the implementation is quickly able to filter out job lists to which no solution is

available, however of the twenty job lists considered, only twelve were successfully solved while the others exceeded their allotted computational time.

In parallel with aligning jobs with lower energy tariffs, there is also work in reducing peak energy consumption to avoid heavy fines from energy suppliers. Traditional methods for this include installing naive power cutters which cut power to equipment if the production lines energy consumption approaches the consumption limit (Artigues et al, 2013). This method can be very unfavourable due to its negative influence on the production rate and it can only be applied to selected equipment. More advanced methods include load shifting which is introduced in section 4.2.

It is stated by Fang et al (2011) that generating schedules optimised for minimal makespan is computationally difficult. Therefore adding additional objectives serves to make the problem more challenging. Using a Mixed Integer Linear Program, they investigate generating a schedule optimised for makespan, peak energy consumption and carbon footprint, where both job start times and process parameters can be changed to a limited degree. When applied using a commercial solver, the optimal solution could not be found after 24 hours of processing. Their method for overcoming this is to relax the constraints to reduce the complexity of the model, however the results were heavily dependent upon the specifics of the production line such as inter-job storage. While effective results are eventually produced, the authors state that this particular method may not scale well, resulting in prohibitive computational times for industrial-sized problems.

In the proposition by Brozzone et al (2012), a schedule with minimal makespan is generated using an Advanced Planning and Scheduling system before being further optimised by an Energy-Aware Scheduler to minimise energy consumption peaks. Modelled as a Mixed Integer Programming model, the manufacturer specifies a maximum allowable peak power and the schedule is modified while increases to makespan and tardiness are discouraged. Experimentation demonstrates that as a maximum peak power ceiling is introduced, both the makespan and computational time increase until the point where a suitable schedule cannot be determined within the allotted time window. They conclude that commercial solvers cannot

always be relied upon to generate suitable results in sufficient time. When replaced with a Randomised Neighbourhood Search algorithm, a near-optimal solution was attained with an increase in makespan. An improvement to this methodology is offered by Xu et al (2014), who consider the energy-related objectives during the initial generation of the schedule instead of separately. Unlike Bruzzone's implementation, this allows them to locate the optimal solution with a reduced makespan and tardiness, along with improved computational efficiency.

In order for a schedule to be optimised for minimal peak energy consumption it is necessary to keep track of jobs executing in parallel (Fang et al, 2013). Fang et al consider this with two comparative scheduling approaches – a disjunctive and an Assignment and Positional formations, for a flow shop scheduling problem. Both approaches are tested using a commercial optimiser. Interestingly, in their implementation the computational time peaks as the user specified energy ceiling reaches the mid-range of the tested spectrum. Ultimately, results show that the Assignment and Positional formulation performs much faster and scales well with the number of jobs. Further improvements to this led to a reduction in computational time, most prominent in the mid-range of the energy ceiling spectrum. However, as is becoming true in a number of implementation strategies, computing a near-optimal result in a reasonable amount of time becomes difficult as the number of machines or jobs increases.

Currently, the only known commercial scheduling software which considers energy related objectives is the Energy in Production Planning and Scheduling module by Transfact (E-PPS, No date). In parallel with the standard manufacturing considerations, the E-PPS software which is implemented as an add-on module to a traditional production planning system, is able to actively reduce energy consumption through intelligent scheduling and organisation of the production process (Pechmann et al, 2012). However, to achieve this, the software must have maximum visibility of the production line in addition to substantial prior knowledge of the production line and its energy consumption. Traditional backwards scheduling is used followed by adjustments to the individual job start times, resource assignments and job pre-emption. During a test implementation at a semiconductor manufacturing site, the E-PPS successfully resulted in a 5% reduction in both total and peak energy consumption. However its success is

dependent upon the flexibility of the production line and the scheduling constraints (Pechmann and Schöler, 2011).

Thus far, the discussed techniques have primarily focused on the optimisation of electrical energy usage. While this can be considered the most popular energy medium used in manufacturing, depending on the sector, different energy mediums are heavily used. In cases where energy is locally converted into different forms, the optimal use of these energy converters can result in significant savings. This is the work by Rager et al (2015), who examines the prospect of optimising a schedule for steam demand from on-site boilers. A Genetic Algorithm and two memetic algorithms with differing local search mechanisms are trialed to produce the optimal result. While they state that no search heuristic algorithm presents a clear computational advantage, the Genetic Algorithm is seen to produce suitable results in a large or not-well-understood search space. Using a method similar to the one used by Artigues et al (2013), a dominance rule is used to assign jobs to machines. This is then optimised for steam demand by one of the three heuristic algorithms, with the final results compared to those from a commercial solver. Unfortunately as the solver was only able to produce results for simple test problems, comparative analysis was limited. Regardless, all three heuristics were able to significantly improve the original result produced by the solver with little variation between the three results.

4.2 Machine or Schedule Centred Approach?

As discussed in the research aims, this project's aim is to minimise the energy consumption variance in a production line, resulting in a more distributed utilisation of the power infrastructure over time. Currently, despite an extensive literature review, no similar research has been discovered. As a result, deciding the generalised solution for the discussed problem cannot draw on exact previous findings. Ultimately, two generalised approaches were identified in the section 4.1 literature review - adjust the process parameters such that the individual job energy profiles have minimal variance, or modify the job start times so that the production lines

energy profile has a minimal variance. While a large amount of research has been dedicated to the former, it is believed that the overall contribution to energy consumption variance is limited in comparison with the latter strategy. This is due to the fact that regardless of the optimisation, the machine is still required to execute the individual manufacturing operations. This will result in a variance in the energy consumption as the machine transitions between operations and states. Further reductions in consumption variance can only be realised through modification of the actual manufacturing machine. This can include motor soft-starters and other machine-level energy management systems. Further limitations come from the fact that certain processes may only be able to operate within a limited parameter range or the parameters may need to be fixed. Additionally, this strategy does not immediately consider how the production lines energy consumption variance will be affected by concurrent manufacturing jobs. The concept of globally optimising all manufacturing jobs without considering process parameters can be considered much more appealing as it considers the variance produced by concurrent jobs, while allowing them to run with the parameters preferred by the manufacturer. This strategy is conceptually similar to load shifting, a traditional energy management method which in a manufacturing environment involves moving job start times to periods of lower energy tariffs, and ensure that their instantaneous energy consumption does not risk breaching the agreed supply limit (Brown et al, 2012).

4.3 Proposed Approach

The literature review in section 4.1 demonstrates that introducing energy considerations into production scheduling can produce promising results while minimising the negative effects to the traditional scheduling objectives. There is also significant support for this optimisation approach in section 4.1, where a number of researchers (Yildirim and Mouzon, 2012; Mouzon et al, 2007; Xu et al, 2014; Rager et al, 2015; Emec et al, 2013) detailed the advantages of solely modifying job start time to minimise energy consumption peaks. To that end, this research project presents a production schedule optimisation system which aims to minimise the

variance in production line energy consumption through the manipulation of job start times. To ensure conformity with traditional objectives such as deadline compliance, a schedule is initially generated using a traditional production line planning software. From a production line manager's viewpoint, this means the pre-optimised schedule will already have a suitably optimal job-machine assignment, along with confirmed job / process release dates and deadlines preassigned by a trusted piece of software. Analysis of production schedules optimised for minimal makespan shows that they are typically backwards scheduled, meaning they are devised retrospectively from the deadline. While this is used to minimise the holding time of completed products or sub-assemblies, it also allows for a degree of manoeuvrability in the job start times even in schedules optimised for minimal makespan. Once this original schedule has been generated, it can be optimised for minimal energy consumption variance. This chapter concludes with an explanation into the key algorithm employed to optimise the schedule. Following this, the next chapter introduces the individual components of the optimisation system and its overall implementation before its performance is analysed and improved upon in chapter 6.

4.4 Selection of Heuristic Optimisation Algorithm

The literature review reveals, as expected, that the selection of the main optimisation algorithm has a significant influence on the optimality of the final result, the computational performance and the solution scalability. One of the dominant issues found in performing schedule optimisation is the computational difficulty. Many researchers (Shrouf et al, 2014; Fang et al, 2011; Rager et al, 2015; Brozzone et al, 2012) have investigated the use of commercial solvers to produce either the optimised result, or a comparative result. The details of these solvers are not revealed, nor is it specified if they are deterministic or not. In the results, while these solvers perform particularly well for computationally simple problems, their scalability is poor. For the research project introduced here, the size of the search space, which contains all possible solutions allowed by the constraints, can be calculated as the sum of discrete time steps T , a set

of jobs J can possibly begin at, within their constraints (4.1). With $d_{u,i}$ and $r_{u,i}$ respectively representing the deadline and release date of process u , of which job i is a member.

$$\text{Number of possible combinations} = \prod_0^{\|J\|-1} [(d_{u,i} - r_{u,i}) \times T] \quad (4.1)$$

Equation 4.1 establishes that as the number of jobs in the schedule increases, the number of possible combinations increases exponentially. As such the problem can soon become impractically large for a polynomial time algorithm to solve in an acceptable amount of time (Fang et al, 2011; Shrouf et al, 2014; Brozzone et al, 2011). To this end, it was decided that a search heuristic would be used for the optimisation. Genetic Algorithms are a popular choice in the literature review in section 4.1, where they have stable computational times for a range of problem sizes (Yildirim and Mouzon, 2012; Shourf et al, 2014, Rager et al, 2015). They are also prominent in current traditional scheduling approaches (Pinedo, 2012, ch. 14.4). In a review of literature by Filho et al (2012), Genetic Algorithms have been applied to solve a large number of Flexible Manufacturing System scheduling problems, either on their own or through a hybrid algorithm technique. More specifically they have been successful in solving a range of specific scheduling problems (Qiu et al, 2009; Bierwirth and Mattfeld, 1999), which can include multiple project scheduling (Gonçalves et al, 2008). It is important to remember that Genetic Algorithms should not be considered as a “one size fits all” answer. As the search space increases, the error relative to the best known solution can increase as documented by Bierwirth and Mattfeld (1999), whose work investigates the use of Genetic Algorithms in job scheduling. However it should be noted that in their experiments, in a schedule containing 2000 manufacturing operations, the error was only 7.59% from the best known solution. Depending on the manufacturer, this level of error may be acceptable when compared with the benefits of a reduced computational time.

Like other heuristic search algorithms such as Simulated Annealing, Evolutionary Strategies, and Tabu Search etc., a Genetic Algorithm can be described as a general purpose search algorithm which can, and has been, applied to a number of different applications (Ailva and Falcão, 2008). As such, it can be concluded that there is no clear advantage over using any one

algorithm. That said, with its prevalent use in the previous literature review, Genetic Algorithms were chosen for the main optimisation algorithm.

A Genetic Algorithm itself is a population based algorithm where a population of candidate solutions to the problem in question, known as ‘chromosomes’, are evolved over a number of iterations until a termination condition is reached (Mitchell, 1998, pp. 8). This condition can be when a suitable solution is found, or when a predefined amount of time or iterations have occurred. In each iteration, each candidate solution chromosome is evaluated to determine its “fitness” or optimality against one or more objectives. Natural selection is then emulated with fitter chromosomes having a higher probability of surviving and thriving as the population is evolved through the use of genetic operators. Like biological evolution, this all aims to produce a suitably optimal result without having to consider every entity in the search space. In the context of this particular problem, each candidate solution chromosome is a representation of a potential production schedule. Each chromosome contains a fixed length sequence of ‘genes’, which represent the job start times. Because the aim of the algorithm is to find the schedule which produces the minimum variance in the production line’s energy consumption, each candidate schedule chromosome’s fitness value will be its predicted energy consumption variance. The Genetic Algorithm is therefore tasked with locating a represented schedule with the lowest fitness value.

4.5 Summary

The literature review in section 4.1 demonstrates that while significant research has been devoted to energy-based optimisation strategies, no research has been uncovered with a focus on minimising energy consumption variance. This subsequently reinforces the originality of the research. However as a consequence, the generalised approach to solving this problem along with the main algorithm selection has been decided based on findings from other energy based optimisation research.

CHAPTER 5

SCHEDULE OPTIMISATION SYSTEM – IMPLEMENTATION

Following from the algorithm selection in section 4.4, this chapter discusses how the various components of the schedule optimisation system come together. Along with the fundamental operators native to the Genetic Algorithm which are modified to suit the application, additional functions are also introduced which aid in its integration into the overall optimisation system and to improve the optimality of the final result.

5.1 Heuristic Algorithm Encoding Scheme

In order for the Genetic Algorithm to solve the optimisation problem, a suitable encoding scheme must be selected to represent potential solutions in a form which is both independent of the specific problem and computationally efficient. The choice of scheme is critical due to its heavy influence on the overall accuracy and computational time (Karaboga, 2000, Ch. 1.1.2). In this implementation, the Genetic Algorithm will be required to optimise every job start time in the production line schedule. Therefore each candidate schedule considered by the Genetic Algorithm should represent the start times for every job. In the production schedule these will likely be presented as a form of date time stamp, however it can be seen as inefficient for the Genetic Algorithm to consider them in their natural form. Computationally a date time stamp will likely be processed and stored as a composite data type; consisting of several primitive types which each store a singular time component such as hours, minutes, etc.

To reduce the size of the problem space, the Genetic Algorithm will only be able to reassign a job's start time to a finite number of discrete time points separated by a constant time period T . Therefore each job's start time could easily be represented by the number of time steps offset from a common constant reference point in time. Such constant reference points include the

release dates for each process in the schedule. To provide a single point of commonality, the earliest release date of any process s_e , is used as the lower bound. Similarly, the upper bound can be assigned as the latest process deadline D_{Max} . This method also permits time based constraints to be encoded, allowing for computationally quick checking of constraint compliance. The encoding (5.1) and decoding (5.2) equations for converting a jobs start time timestamp s_i between the time and chromosome representation g_i are as such:

$$\text{Encode from time to chromosome domain: } g_i = (s_i - s_e) / T \quad (5.1)$$

$$\text{Decode from chromosome to time domain: } s_i = s_e + (g_i \times T) \quad (5.2)$$

These equations can also be used for encoding/decoding additional time based information. This encoding strategy, known as real-value encoding, allows for the encoding of a potential solution while minimising the differences between it and the actual production schedule it represents (Herrera et al, 1998). An example production schedule along with its encoded form can be found in figures 5.1a and 5.1b.

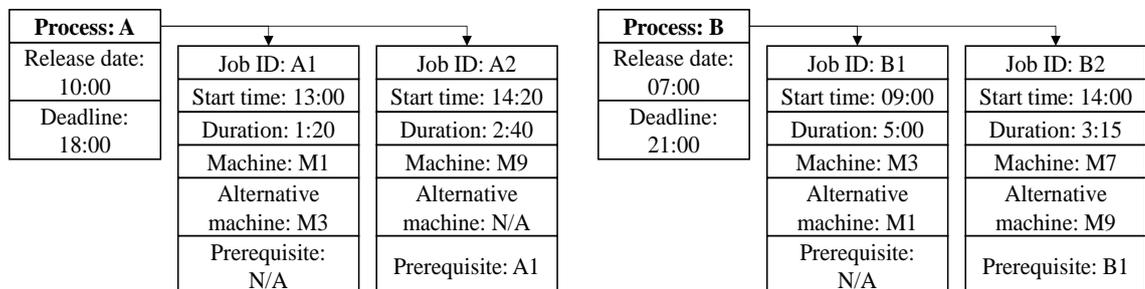


Figure 5.1a - Example of a two process four job production schedule.

Original job ID	A1	A2	B1	B2
Encoded job ID	0	1	2	3
Prerequisite	-	0	-	2
Chromosome	360	440	120	420
Duration	80	160	300	195
Release date	180	180	0	0
Deadline	660	660	840	840

Figure 5.1b - Encoded schedule (from figure 5.1a) with $T = 00:01:00$ and $s_e = 07:00$ (release date for process B).

Figure 5.1b shows that additional none timestamp based information such as job ID can also be encoded. Taking the prerequisite constraint as an example, it is computationally quick to process, store and lookup jobs based on a singular number instead of an alphabetic or alphanumeric tag. While most of the constraints are originally time stamps, and as such can be easily encoding with the scheme discussed, one constraint which cannot easily be encoded is resource availability. As it is assumed that a machine can only process a single job at any one time, a primitive and efficient method for ensuring this is to pair each machine with a discrete semaphore-based time series $M_k(t)$, to represent its usage ranging from the earliest start time to the latest deadline, in steps of T . For each discrete time step, a machine can either be, or not be, in use. Therefore a job can be reserved on a machine by setting the time series semaphores which fall within its processing interval.

5.2 Genetic Operators

To determine a suitably optimal solution, the Genetic Algorithm is tasked with intelligently exploring the potential search space. As Genetic Algorithms are based on the concepts of natural selection and evolution, the tools used by the Genetic Algorithm can be seen as numeric emulations of these natural processes.

5.2.1 Population Generation

When the Genetic Algorithm is first initiated, its knowledge of the search space is nil. Therefore the first population considered is pseudo-randomly generated, with the pseudo-randomness coming from the fact that each candidate schedule chromosome must comply with the schedule and production line constraints. For each schedule chromosome, the generation constraints for each of its job start times are shown in (5.3).

$$\max \left\{ \begin{array}{l} r_{u,i} \\ s_i^h + p_i^h \end{array} \right\} \leq g_i \leq d_{u,i} - p_i \quad (5.3)$$

Equation (5.3) ensures that for each randomly generated job start time, the job will begin on or after the jobs associated process release date, $r_{u,i}$. Where there are prerequisite jobs h , with start and processing times denoted by s_i^h and p_i^h respectively, (5.3) ensures the current job starts after the last prerequisite job has finished, $\max\{s_i^h + p_i^h\}$. Acting as an upper bound, the job cannot begin any later than its process deadline minus its processing time, p_i . Equation (5.3) considers all constraints except resource allocation. While originally included, it resulted in a significant increase in chromosome generation time as even when encoded, it is costly to consider due to its time series nature. As all constraints are checked in the fitness function (see section 5.3), not considering it here is seen as an appropriate balance between generating a population of valid schedules and generating them quickly.

Equation (5.3) is applied to the start times of every job in $N-1$ schedule chromosomes, where N is the number of schedule chromosomes in the population considered by the Genetic Algorithm. The final schedule chromosome is a direct encoding of the original schedule generated by a production planning software which does not consider energy consumption. This is done to provide result insurance and to impose an upper bound on the optimisation. In reference to the former, as a Genetic Algorithm is able to clone its best chromosomes through elitism, by including the original schedule it ensures that if no other valid schedule can be found after a predetermined number of iterations, the original will be returned to the production line manager. This may also be the case if the original schedule is, by coincidence, the most optimal in terms of energy consumption variance. The latter factor is to ensure that the Genetic Algorithm never returns a schedule which has a worse energy consumption variance than the original. This also applies if the original schedule is already the most optimal one.

To achieve a more optimal reduction in energy consumption variance, the optimisation system is permitted to extend the makespan of processes so long as the schedule constraints are maintained, specifically the processes release date and deadline. For processes where this is not suitable, process prioritisation can be used to limit the amount of allowed makespan expansion. Each process is assigned one of four priority levels which signal the optimisation system to

artificially shift the related processes constraints forwards or backwards in time. For schedules which are devised beginning from the process release date, the highest priority results in the related processes deadline being brought forward in time up to the completion of the last job in that process. This means that the schedule optimisation system can operate as normal, however that processes makespan cannot be extended past the last job completing. The lowest priority level has no influence on the constraints, while the intermediate two levels bring the deadline forward accordingly. For schedules which are devised retrospectively from the deadline, the exact same methodology can be applied to the process release date, with it being artificially moved closer to the first job in the process, producing the same effect. While a multi-objective optimisation system could simultaneously consider both energy consumption variance and makespan, this would produce solutions along a pareto front. As the original schedule is already optimised for minimal makespan, employing a multi-objective system which reconsiders it along with energy is believed to be unnecessary. It is assumed that should maximising overall production throughput be the primary objective to a manufacturer, optimising a schedule for energy consumption variance will not be considered.

Once the initial population has been generated, the fitness of each schedule chromosome is calculated using the methodology discussed in section 5.3. Once the fitness values for each schedule chromosome are known, the population can be evolved.

5.2.2 Selection

The purpose of selection is to construct a reproducing population with which to evolve into the next generation of the population. Every chromosome has an opportunity to be included in the reproducing population however chromosomes with a better fitness value are favoured. This can be seen as an emulation of ‘survival-of-the-fittest’, and aims to maintain a diverse population where the average chromosome fitness betters with each iteration. As this Genetic Algorithm is tasked with finding a schedule which produces the lowest energy consumption variance, a better fitness value is a lower one.

Due to its responsibility with constructing the reproducing population, the selection procedure has significant influence on the overall effectiveness of the heuristic search and the optimality of the best solution found (Karaboga, 2000, ch. 1). Tournament selection was chosen to implement this functionality. This is primarily due to its suitable compatibility with the fitness value assignment scheme, where chromosomes representing schedules which do not comply with the constraints are assigned a very high number, which is ideally outside the practical boundaries for valid schedules (this is explained further in section 5.3). In a rank-based selection scheme, a highly undesirable chromosome would have significantly little probability of being selected despite the fact that including them in the reproducing population is considered important to maintain diversity in the next generation. To this end, tournament selection was seen to be suitably considerate of the wide range of fitness values. There are also additional performance related reasons in support of this decision as tournament selection does not require sorting of the population and the overall behaviour of the selection process is heavily influenced by the size of the tournament population (Blickle and Thiele, 1995, ch. 3). The tournament population is built by randomly selecting schedule chromosomes from the main population with replacement. Once built, the schedule chromosome with the lowest, and therefore more optimal, fitness is extracted and placed within the reproducing population. This is repeated until the reproducing population is of size $N-2$, leaving space for two chromosome elitism. Once a reproducing population has been collated, it can be manipulated and evolved using the genetic operators – crossover and mutation.

5.2.3 Crossover and Mutation

The purpose of the genetic operators is to emulate reproduction with an aim to producing a more optimal population. Initially, the schedule chromosomes in the reproducing population are randomly paired, with each having an equal probability of breeding and producing children. Should a pair be selected, the crossover operator, emulating chromosomal crossover, creates two child chromosomes using the job start times from two parent chromosomes. For each job in

the child schedule chromosome, its assigned start time can come from either parent. The supplying parent is selected at random. The two child schedule chromosomes are polar opposites of one another, in that if the start time for job i in child one comes from parent one, the start time for the same job in child two will source from parent two. Two crossover strategies were investigated for this implementation – single-point crossover and uniform crossover. These were chosen to represent the computational complexity range of different strategies available, with single-point crossover representing the simplest approach and uniform crossover, the more complex approach. An example of uniform crossover can be seen in figure 5.2.

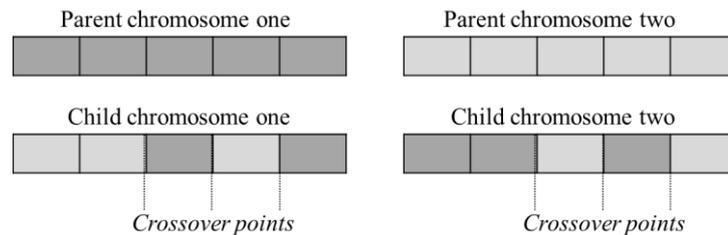


Figure 5.2 - Diagram showing the act of uniform crossover.

Each strategy was evaluated in a full implementation of the schedule optimisation system with a comparison of system runtimes along with the percentage of variance optimisation relative to each schedules original variance. The results are shown in table 5.1.

Table 5.1 - Comparison results of uniform and single-point crossover strategies (Results are averaged over five consecutive runs. Ran on an Intel i5 Windows 7 PC).

Number of jobs in schedule	Optimisation (%)		Runtime (mins)	
	Uniform	Single-Point	Uniform	Single-Point
8	72.96	72.92	247.56	233.43
12	28.67	28.67	291.53	299.88
20	68.77	66.64	231.19	225.58
30	60.06	60.01	80.60	84.01
40	57.75	58.83	18.55	23.17
50	48.18	54.28	8.11	10.10

With table 5.1 showing minor or inconsistent difference between the two strategies, uniform crossover was selected. Depending upon the crossover probability, if a pair of schedule chromosomes is selected to crossover, for every job in child one, there is a 50/50 probability that the start time will be sourced from parent one or parent two.

Mutation of the individual job start times also occurs to introduce new areas of the search space for the Genetic Algorithm to consider. Each start time has a small probability of being mutated. If selected, the start time is altered to a new encoded start time using equation (5.3).

5.2.4 Partial Validation of Schedule

During both crossover and mutation, there is the possibility of schedule chromosomes breaching one or more of the constraints as a result of the random nature of the operators' assignment strategies. To maximise schedule validity in the Genetic Algorithms population, following each operator, the schedule chromosome undergoes a series of checks to determine its compliance with certain constraints. This can be seen as an emulation of the error-correcting abilities of DNA polymerase enzymes which correct mistakes made during DNA replication. Again, compliance with all the constraints is not undertaken to minimise computational time. A full constraints compliance check is undertaken in the fitness function (section 5.3).

Post crossover, each start time which immediately follows a crossover point is checked to see if it complies with the constraints as given in (5.3). If the start time is found to be breaching one or more constraints, it is adjusted according to (5.4), to start directly after the last prerequisite job to finish has completed. While it is possible that naively adjusting one start time while not considering the ones that come after will simply shift the problem to the next start time, this can be seen as an efficient balance between efficiency and schedule validity in the overall population. Should the schedule chromosome be selected for mutation, the problem may be rectified by either the mutation operator or the post mutation checks which examines all the start times.

Post mutation, each start time is checked to determine if it complies with (5.3). If a start time does not comply, it is naively reassigned according to (5.4). While a random reassignment would be beneficial for the search, there is no guarantee that the compliance issue will not persist after the reassignment. By reassigning the job to begin as soon as possible, this ensures if

there is still a compliance issue, it is due to the rest of the schedule and not that particular start time.

$$g_i = \max \left\{ \left(s_i^h + p_i^h \right) \right\} + 1 \quad (5.4)$$

5.3 Fitness Functions

In the context of this optimisation system, the fitness function serves two purposes: a) thoroughly determine if a schedule chromosome is valid and can operate within the constraints of the schedule and production line; and b) if it is valid, reference a library of historical energy profiles and predict the energy consumption variance that would be produced were that schedule actually executed. This variance is assigned as the candidate schedule chromosome's fitness value. Thereby the lower the fitness value, the more optimal the schedule chromosome.

Prior to any predictions, the candidate schedules are checked against the constraints in a procedure much more thorough than the checks which occur inside the genetic operators. In their encoded form, the following are grouped together based on their respective jobs:

- job start times, extracted from the schedule chromosome,
- job durations,
- resource assignments,
- prerequisite job relationships,
- parent process deadline.

As it is assumed that the job duration also includes any necessary setup, inspection and unloading of parts, the job's duration does not need to be adjusted to consider this.

This grouping permits the schedule to be considered from an individual process viewpoint. The schedule is checked to determine if there is any overlapping of related jobs which must be executed in series, and to ensure that the last job in the process to finish does so either before or on the deadline. Assuming a candidate schedule passes these checks, it is then checked to ensure

it does not over-utilise the production lines resources. As introduced in section 5.1, each machine is assigned a discrete usage profile which ranges the full time span of the schedule with time spacing T . For each job, the system queries the appropriate machines usage profile to determine if all the discrete time slots for the duration of the job can be reserved. If so, those slots are reserved. If a requested slot has already been reserved by another job, this check fails.

In total, each schedule must successfully pass four checks with binary results – pass or fail. These consist of (5.3) and (5.5).

$$\max\{M_k(t)\} \leq 1 \quad (5.5)$$

As before, (5.3) ensures that all prerequisite jobs are complete prior to the job itself starting, that no job begins prior to its parent process release date, and that the process deadline is not breached. Equation (5.5) ensures that each of the available resources M_k are never over utilised.

Each check is undertaken sequentially, allowing for easy integration of further checks. Even with all the constraints, the above checks are necessary due to the fact that, a) the initial generation constraints do not consider resource utilisation, for time efficiency it is seen to be more efficient to incorporate this into the centralised checks here than including it in the population generation and post operator checks. And b), due to the random assignment of start times within the available time, there is the possibility that the generation algorithm will randomly assign the first job in a process to run at the very end of the processes available time. As such because of the constraints the generation algorithm will have no choice but to assign the trailing jobs to overlapping start times. While additional constraints could be added to prevent this, it is believe that this would influence the overall performance of the Genetic Algorithm by overly constricting its random nature.

In the case of resource utilisation, if a resource is over utilised additional options are potentially available besides the standard pass or fail. The schedules for this particular application allow for manufacturers to specify a set of alternative machines on which the job could potentially run on. In the case where the default machine is not available, the fitness function will attempt Dynamic Machine Reassignment (DMR). Here the alternative machines are queried to determine if they

can execute the conflicting job. Should a machine be able to take on the conflicting job, the schedule will be adjusted to reflect the change. The alternative machine is assigned as the jobs default machine, while the original default machine is moved into the alternative machine list. The reassignment process occurs in a deterministic manner permitting repeatability. It should be noted that performing DMR does not guarantee that a resource conflict will be resolved. Should no alternative machines be available to execute the job, or if the manufacturer has not specified any alternatives, the schedule will fail the check.

Throughout the checking procedure, should any check fail, the candidate schedule is classified as invalid. This implementation presents an issue with regards to the representation of invalid schedules and their schedule chromosomes. They must be represented by assigning them a fitness value in order for the Genetic Algorithm to learn, however the fitness value must be such that the Genetic Algorithm will never return an invalid schedule as the final result. Thanks to the inclusion of the original non-energy optimised schedule, this is solved in principle as the system will never return a solution worse than the original. Therefore any invalid schedule can be assigned a fitness value which is above the fitness value of the original schedule. In practice, the fitness function assigns invalid schedule chromosomes with a fitness value equal to the maximum value of a primitive double data type – 1.7×10^{308} .

Candidate schedules which successfully pass all of the checks are classed as valid and are permitted to continue and enter one of the two energy consumption variance prediction engines. Which prediction engine is selected is dependent upon the prediction mode the system is operating in. This is discussed in detail in sections 5.3.1 and 5.3.2.

5.3.1 Probable Energy Consumption Variance Prediction Engine

The purpose of the probable energy consumption variance prediction engine is to generate a highly accurate prediction of the energy consumption waveform based on the given candidate schedule. The predicted waveform is constructed by inserting empirical job energy profiles at the appropriate moment in time into a time series energy profile. The high accuracy is primarily

sourced from the use of the Intelligent Historical Library (chapter 3), which is used to select the most probable profiles to use during the prediction's construction. To accomplish this, along with standard information, the production schedules for this application also hold additional data relating to how the job will influence the machines mechanical status (i.e. tool wear, machine lifespan, etc.). For identical jobs, this type of data will match the metadata of the profiles within the intelligent library. The job influence data itself is machine specific and is determined by historical maintenance data, documentation, domain expert knowledge, or a combination of the three. Prior to the optimisation system commencing, it is provided with information relating to each machine's current mechanical status. From this point, it is possible to calculate how each machine's status will change as the jobs are executed; this is demonstrated in figure 5.3.

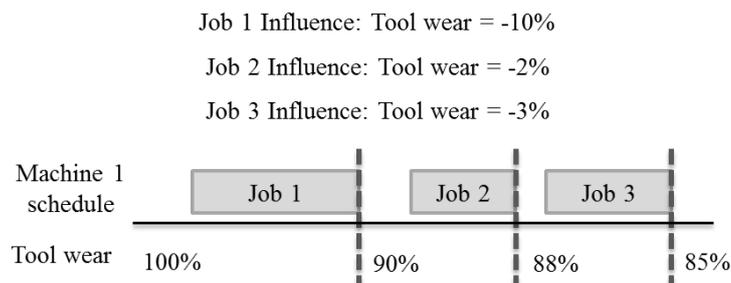


Figure 5.3 - Image showing how the fitness function simulates the job influence on a machine. In this example, tool wear is used as a singular influence parameter.

For each job, once the machines future status has been predicted, it is formed into a query for the intelligent library, which returns the profile with the closest matching metadata.

After this has been completed for each job in the schedule, the production lines energy profile can be compiled via the construction of the individual machine profiles $profile_{Predicted}^{Mk}$ for all machines k . Each machine's energy consumption profile consists of a periodic time series whose length is an integer multiple or divisor of T . This allows for the energy profiles to be far more granular than T . As this particular application of the optimisation system focuses on electrical current, initially the machine profiles consist purely of the machines idling current. This means that if a machine M_k is never used in the schedule, it will be treated as constantly idling. The jobs are then sorted in chronological order and the selected energy consumption

profile is overlaid onto the appropriate machine consumption profile beginning at the job's start time. This act overwrites the value currently assigned to the time series at that particular location. This is repeated for all jobs in the schedule. Once complete, any individual post-job idling time is filled with an updated idling current of the particular job just executed. This is to reflect that the idling current may now have changed as executing the job may have left the machine in a different position and/or configuration. The optimisation algorithm assumes that when not in use, a machine is left idling.

Once these individual machine profiles have been generated, a predicted production line energy consumption profile can be compiled by summing the individual machine profiles according to (5.6).

$$profile_{Predicted}^{Prod}(t) = \sum_{k=0}^{|k|} profile_{Predicted}^{Mk}(t) \quad (5.6)$$

Finally the variance of the predicted production line energy profile E_{Var} , and the fitness of the schedule chromosome is calculated using (5.7).

$$E_{Var} = \frac{T}{(D_{Max} - S_e) - 1} \sum_{t=S_e}^{D_{Max}} \left[profile_{Predicted}^{Prod}(t) - \overline{profile_{Predicted}^{Prod}} \right]^2 \quad (5.7)$$

Throughout the remainder of this report, results produced by this prediction engine will be labelled as 'most probable'.

5.3.2 Best Case Energy Consumption Variance Prediction Engine

Along with the most probable result, the use of a library of historical energy profiles allows for the lower bound on the production line's energy consumption variance to be predicted. This allows the manufacturers to determine where the probable result lies in relation to the lower bound. To enable this, instead of selecting the historical energy profiles based on probability, they must instead be intelligently selected to minimise the total energy consumption variance. Because the energy-based focus in this research is on electrical current, a best case prediction

could be theoretically made by intelligently generating the job start times, as before, but for each job, simply use the historical profile with the lowest variance. However this implementation will not work if the energy focus were on power factor, where inductive and capacitive loads can interfere and negate one another. Therefore to maintain expandability, when the optimisation system is set to locate the best case result, the Genetic Algorithm itself is tasked with locating for each job, both a suitable start time and an historical profile to use in the prediction engine.

In this implementation, the Genetic Algorithm operates on both the encoded job start times and profile selection simultaneously within singular chromosomes. Figure 5.4 shows a comparison of the chromosomes compositions when the optimisation system is set to both most probable and best case predictions. In figure 5.4 it can be seen that when the optimisation system is set to locate the best case result, the length of the chromosome doubles. The first half contains the encoded job start times while the latter half contains ID numbers which act as a form of encoding for which historical profile to assign for each job. Using a form of real-value encoding, all historical energy profiles are grouped based on the job they reference, and are then assigned zero-based unit incrementing integer identification numbers unique within the context of each job grouping. To maximise efficiency and minimise program size, both halves of the chromosome are processed using the methodologies discussed in the previous sections. Each profile ID gene is treated and processed as a manufacturing job start time, just with no prerequisite jobs, zero processing time, a release date of zero, and a deadline equal to the number of profiles available for the related job.

exact nearest neighbour results with the appropriate profile ID and updating the schedule chromosome with the correct information.

As before, results produced from this prediction engine will henceforth be referred to as ‘best case’.

5.4 Overall Algorithm Design and Implementation

The schedule optimisation system discussed in this chapter was realised and implemented using the C# programming language. An object oriented language was selected as it was believed that this fits with the problem to be modelled. As schedule chromosomes and jobs are multivariate data, representing them as objects with internal parameters and functionality was seen as a suitable representation. C# itself was chosen because of its native support for multithreading and .NET remoting, which allows for the potential integration of other programs such as traditional schedulers, and for agent-based distributed processing.

Existing formats for production schedule data were investigated in an effort to allow for interoperability between existing production schedules and the developed optimisation system. However, due to the necessity to include energy-related information, a custom scheduling format was devised. An example of this format in its pseudo-raw form can be seen in figure 5.5.

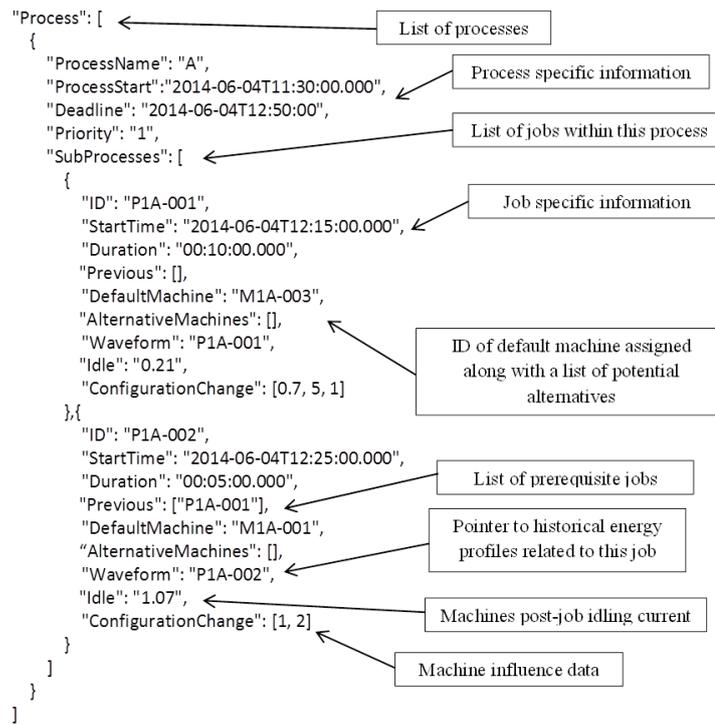


Figure 5.5- Example of a single process, two job schedule in the format accepted by the optimisation system.

Figure 5.5 shows that the schedule file contains all the information and constraints needed by the optimisation system to produce a valid schedule. Along with the constraints, each job contains a list of alternative machines it can be executed on, for use with Dynamic Machine Reassignment. There is also a pointer to the historical energy profiles for that job, which in this case for clarity is set to the job’s alphanumeric identification tag. Finally, there is the post-job idling current, along with the machine influence data required for constructing queries for the Intelligent Historical Library in the most probable prediction engine. All these files are stored in the JSON format which is human-readable, and can be efficiently interpreted by the program.

Similar to the schedule, the information regarding the production line resources are held in a different custom format as seen in figure 5.6. Every machine present in the production line is listed in this file with a unique alphanumeric identification tag, along with the default idling current of that machine. Additionally it is in this file that the other half of the machine influence data, the machine status data, is held. Each status variable holds three numbers – the first indicates the machine status prior to the schedule commencing, while the next two indicate the lower and upper bounds of the status permitted by the production line manager. Using tool wear

as an example, the values [50, 0, 100] indicate that the tool is currently halfway through its acceptable wear cycle. Jobs which use this machine will contain a singular value representing the amount of tool wear inflicted during the job, and this is added to the current value in order to predict the future machine status. The data itself is not supplied with a unit of measure; as such the production line manager is expected to maintain consistency. Throughout the schedule's run, the predicted status value for each machine should not leave its specified boundary range. It is assumed that the production line manager will schedule any required maintenance operations to assure this. Because maintenance can be represented as ordinary jobs, the influence data can be used to reset the machines status value.

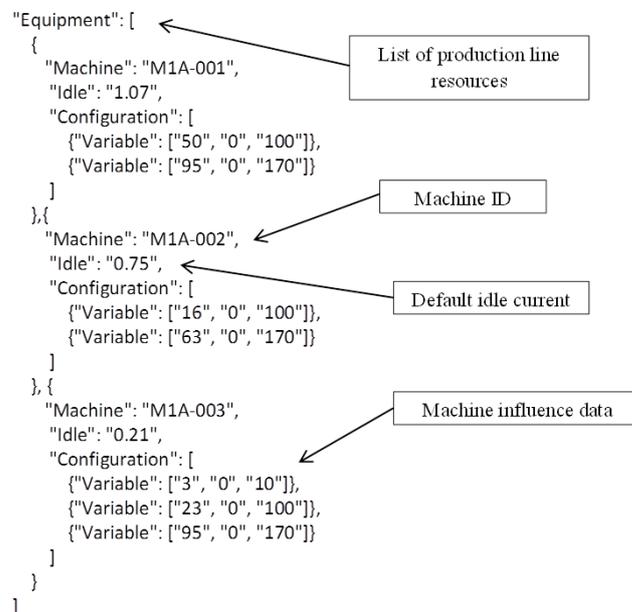


Figure 5.6- Example of the production line information in the format accepted by the optimisation system.

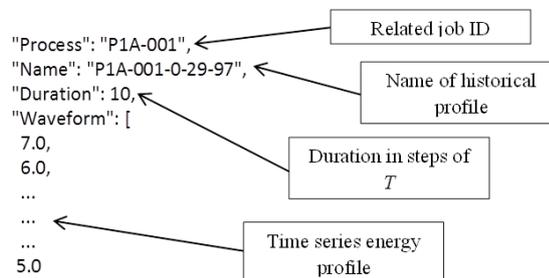


Figure 5.7- Example of a historical job energy profile in the format accepted by the optimisation system.

Figure 5.7 shows an example of a file containing a historical energy profile used within the Intelligent Historical Library. Each historical energy profile is held in its own file and is

distinguished by a unique name consisting of the associated job's ID tag, along with the profiles metadata. The example shown in figure 5.7 contains the unique name *PIA-001-0-29-97*. Here, *PIA-001* is the associated job tag, while *0-29-97* are the three pieces of machine influence metadata. The profile files themselves do not directly contain the metadata. Instead, the metadata for all profiles is located within a singular file known as the *Master-Log-File*. This file contains the metadata along with pointers to the individual files grouped based on their associated jobs, and acts as a centralised directory. The pointer to the historical profiles located within the schedule files points to one of the groupings, to which the Intelligent Historical Library extracts the metadata records for use with the exact nearest neighbour search algorithm. From that result, the actual pointer to the energy profile file can be extracted and used.

Ideally, the energy profiles along with the *Master-Log-File* should be implemented within a database structure. However since they are all loaded into memory at runtime to minimise resource access time, there is no foreseeable advantage to using a database over a flat file system. Additionally, using a flat file system permits maximum portability.

Initial testing of the entire optimisation system demonstrated that the concept of schedule invalidity presented further issues. Because the search space likely contains a high concentration of invalid schedules, combined with the necessity to not overly constrain the population generation and to represent invalid schedules, the population is quickly saturated with invalid schedule chromosomes. As a result, little to no optimisation progress is made. To overcome this problem, the population would be periodically revitalised by regenerating it. Once the Genetic Algorithm has completed a predetermined amount of iterations, the algorithm is forced to restart and begin anew with a fresh population. To maintain overall progress, prior to the restart, the most optimal solution found is extracted and injected into the newly generated population. As explained previously, during the initial iteration, the schedule injected into the population is a direct encoding of the original production schedule.

Figure 5.8 shows how the components of the entire optimisation system connect with one another and demonstrates how the Genetic Algorithm is integrated into the system. Operating in parallel with the crossover and mutation operators, elitism is used to locate the two fittest

schedule chromosomes and clone them. The clones are then reintroduced into the population after bypassing the reproducing operators. The size of the population is maintained by constructing the reproducing population with $N-2$ schedule chromosomes. Once the reproducing operators have concluded, the clones are reintroduced making the population size N .

After completing the outer Genetic Algorithm restart iterations, the optimisation procedure is considered complete. The fittest schedule chromosome ever found is extracted from the population and its job start times are decoded back into the time domain using (5.2). Any changes to the resource assignment as a result of Dynamic Machine Reassignment are also extracted. The original schedule is then updated with all this new information and output to the production line manager.

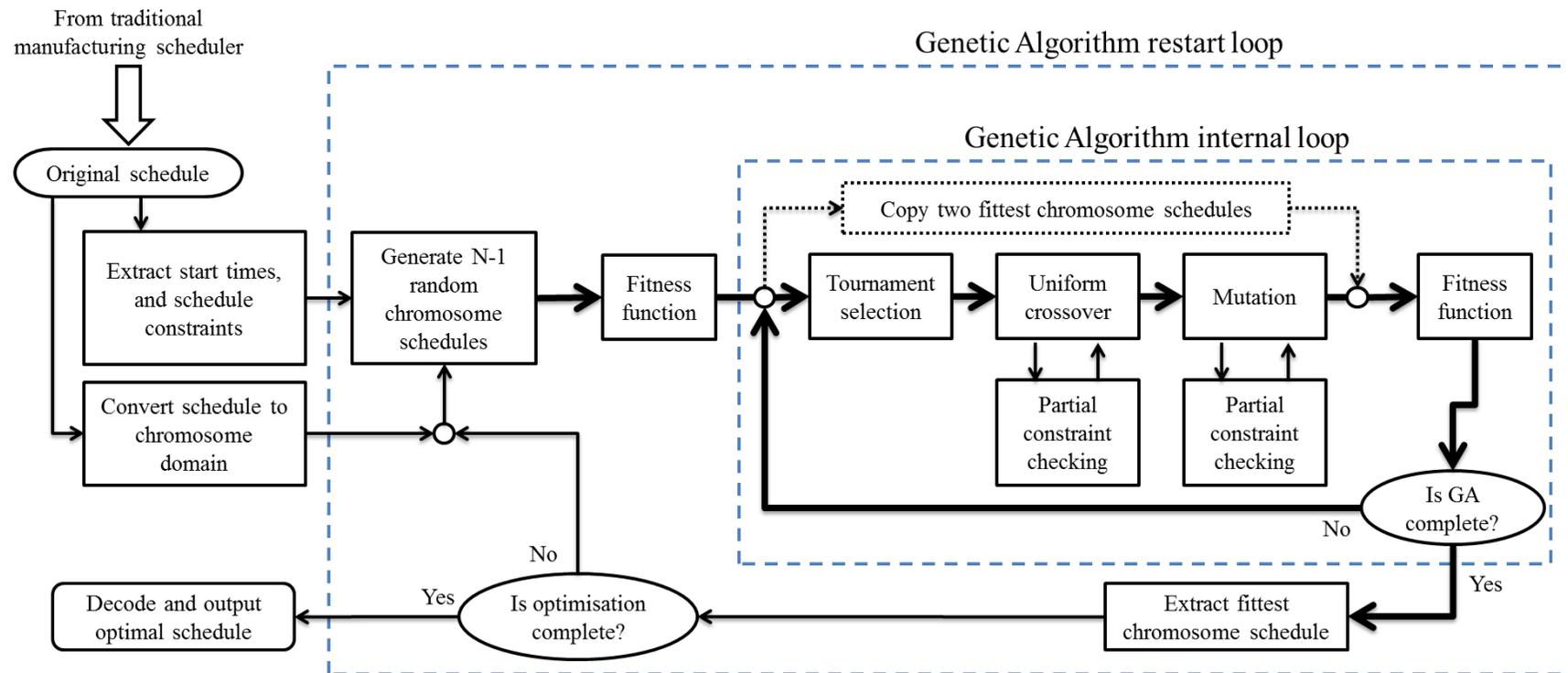


Figure 5.8 - Overview of Optimisation System.

CHAPTER 6

SCHEDULE OPTIMISATION SYSTEM – PERFORMANCE

ENHANCEMENTS

With a fully constructed schedule optimisation system, a logical continuation of the work is to optimise its performance such that it is able to return the upper bound of suitably optimal results while minimising runtime. This stage of development can be considered equally as important as the design and implementation stages as it directly affects the suitability of the optimisation system to the given problem. In this chapter, the internal parameters of the Genetic Algorithm along with the system's termination conditions are optimised through experimentation. Finally section 6.3 details the development of a coarse preliminary prediction methodology which aims to give production line managers an insight into the level of potential optimisation available within a schedule prior to committing to the full optimisation procedure.

6.1 Improvements to Genetic Algorithm Search Efficiency

Within the literature review in section 4.1, while many researchers utilise Genetic Algorithms in their work, one matter which they seldom mentioned is the optimisation of the Genetic Algorithms internal parameters such as the probabilities of crossover and mutation. To maximise search efficiency and to aid in successfully locating a suitably optimal result, the optimal value of these probabilities must be discovered (Lin et al, 2003; Yang 2014, ch. 5). Additionally, in this implementation, an optimal size of the tournament selection population must be found. Were it overly large in comparison to the main population, diversity would rapidly reduce as local optimal solutions begin to dominate the reproducing population. Should the selection size be too small, fitter schedule chromosomes may not be selected.

Experiments were carried out to determine how the probabilities of crossover and mutation, and the tournament selection population size, $P_{Crossover}$, $P_{Mutation}$ and $N_{Tournament}$ respectively, influenced the performance in terms of the most optimal found solution. In the context of this work, the most optimal found solution refers to the most optimal solution currently discovered by the optimisation system. The results of this experiment can be found in figures 6.1a and 6.1b. It can be seen that overall, the values of $P_{Crossover}$ and $N_{Tournament}$ have little influence on the most optimal found solution when compared with $P_{Mutation}$. Beginning with a mutation probability of 1, the algorithm makes no progress as every job start time in every schedule chromosome is guaranteed to be mutated. Therefore the original schedule is returned, shown in figures 6.1a and 6.1b by the upper flat ceiling. In both cases, while the most optimal found solution eventually reaches its respective lower boundary before gradually inclining, this occurs at different mutation rates in each schedule. For an eight job schedule, this low period occurs at the mutation probability of approximately 0.7, while for the 50 job schedule it occurs at approximately 0.05. Therefore it is concluded that:

- a) When compared against the mutation probability, crossover probability and tournament selection size have an inconsequential influence on the most optimal found solution.
- b) There is no viable combination of Genetic Algorithm parameters which permits maximum performance in returning a suitably optimal solution while maintaining universal compatibility with all schedules and constraints.

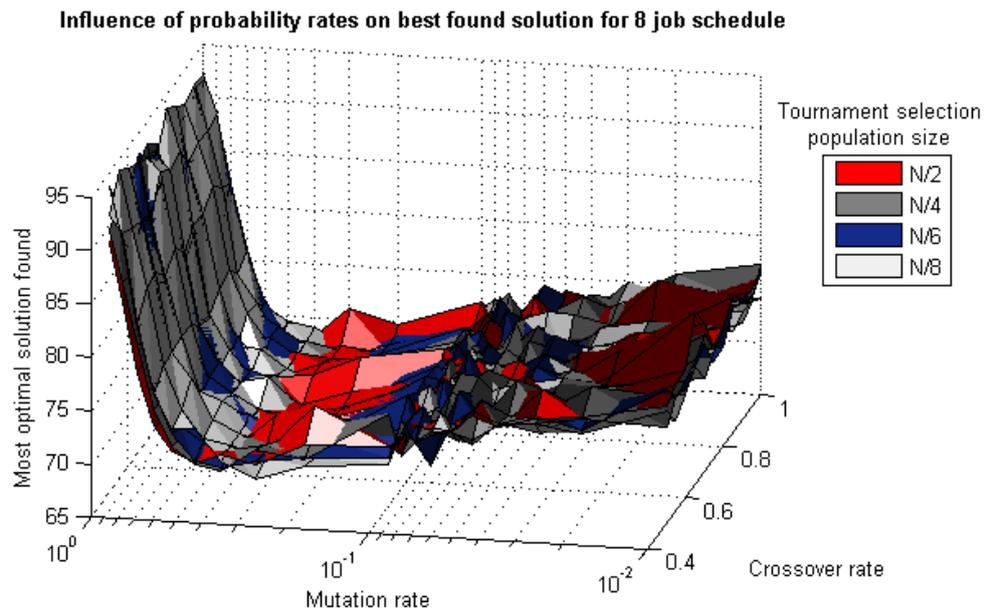


Figure 6.1a - Graph showing the influence crossover and mutation rates along with tournament selection size have on the most optimal found solution with an eight job schedule. Results averaged over ten consecutive runs.

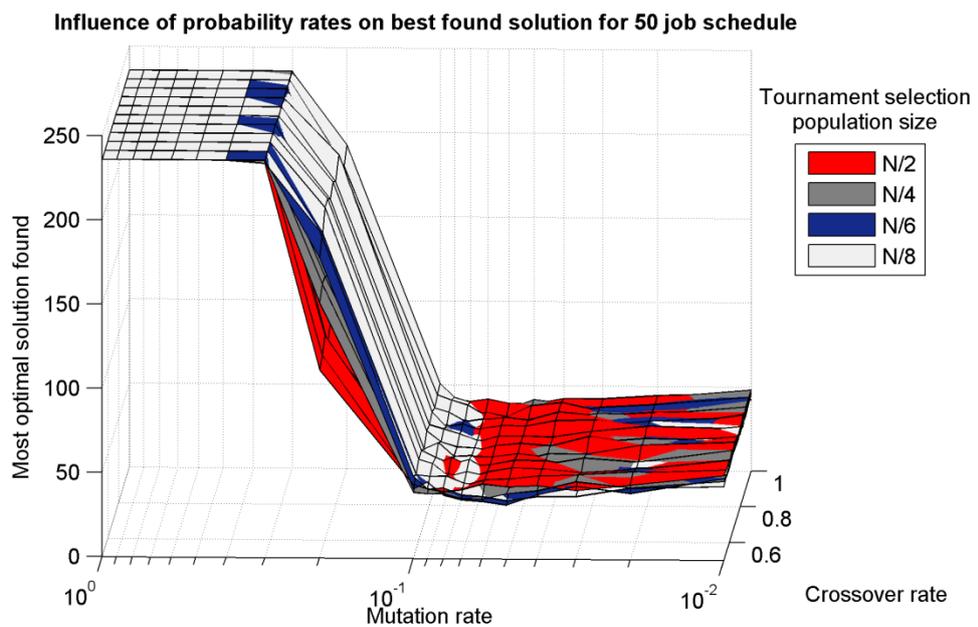


Figure 6.1b - Graph showing the influence crossover and mutation rates along with tournament selection size have on the most optimal found solution with a ten job schedule. Results averaged over ten consecutive runs.

To this end, the optimal universal parameter values were considered to lie in the most common optimal point in both test results. These values were chosen as $P_{Crossover} = 0.5$, $N_{Tournament} = N/4$, and $P_{Mutation} = 0.1$, and were held for the remainder of this work.

In many applications, the Genetic Algorithms parameters are dynamically adjusted in response to the progress made by the algorithm as it executes (Lin et al, 2003; Ailva and Falcão, 2008).

While this approach is seen to be the more suitable, it was decided that fixed values for the crossover and mutation probabilities, and tournament selection size would be used. This decision was made on the grounds that the results in figures 6.1a and 6.1b show that certain parameter values result in no progress being made. Additional computation overhead needed for the adaptation procedure would also be avoided.

The final parameters to be optimised were the termination conditions. In initial testing, these were given the default values of 100 generations of the population followed by 1000 Genetic Algorithm restarts, with a population size of 100 schedule chromosomes. As the global optimal value for each schedule cannot be known beforehand, the optimisation system is forced to execute for these predetermined number of generations and algorithm restarts. As such the value of these termination conditions have a significant influence on both the best solution found, and the overall runtime of the system. Experiments were conducted using a 50 job schedule. As the largest and therefore most complex schedule considered in this work, it was seen as the upper boundary for the termination conditions under the assumption that a problem with more jobs and therefore constraints would require more iterations to solve. For this to be achievable, the optimisation system needs to evolve a population with sufficient generations in order to explore the search space sufficiently, and renew that population once the diversity of valid schedule chromosomes has dropped below acceptable levels. Numerous different combinations of the two conditions were tested in an effort to locate the pairing which returned the most optimal solution. All combinations of 100, 300, 500, 700 and 1000 restarts, and 10, 50, 100, 200 and 500 generations were evaluated. These were selected to allow for a suitably wide spectrum to analyse, while maintaining an acceptable runtime. The results, shown in figure 6.2, reveal that 100 generations surrounded by 700 algorithm restarts gives the optimisation system sufficient opportunity to find a suitably optimal solution. All results and experiment data subsequently produced use these termination conditions.

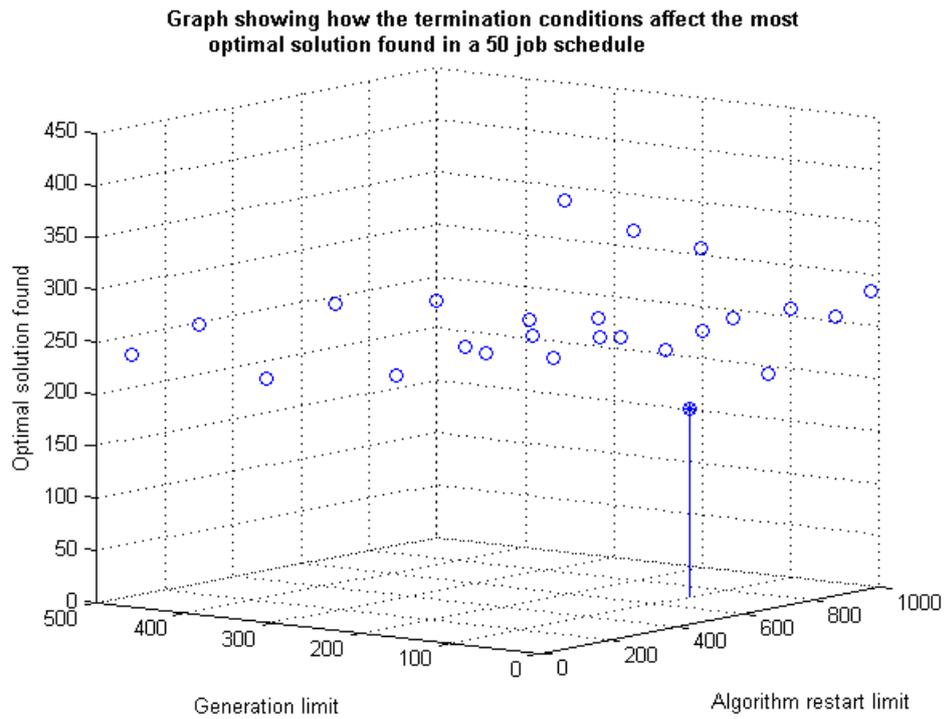


Figure 6.2- Graph showing how the termination conditions affect the most optimal solution found in a 50 job schedule. Optimal pairing shown by stem. Results averaged over ten consecutive runs.

It should be noted that the population size of 100 schedule chromosomes was left at its default value and was not investigated like the other termination conditions. This was left as such as an analysis of the population revealed that despite the procedure of restarting the Genetic Algorithm periodically, the population was still becoming heavily overrun with invalid schedule chromosomes. Therefore it was determined that a population of 100 schedule chromosomes represented a sufficient balance between having a large enough population to ensure diversity while not majorly exacerbating the runtime of the optimisation system. The effects of the population size on the optimisation systems performance is later investigated in section 7.7.

6.2 Improvements to Runtime

In initial testing, the optimisation process demonstrated an acceptable runtime given the default termination conditions specified in section 6.1. However when tested with high granularity (150ms per data point) profiles as detailed in chapter 3, the optimisation system began to suffer from significant runtime performance degradation.

Analysis of the program's execution revealed that due to the high granularity of the energy consumption profiles and therefore the length of their time series representations, a significant proportion of the runtime is spent in the fitness function, sorting and compiling the individual machine and final production line energy profiles, along with calculating the final variance. This is also hindered by the operations being of a sequential nature and by the fact that these must also be undertaken for every valid schedule chromosome in every generation of every population. To improve runtime performance, three distinct improvements were made to the optimisation systems implementation – program code optimisation, multithreading, and caching.

Improvements to the program code were initially made to allow for a more consistent execution. This included loading all information required by the optimisation system, such as the historical energy profiles into memory. Alterations to the program code for the two prediction engines were also made to increase efficiency. Furthermore, efforts were made to introduce multithreading where possible. A Genetic Algorithm is a sequential algorithm, with each generation being acted on based on the previous generation. As such, it is not possible to parallelise the algorithms execution. This is not the case with the fitness functions, where the same operation is applied to each schedule chromosome in the current generation independently. The full schedule validity checking part of the fitness function was therefore adjusted to execute in parallel. The same could not be applied for the actual prediction engines, where changes due to program code optimisation did not allow for the parallelisation of this particular function. Unfortunately, as table 6.1 shows, multithreading provided a negative performance increase. This is likely due to the fact that the overhead necessary to execute the code in parallel outweighs the performance benefits. However, it can be seen that time difference between the two executions does rapidly decrease as the number of jobs increases. As only the job constraint checking is parallelisable, there is the possibility that as the number of jobs increases, multithreading may become beneficial. As such, for reasons of expandability, multithreading was kept in the program.

The final step to improving performance was to integrate caching functions to minimise repetitive calculations. Execution analysis demonstrated that while the efficiency of the fitness function was increased with the program code improvements, a large amount of the total runtime is still spent executing them. Because the genetic operators only take effect based on a predetermined probability, and thanks to factors like elitism, there is a probability that schedule chromosomes in the current generation were present, unchanged in the previous generation. This leaves the potential for unnecessary re-evaluation of unchanged schedule chromosomes. This is also the case were the Genetic Algorithm to accidentally revisit parts of the search space. Therefore to maximise time efficiency a hash-based caching system was implemented. These have a history of improving performance in Genetic Algorithms. In the work by Povinelli and Feng (1999), the implementation of a hash-based cache resulted in a 50% decrease in computation effort. Prior to evaluating a schedule chromosome, the cache is queried to see if it contains a chromosome with identical job start times. If it does, the stored fitness value is copied and assigned as the current schedule chromosome's fitness. If no matching chromosome can be found, the schedule chromosome is evaluated in the fitness function. Once the evaluation is complete, a copy of the chromosomes job start times along with its new fitness value is placed within the cache. As the string of job start times can potentially contain a large array of numbers, they are not directly placed in the cache with the fitness value as this leaves the potential for a costly query time. Instead a hash of the start times is inserted, generated via Fletchers checksum algorithm to allow for a single key-value pair entry. The Fletchers algorithm was selected for this as it is one of the many algorithms sensitive to number order (Maxino and Kooperman, 2009). Thereby reordering the job start times will produce a different hash value. To minimise lookup time, the cache is implemented as a hash-table.

Due to increasing hash collisions and increased memory usage resulting from a large cache, it may not be possible to maintain a cache of all considered schedule chromosomes. Therefore it will need to be cleared and reinitialised either periodically or after it exceeds a certain size. Work by Cooper and Hinde (2003), along with Povinelli (2000) investigate the best ways to achieve an effective cache size. A range of methods are approached, with both investigating

“short-term memory” caches which solely hold the hash values for the previous generation. Ultimately, Cooper and Hinde conclude that using a combination of short term memory combined with an intelligent long term memory which is never cleared but only contains the fittest chromosomes found through the Genetic Algorithms run is best. Alternatively Povinelli concludes that using a modified cache outperforms all alternatives considered. His cache holds all considered chromosomes but reinitialises the cache once a predetermined amount of hash collisions occur. While all previous cache data is lost, he states that this is not counterproductive. As the Genetic Algorithm runs, the chromosome diversity will decrease and many of the earlier cached chromosomes will no longer be considered. The newest chromosomes will be quickly re-cached. This operation acts to minimise and revitalise the cache periodically. As both results demonstrate that a short-term cache performs best, the caching system implemented within the schedule optimisation system was cleared and reinitialised every ten generations. The runtime of the optimisation system with various optimisation features can be seen in table 6.1. An analysis of the runtime performance can be found in section 7.3.

Table 6.1 - Optimisation system runtime throughout different stages of runtime improvements. Results averaged from ten consecutive runs.

Optimisation system runtime...			
Number of jobs in schedule	With optimised code (mins)	With optimised code and caching (mins)	With optimised code, caching and multi-threading (mins)
8	200.60	187.76	233.43
12	279.50	252.75	299.88
20	236.61	208.59	225.58
30	105.67	75.52	84.01
40	38.72	17.79	23.17
50	31.85	8.13	10.10

6.3 Coarse Prediction

Despite the improvements to the execution efficiency, the runtime of the optimisation system can be lengthy, with runtimes of five hours measured for some schedules. While this can be seen as acceptable, it is not always preferred. The reasoning behind this is that a manufacturer

may commit to a lengthy optimisation process on a schedule where, due to the constraints, the level of potential optimisation is very limited. Therefore providing the production line manager with an estimate on the level of potential optimisation would be beneficial. Unfortunately, while this cannot be known prior to the optimisation procedure, simplifying the prediction and sacrificing accuracy for execution speed would be a suitable method for providing an approximate level of potential optimisation in an acceptable amount of time.

6.3.1 Reducing Historical Energy Profile Length

As seen during the development phase, one of the primary factors attributed to the lengthy runtime is the use of high granularity historical energy profiles. While these are key to providing an accurate prediction, manipulating such a volume of data over thousands of iterations does result in a lengthy runtime. As such, one of the obvious methods to reduce the runtime is to reduce the granularity of the historical energy profiles.

Compression, simplification and approximation are well-researched fields with a large focus in reducing the size of time series data for numerous applications. For this particular application, the selected time series compression algorithm must comply with the following specification:

1. Maintain visual appearance of the time series at a level respective of the compression factor.
2. Maintain significant step changes, respective of the compression factor.
3. Reduce each profile by a consistent compression factor.
4. Reduced time series must have a uniform time period between data points.
5. Present reduced time series in a raw numerical time domain form.

Regarding the above specification, remarks one and two are designed so a prediction made using the reduced length energy profiles will still remain within an acceptable error margin. Remarks three to five are in place to minimise the complexity of using the reduced length energy profiles. By keeping them in the same format as the original energy profiles –

represented as an array of values with a consistent temporal spacing, the prediction engines can utilise the reduced length profiles with only very minor changes to the overall program. Reduced length profiles which utilised inconsistent time spacing or symbolic representations would require further processing before they can be utilised.

Some of the most popular time series compression algorithms are based around decomposition functions such as Fourier and wavelet transforms (Zaniolo, 2008). Despite their reputable history, they are unsuitable for this particular application as they do not compress within the time domain. More refined research has also been conducted which focuses on the compression of energy data (Ringwelski et al, 2012; Unterweger and Engel, 2015). However, as before, the algorithms used are unsuitable due to the fact that they encode the information. This is in direct violation of remark five.

Given the above specification, approximation functions hold potential for this particular application. In their research investigating compression for data streaming, Palpanas et al (2004) state that the selection process of a time series approximation algorithm should not purely consider the loyalty of the compressed waveform. Depending upon the application, other features such as visual appearance should be considered with a larger emphasis. Given that this application has a strong focus on the overall shape of the waveforms, visual appearance and the translation of major changes in the waveform between the original and compressed waveforms can be considered top priority.

Given the enormity of data which is now stored, methodologies have been developed for reducing time series data for the purposes of transmitting and displaying it on a variety of screen sizes. The most favoured methodology for achieving this, is through the identification of 'informative data points' (Fink and Pratt, 2004; Burtini et al, 2013; Fu et al, 2005). These are defined as data points where the amount of change between the neighbouring points is beyond a certain tolerance. As a result, these points are considered highly entropic (Fink and Gandhi, 2011; Butters, 2014). Typically this information is used to determine a suitable compression ratio which minimises information loss. Line simplification algorithms operate in a similar fashion. By removing redundant data points which do not significantly contribute to the overall

'shape' of a time series, the number of points necessary for rendering a waveform is reduced. This technique is especially necessary for representing large datasets (Keevey and Smyth, 2015). While all the above mentioned methods are able to compress a time series and represent it in a raw numeric form, their basis of the compression ratio around the highly informative points means that the compression ratio will not be consistent for every profile.

To allow for a consistent compression ratio, methods such as down-sampling, resampling and aggregation approximation can be used. The latter operates by dividing the waveform into equal length blocks. A mathematical function then approximates the data points within that block into a single value. In many applications, this function is simply arithmetic mean, however polynomial and linear functions can also be used (Keogh and Pazzini, 2003; Fu et al, 2005; Burtini et al, 2013). The two former solutions are to down-sample or resample the original waveform. While these can be used to compress a time series, other methods are preferred due to the fact that with down-sampling, the compressed waveform is highly dependent upon the original sampled data points (Al-Naymat and Taheri, 2008; Fu et al, 2005). Resampling was initially considered but was not adopted due to the high error in the compressed waveforms, when compared against the originals. This can be seen in table 6.2.

Aggregation based compression appears to be the most suited approach, in line with the specification barring remark two. While ensuring a consistent compression ratio, so long as the block size is maintained throughout all waveforms, important waveform features and step changes can be lost in the process. As seen in figure 2.9, machine energy profiles can be characterised by immediate and significant step changes. By applying aggregation, these steps will not only be smoothed with a reduced amplitude, but for peaks, their period will increase. An example of this can be seen in figure 6.3, where the initial inrush peak is significantly reduced in amplitude and the amplitude of the surrounding data points is erroneously increased, resulting in a poor fit. Given the very short time period of the inrush peak in comparison with the required compression ratio, it would be acceptable for the compression algorithm to simply not consider the peak. Similarly, were a significant step change to occur in the middle of a

block, it may be advantageous to only consider the data points on one side of the step, instead of all points within the block; thereby conserving the step change and its amplitude.

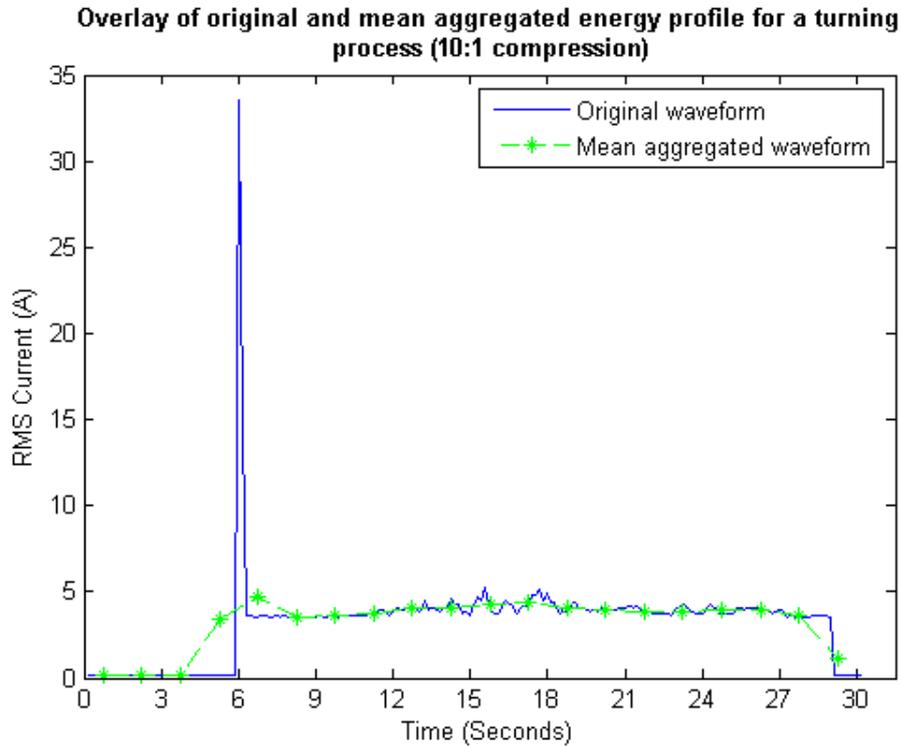


Figure 6.3 - Overlay of original and mean aggregated energy profile for a turning process (10:1 compression). Original profile is an empirical recording from a metalworking lathe.

To realise this, a custom compression algorithm was developed which combines mean aggregation with informative point detection to determine which points to consider, in order to achieve a high accuracy compression. The methodology is discussed below, with an example 4000 point waveform as seen in figures 6.4.

Step 1

Calculate the absolute moving difference of a profile using (6.1). See Figure 6.5. Normalise

between 0 and 1 to produce $abs\Delta_{profile}^{Normalised}$.

$$abs\Delta_{profile} = abs(profile(t-1) - profile(t)) \quad (6.1)$$

Step 2

Sort data into bins based on amplitude using a histogram. Discard bins which contain the lower 90% of data points. Data points in remaining bins are classified as the most informative data

points $P^{Informative}$ (Figure 6.6).

Step 3

Divide the profile into sets of features denoted by F , based on the boundary points specified in

$P^{Informative}$. See (6.2).

$$F_i(t) = \text{profile}(t) \quad \text{where} \quad P_{i-1}^{Informative} \leq t < P_i^{Informative} \quad (6.2)$$

Step 4

Divide profile into a set of equal sized blocks denoted by B , each of size B^{Size} which denotes the compression factor. (Figure 6.7).

$$B_i(t) = \text{profile}(t) \quad \text{where} \quad (i \times B^{Size}) \leq t < (i \times B^{Size} + B^{Size}) \quad (6.3)$$

Step 5

For each block B_i , determine which features F_i lie within its boundaries, and which of those contains the majority of data points within the boundaries of B_i . Remove the data points from this block that do not belong to the majority feature. In cases where there is no majority, select the first feature in the block.

Step 6

Average the remaining data points within B_i to produce a single point for each block (Figure 6.8).

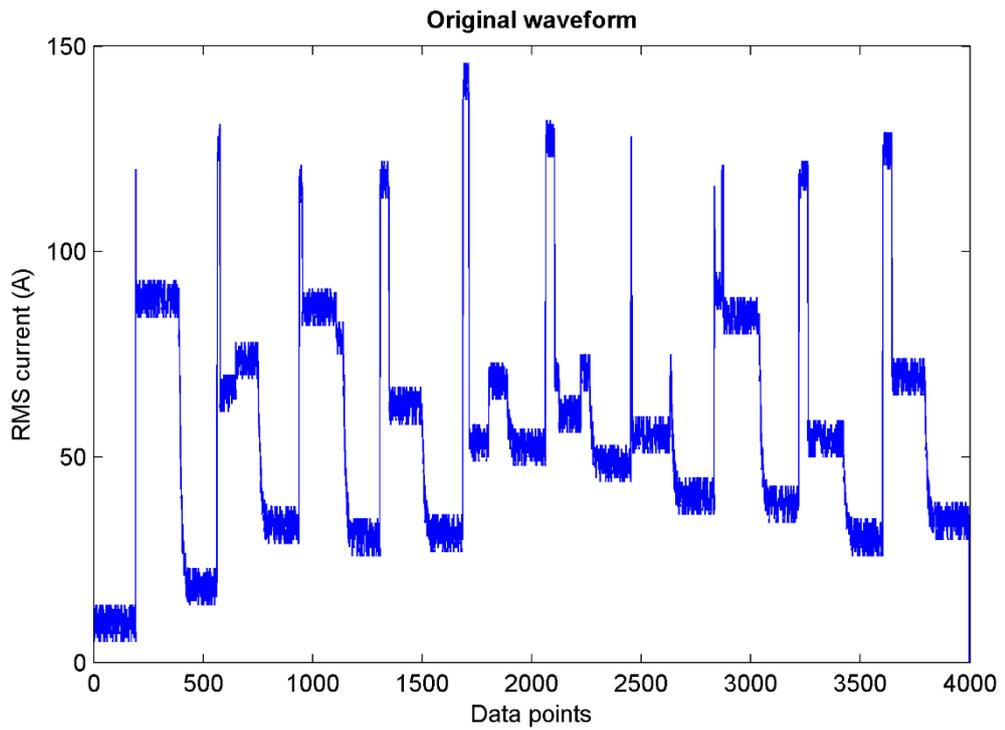


Figure 6.4 – Example original 4000 data point energy profile waveform.

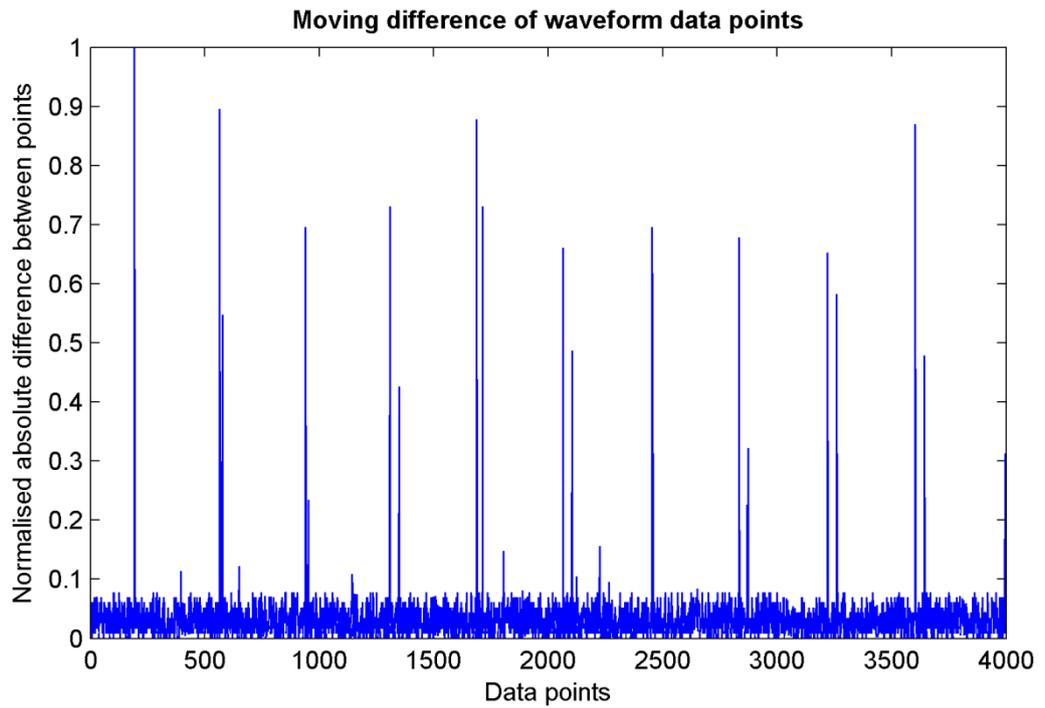


Figure 6.5 - Graph showing the absolute moving difference of data shown in figure 6.4.

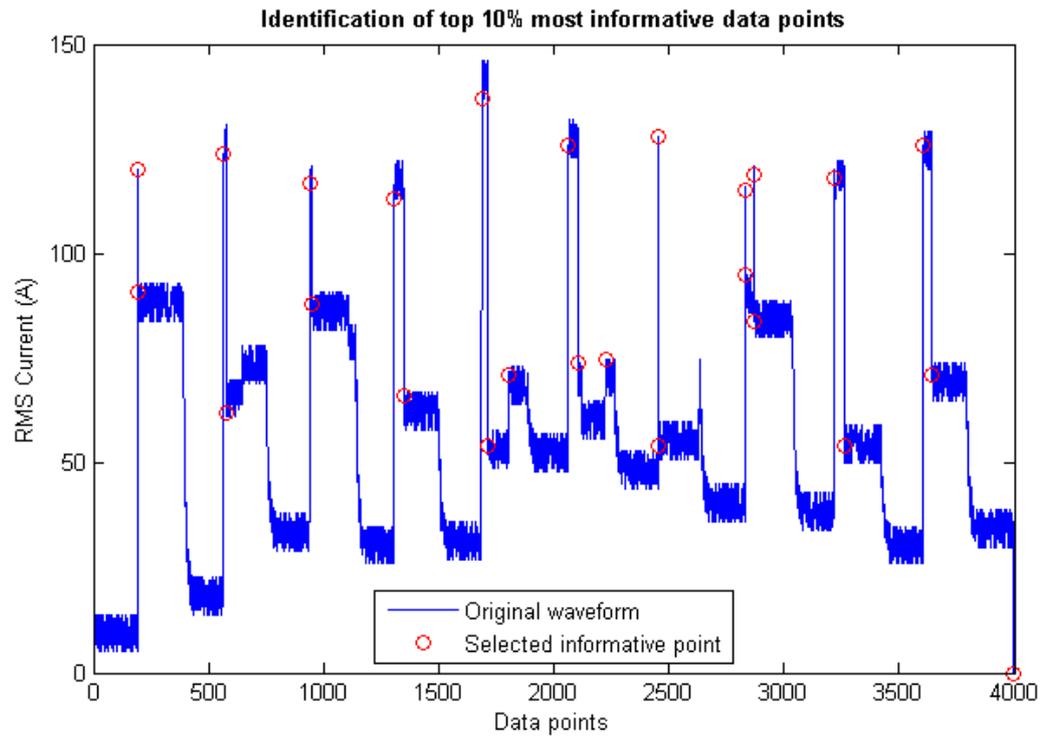


Figure 6.6 - Original waveform overlaid with identification of upper 10% of informative points.

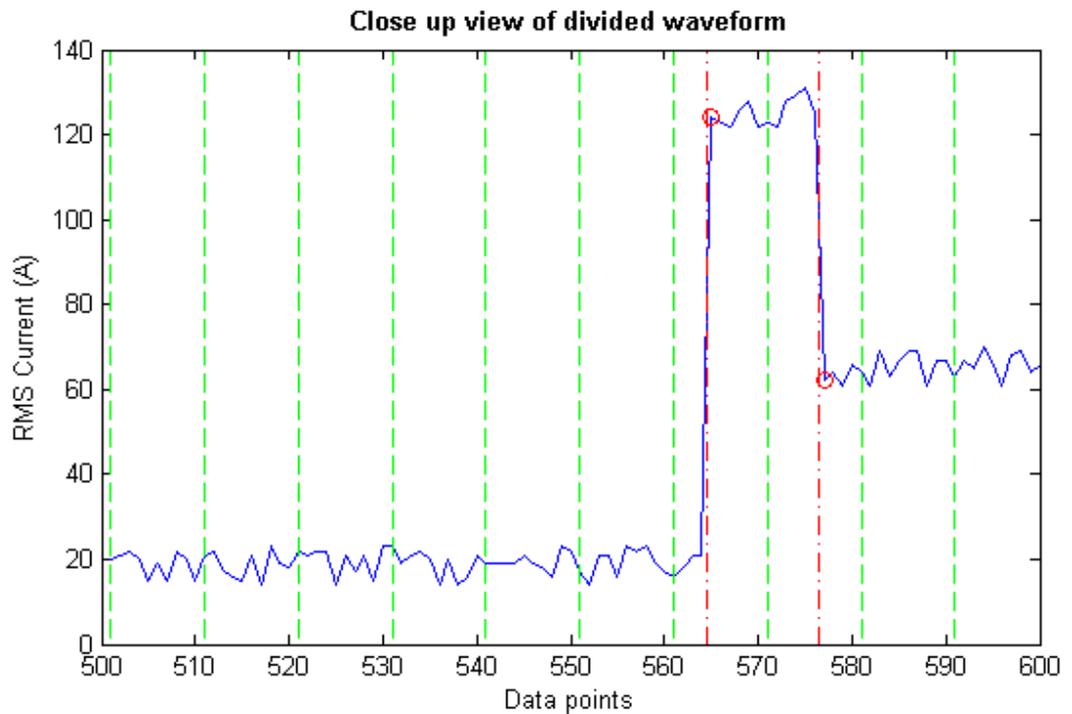


Figure 6.7 - Sample of waveform divided into blocks (green dashed lines) and waveform features (red dot & dashed lines).

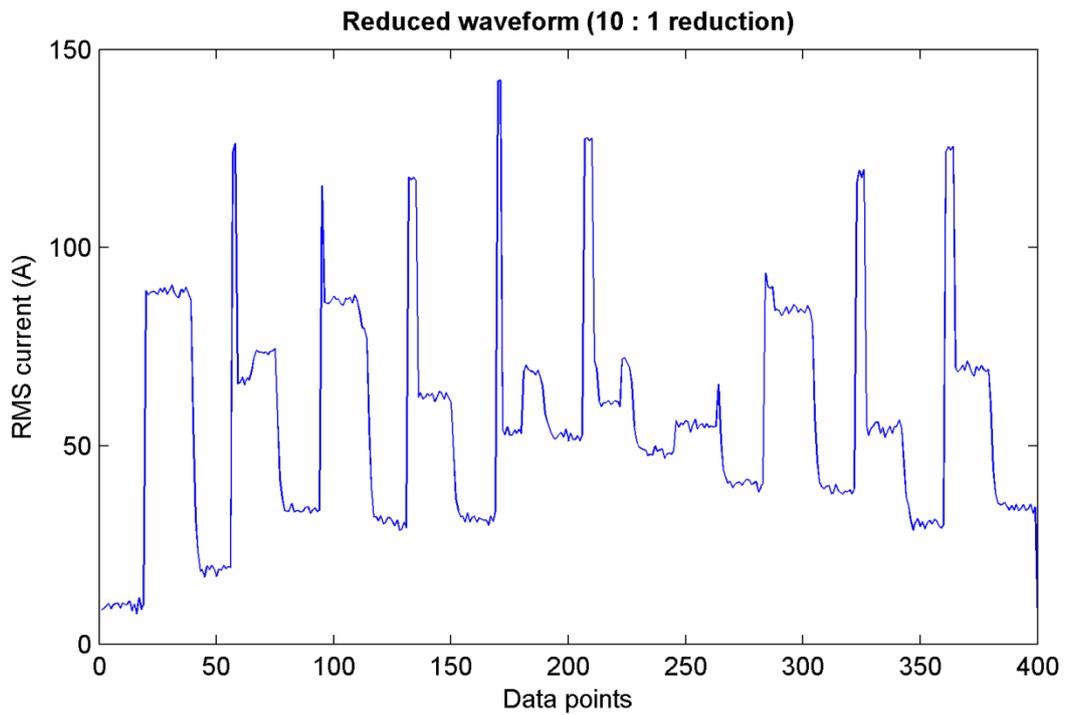


Figure 6.8 - Output waveform compressed from the original by a factor of 10.

Using the above method, any profile can be compressed by a fixed ratio which is dependent upon the block size B^{Size} . In this application, the block size was set to ten. Depending upon the shape of the waveform, this method will produce results identical to standard mean aggregation. However in cases where there are significant step changes, or where there are high amplitude peaks which only last for a short time, relative to the B^{Size} , this algorithm will produce a more accurate fit. A comparison against mean aggregation can be seen in figure 6.9.

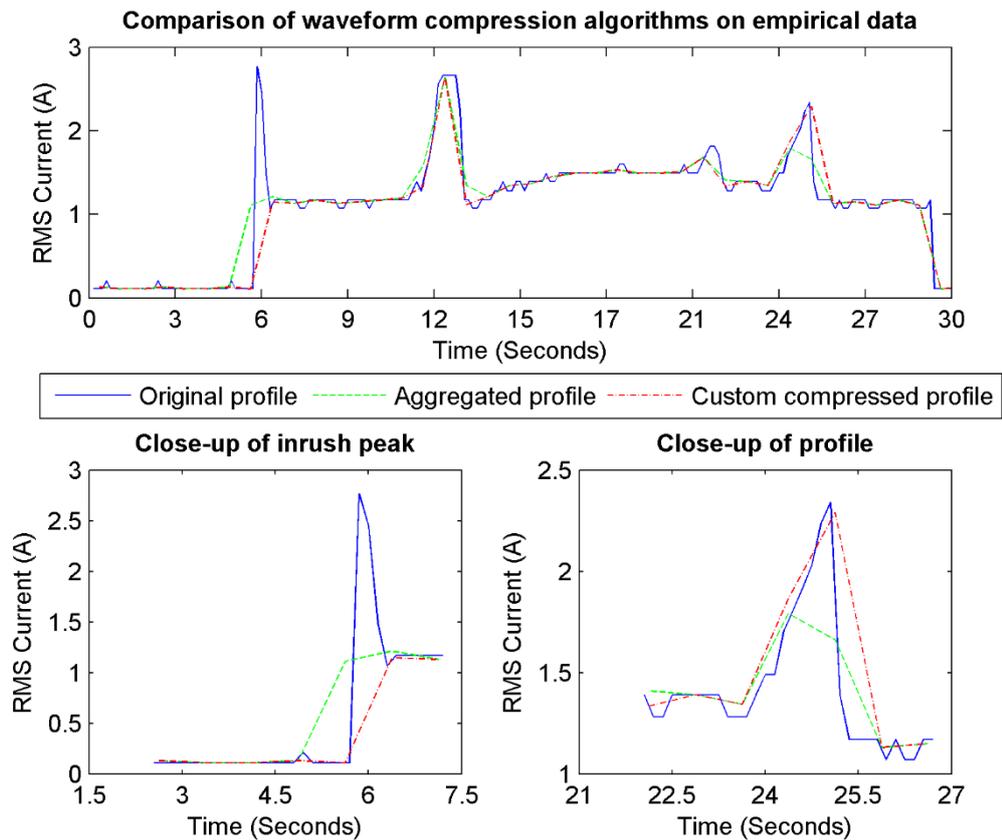


Figure 6.9 - Comparison of standard mean aggregation and the custom compression algorithm on empirical data. Compressed waveforms interpolated for comparison.

As can be seen in figure 6.9, the compressed waveform produced by the custom algorithm is able to follow the original waveform more accurately in parts of the waveform where there is a large amount of change. In both cases, the compression method essentially removes the inrush peak. As its period is only a few data points in length, it can be said that there is no method which can translate this through and still produce a faithful compression. Including it would require its period to be artificially increased. However the custom algorithm produces a slightly more faithful representation of the original. Tables 6.2 and 6.3 shows a comparison of errors over two compression ratios, produced by four different compression methods which closely fit the requirement specification. These include the custom compression algorithm, standard mean aggregation, down sampling and resampling. The results in tables 6.2 and 6.3 demonstrate that the custom compression algorithm and mean aggregation produce some of the most accurate compressions. Ultimately however, the error differences between the two methods are minor and inconsistent. This is because the feature filtering aspect of the custom compression

algorithm may only act upon a very small number of blocks, relative to the total number used to compress the waveform. Furthermore, because the custom algorithm can completely remove short period waveform features such as inrush peaks, the overall errors, especially the maximum error, can be higher. The uncompressed waveforms used here can be found in appendix D.

Table 6.2 - Table showing compression induced error from four compression techniques. Compressed waveforms are expanded using 'last known value' interpolation to compare against original. The compression ratio is 10:1.

	Custom algorithm error			Mean aggregation error			Down sampling error			Resampling error		
	Mean	Max.	RMS	Mean	Max.	RMS	Mean	Max.	RMS	Mean	Max.	RMS
Test waveform 1	0.096	10.556	0.400	0.101	9.854	0.393	0.132	14.090	0.581	0.110	11.517	0.429
Test waveform 2	0.595	3.127	0.836	0.560	2.492	0.744	0.645	3.630	0.943	0.545	2.989	0.745
Test waveform 3	0.262	4.869	0.366	0.224	4.119	0.302	0.279	4.910	0.418	0.241	4.844	0.327
Test waveform 4	4.683	17.500	5.588	4.597	13.900	5.362	6.033	19.000	7.801	4.752	13.184	5.572
Test waveform 5	1.052	7.712	1.660	1.168	7.391	1.644	1.384	8.010	2.239	1.151	7.098	1.622
Test waveform 6	0.031	0.063	0.036	0.031	0.056	0.036	0.056	0.113	0.067	0.127	0.900	0.230
Test waveform 7	0.113	0.346	0.148	0.063	0.175	0.080	0.113	0.346	0.148	0.112	0.346	0.148
Test waveform 8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.165	0.900	0.295
Test waveform 9	0.078	19.841	0.488	0.090	17.857	0.464	0.128	19.704	0.500	0.302	21.225	0.958
Test waveform 10	7.304	176.150	11.567	7.482	158.535	11.267	15.938	187.826	28.719	20.227	229.493	34.350
Average	1.421	24.016	2.109	1.432	21.438	2.029	2.471	25.763	4.142	2.773	29.249	4.468

Table 6.3 - Table showing compression induced error from four compression techniques. Compressed waveforms are expanded using 'last known value' interpolation to compare against original. The compression ratio is 20:1.

	Custom algorithm error			Mean aggregation error			Down sampling error			Resampling error		
	Mean	Max.	RMS	Mean	Max.	RMS	Mean	Max.	RMS	Mean	Max.	RMS
Test waveform 1	0.096	10.515	0.399	0.102	10.168	0.396	0.176	14.090	0.689	0.156	11.930	0.489
Test waveform 2	0.766	3.395	1.031	0.676	2.746	0.842	0.855	3.620	1.169	0.719	2.931	0.882
Test waveform 3	0.271	4.879	0.371	0.264	4.504	0.345	0.341	4.800	0.474	0.279	4.919	0.362
Test waveform 4	4.785	12.818	5.629	4.726	13.150	5.495	6.135	19.000	7.853	4.846	13.281	5.720
Test waveform 5	1.081	7.315	1.669	1.182	7.365	1.656	1.506	8.010	2.348	1.176	6.702	1.648
Test waveform 6	0.063	0.125	0.072	0.063	0.119	0.072	0.119	0.238	0.139	0.226	0.954	0.332
Test waveform 7	0.237	0.679	0.304	0.127	0.355	0.158	0.237	0.679	0.304	0.237	0.680	0.303
Test waveform 8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.259	0.961	0.428
Test waveform 9	0.143	19.659	0.502	0.153	18.676	0.491	0.252	19.321	0.557	0.595	21.016	1.395
Test waveform 10	17.259	200.045	30.734	17.413	115.279	25.102	29.693	220.211	42.175	37.364	235.051	55.052
Average	2.470	25.943	4.071	2.471	17.236	3.456	3.931	28.997	5.571	4.586	29.843	6.661

Conclusively, through the use of the custom compression algorithm, the length of all historical energy profiles can be reduced to decrease the optimisation systems runtime. However using this method will sacrifice information accuracy. The actual percentage of accuracy loss will be dependent upon the individual historical profiles themselves. A comparative analysis of the reduced accuracy prediction against the actual full prediction is discussed in section 7.7.

6.3.2 Additional Methods for Producing a Coarse Prediction

In section 6.1, work was carried out to determine the optimal termination conditions for the optimisation system, such that a suitably optimal solution can be found prior to the system terminating. While this is necessary for the actual optimisation process, the termination conditions could be adjusted in the hope that the system could locate a solution suitable to demonstrate the level of potential optimisation within a schedule. This would occur within a reduced time frame despite referencing the original full length historical energy profiles. Due to the genetic diversity issues, which necessitate the need to periodically regenerate the Genetic Algorithm population, it was deduced that the most appropriate parameter to adjust would be the population size. Utilising this method will sacrifice result optimality as the search performance of the Genetic Algorithm will be limited. Unlike using reduced length historical energy profiles however, this method will produce an accurate prediction as it references the original full length energy profiles. In terms of gains, the runtime of the optimisation system is non-deterministic due to invalid schedules not entering the relevant prediction engine in the fitness function and because of the fitness cache. However it can be approximated that reducing the population size will have a linear decrease on the systems runtime.

As both methods discussed thus far influence the runtime in different, independent ways, it is possible to combine them to produce a further reduction in runtime. The disadvantage of this is that the disadvantages of both methods – reduced prediction accuracy and less than suitably optimal solution being returned, are combined. However, depending on the production line manager's preference, the potential runtime reduction may be seen as a suitable gamble to

determine the level of potential optimisation within a schedule. A comparative analysis of all these methods is conducted in section 7.7.

CHAPTER 7

EXPERIMENTATION RESULTS AND ANALYSIS

Following its introduction and performance enhancements in chapters 5 and 6, the schedule optimisation system was thoroughly evaluated to determine its performance when presented with a range of manufacturing schedules. In this chapter the testing strategy is presented, along with the individual results and findings from the various features of the system, such as DMR.

7.1 Experiment Setup

The schedule optimisation system was tested on a set of six randomly generated schedules with a varying number of processes and jobs. Each of these schedules was initially generated for minimal makespan using an open-source production planning software which did not account for energy consumption (Frepple, 2015). Prior to the initial scheduling, process runtime constraints were set equal to the processes makespan plus, on average, 55% of extra time to allow for schedule flexibility. This aligns with current scheduling ideologies which allows for a degree of flexibility within the schedule should unanticipated issues arise. Each schedule was designed to operate on a single test production line which consists of ten machines which are left idling when not in use. Table 7.1 briefly details the original schedules. The full schedules themselves can be found in appendix E.

Table 7.1 - Details of original schedules and specified production line used to test the schedule optimisation system. Schedules optimised for minimal makespan only.

Description of test schedules			
	Number of jobs	Time allowed (hr:min)	Required processing time (hr:min)
Test Schedule #1		Eight job / three process schedule	
Process A	3	1:20	0:35
Process B	2	1:10	0:53
Process C	3	2:00	0:35
Test Schedule #2		Twelve job / three process schedule	
Process A	6	2:00	1:12
Process B	2	2:00	0:53
Process C	4	2:20	1:15
Test Schedule #3		Twenty job / four process schedule	
Process A	5	2:45	0:57
Process B	4	3:50	1:38
Process C	6	4:15	2:10
Process D	5	4:53	2:49
Test Schedule #4		Thirty job / three process schedule	
Process A	10	4:00	1:42
Process B	10	5:00	2:48
Process C	10	5:00	2:55
Test Schedule #5		Forty job / four process schedule	
Process A	10	3:30	1:42
Process B	10	5:00	2:48
Process C	10	5:00	2:55
Process D	10	5:00	3:51
Test Schedule #6		Fifty job / five process schedule	
Process A	10	3:30	1:42
Process B	10	5:00	2:48
Process C	10	5:00	2:55
Process D	10	5:00	3:51
Process E	10	5:00	2:03

While the custom energy monitoring system allowed for highly granular energy profiles to be recorded and stored in the Intelligent Library based system (section 3.3), recording them for a variety of manufacturing jobs was impractical within the scope of this project. To that end, for the experimentation phase, the library was entirely populated with artificially generated profiles. This further removed the need to generate synthetic profiles. Figure 7.1 shows the modified library used. The methodology used to generate the artificial profiles for the algorithm evaluation in section 3.3.2, was expanded to produce multiple profiles for the different jobs in the schedule. Profiles belonging to different jobs would be dissimilar, while the difference between profiles of identical jobs would be limited to that of the metadata's influence. This was insured by a combination of randomly generated and user input data.

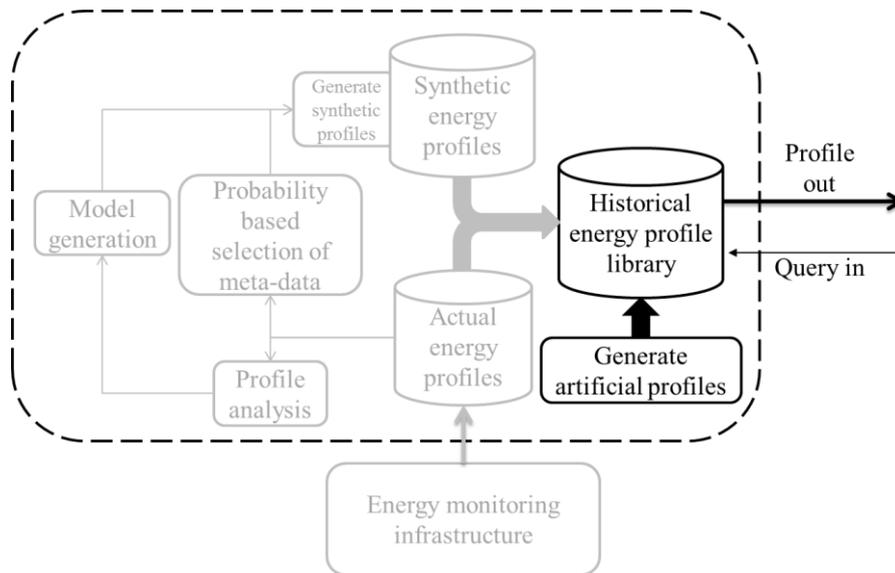


Figure 7.1 - Diagram showing experimental implementation of the intelligent library-based system.

The methodology for generating these artificial profiles is shown in figure 7.2. For each job, a progenitor profile is initially generated. The user specifies the number of waveform features such as step changes and the program places them semi-randomly throughout the profile. Constraints are in place where needed to ensure the final profile is consistent in overall shape with the empirical profiles seen in figure 2.9. Copies of this profile are then made and metadata is randomly generated for each. The profiles are then modified based on the metadata with the user controlling which portions of the profile are influenced by each metadata value. Figure 7.3 shows a profile generated via this method. In comparison with an actual profile, an example of which can be seen, such as figure 2.9, the elemental structure of the two profiles is similar. To maintain consistency with empirical profiles recorded using the custom energy monitoring system (section 2.3), all artificial profiles were generated at the same temporal granularity - 150ms.

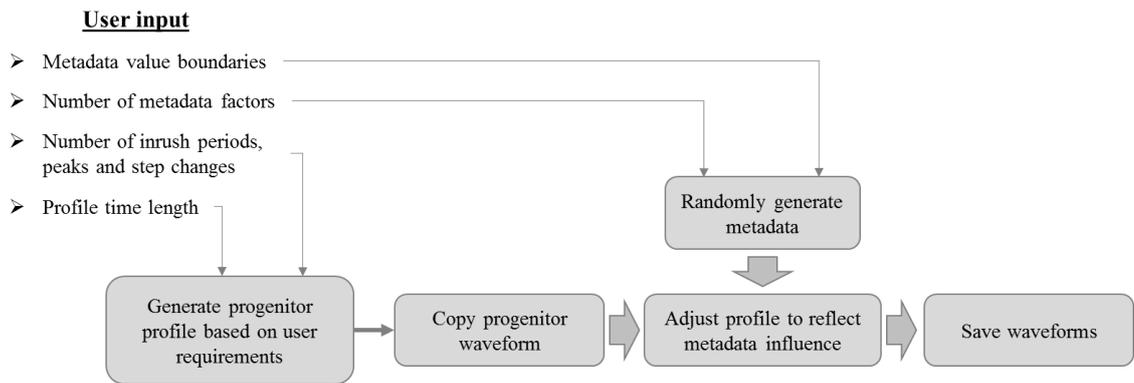


Figure 7.2 - Breakdown of the artificial energy profile generation software.

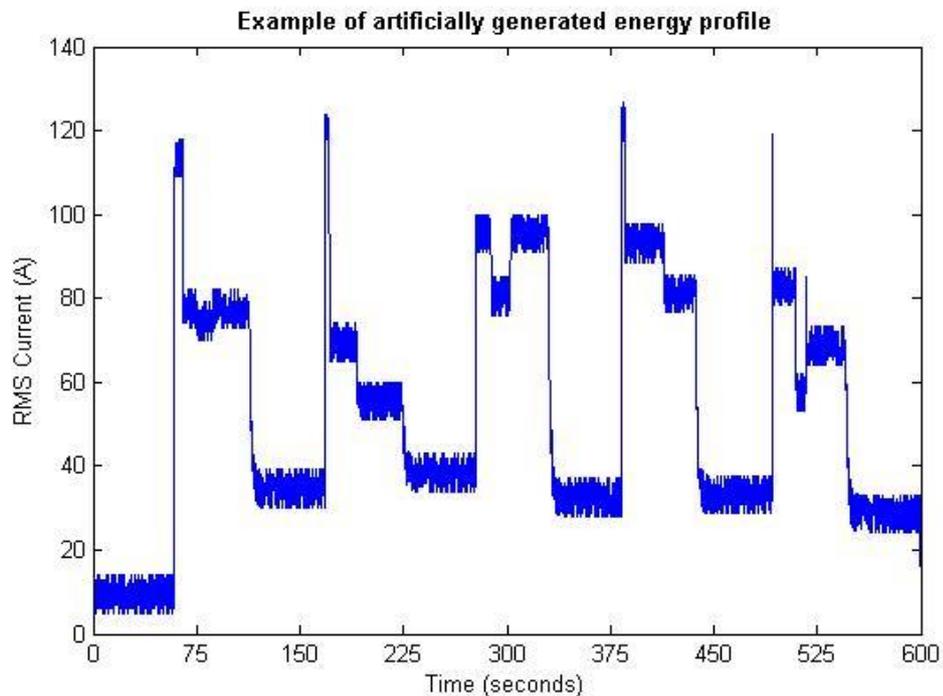


Figure 7.3 - Example of an artificially generated energy profile.

On average twenty profiles were generated per job. Each job was subsequently associated with these historical profiles through randomly generated machine influence data compatible with the profiles metadata, and the Master-Log-File. Similarly, the available production line resources were also given compatible random machine status data. Once complete, the original un-optimised energy consumption variances for each schedule could be calculated, and the schedules could be optimised.

7.2 Production Schedule Optimisation – Overview of Results

The schedule optimisation system was applied to each of the test schedules described in table 7.1 with the minimal time shift value T , set to one minute. During initial testing, each job in the schedule was not supplied with a list of alternative machines. This effectively disabled Dynamic Machine Reassignment in an effort to purely analyse the search performance and capabilities of the Genetic Algorithm within the system. The optimisation results for both most probable and best case predictions for each of the test schedules can be seen in tables 7.2a and 7.2b.

Table 7.2a – Most probable results of the schedule optimisation system (Results produced over ten runs. Percentage of optimisation is relative to the original variance).

Test schedule	Original variance	Optimised variance			
		Observed minimum	Observed maximum	Average	Average % of optimisation
#1	3938.64	1064.10	1083.44	1066.54	72.92%
#2	2052.46	1464.10	1464.10	1464.10	28.67%
#3	3678.66	1092.96	1408.44	1227.18	66.64%
#4	3207.71	1134.54	1436.41	1282.83	60.01%
#5	4511.61	1570.43	2377.28	1857.59	58.83%
#6	5710.03	2339.29	3159.20	2610.59	54.28%

Table 7.2b - Best case results of the schedule optimisation system (Results produced over ten runs. Percentage of optimisation is relative to the original variance).

Test schedule	Original variance	Optimised variance			
		Observed minimum	Observed maximum	Average	Average % of optimisation
#1	3938.64	1014.80	1019.88	1015.31	74.22%
#2	2052.46	1296.78	1296.78	1296.78	36.82%
#3	3678.66	973.58	1258.01	1089.14	70.39%
#4	3207.71	992.90	1284.17	1131.83	64.72%
#5	4511.61	1325.01	2085.69	1596.83	64.61%
#6	5710.03	1939.72	2678.22	2165.25	62.08%

Tables 7.2a and 7.2b demonstrate that the optimisation system is able to reduce the variance on all test schedules by a significant percentage in both the most probable and best case predictions. It should be noted however that the level of reduction is not simply a factor of the amount of jobs within the schedule. In fact, it is probable that the level of reduction is not a consequence of a singular factor, but the complex interactions between the individual process and overall schedule constraints. Figures 7.4, 7.6 and 7.8 to 7.11 show a graphical representation of the production line energy profiles for each test schedule. These show that the

optimisation system takes full advantage of the allowed time which was not originally utilised by the original scheduling algorithm. Additionally, figures 7.5 and 7.7 show that when time permits, the schedule will be altered to prevent multiple jobs running concurrently. This will produce the lowest attainable energy consumption variance for the schedule, not considering the variance reductions related to using different historical job energy profiles.

The primary downside to this optimisation system is that the energy consumption variance is reduced at the cost of an increased total processing time. With the exception of schedule #2, each of the test schedules expands their total processing time to the full timespan allowed by the constraints (see table 7.3). Typically this is seen as highly disadvantageous given that traditional production planning algorithms strive to minimise it. With that said, it can be seen in figure 7.5 and 7.7 that there are periods of inactivity during the total processing time. A naive algorithm could be employed as a post-processing methodology to temporally shift a set of jobs up the timeline to remove this inactivity period. An example of this could be applied to test schedule #1 (figure 7.5), having all jobs within process C shifted upwards by seven minutes with only minor changes to the profiles variance. Even with this, the overall total processing time is still significantly expanded. Manufacturers can control this potential expansion by adjusting the process priorities; however this will limit the level of variance minimisation. As such, it can be seen as the manufacturer's decision regarding which objective is most important. The schedule constraints can be adjusted by the manufacturer to reflect their current needs.

Table 7.3 – Difference in makespan between the original and most probable schedules.

Test schedule	Original makespan (hr:min)	Most probable makespan (hr:min)	Makespan increased by a factor of...
#1	1:05	2:10	2
#2	3:25	3:25	1
#3	5:00	5:50	1.17
#4	4:55	7:00	1.42
#5	6:10	7:00	1.14
#6	7:10	7:50	1.10

In general, the energy consumption of all test schedules is redistributed throughout the allotted time for each process, and it is this which lowers the overall variance and in addition, the peak demand. In actuality, it can be observed that in every test schedule the peak demand is reduced

(see figures 7.4, 7.6, and 7.8 to 7.11). The most significant change occurs between the original profile and the most probable and best case profiles. Overall, changes between the latter two profiles are minor, reinforced by the results in tables 7.2a and 7.2b. This is likely due to the fact that a majority of the difference between the probable and best case profiles comes from the differences in the individual job energy profiles, and not from the schedule alteration itself. This is supported by the results in figure 7.5, where there is no change to the schedule between the most probable and best case results, despite an average difference in energy consumption variance of 51.23.

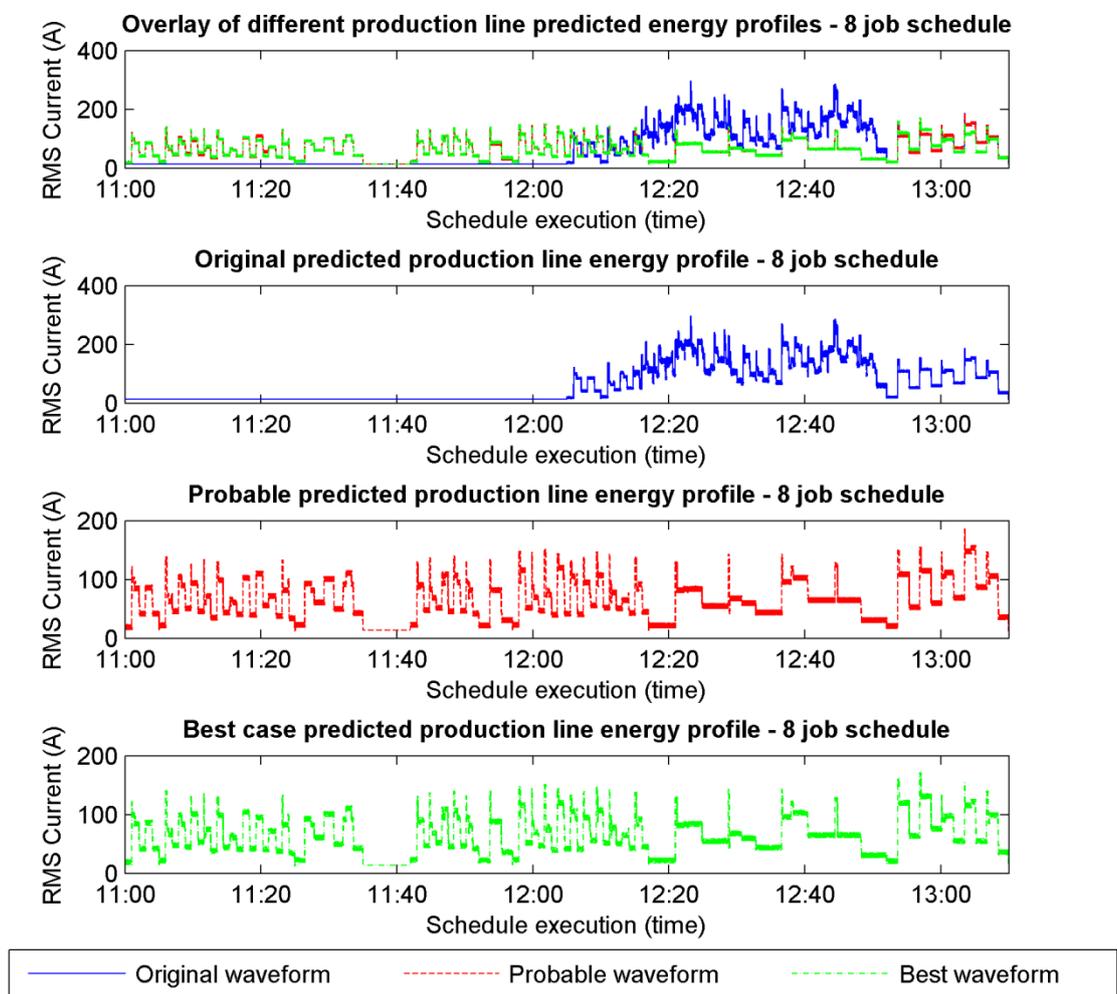


Figure 7.4 - Graphs of individual and combined predicted production line profiles for an 8 job schedule (Test schedule #1).

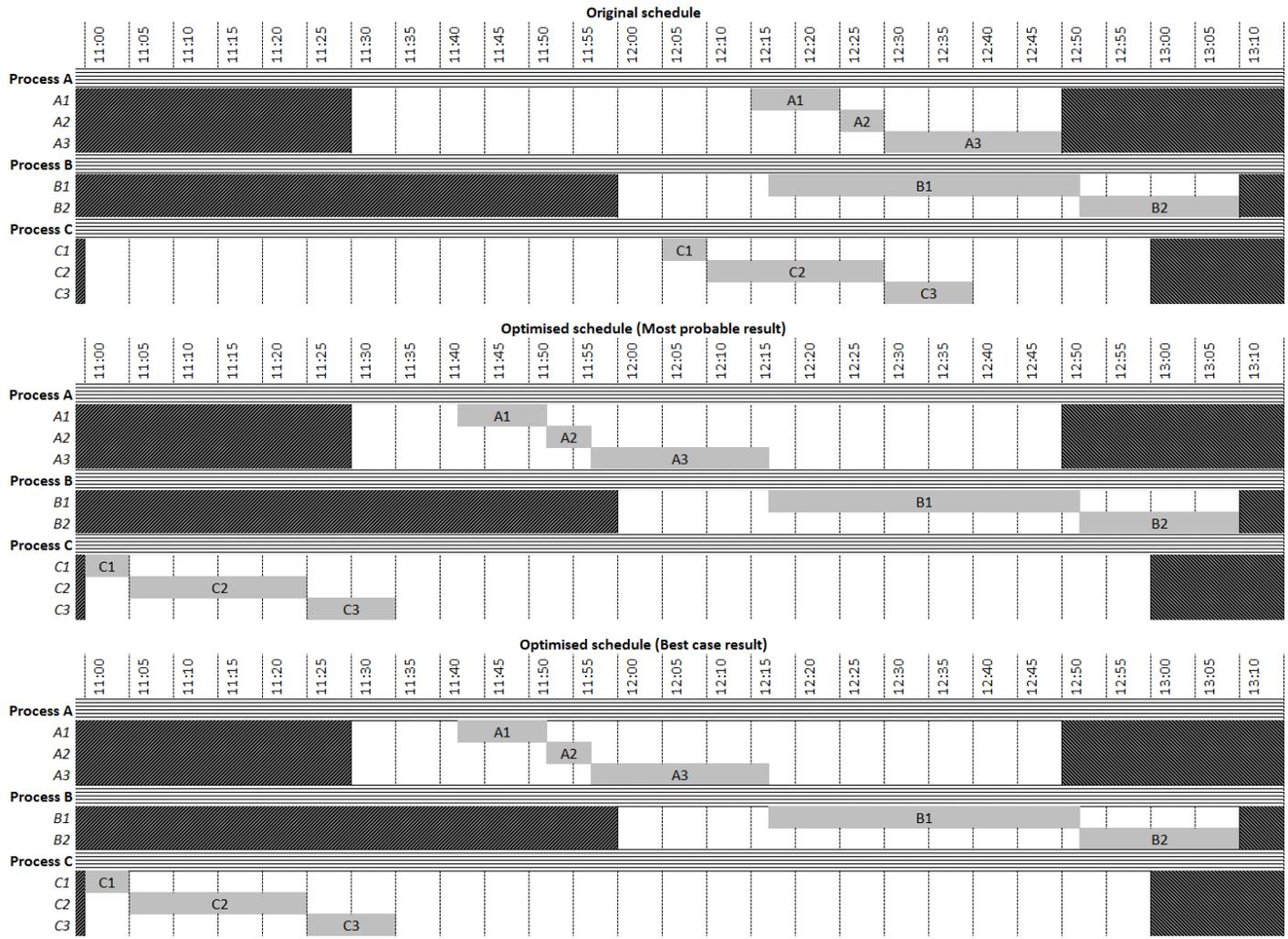


Figure 7.5 - Gantt chart for 8 job schedule (Test schedule #1) showing the job layouts for the original, most probable and best case results.

In the case of test schedule #2, the result is consistently lower than the other test schedules, and this is true for both the probable and best case results. Viewing its energy consumption profile in figure 7.6, reveals that the optimisation system has not fully exploited the available time despite the fact that the schedule indicates that process C is able to execute from 08:00 to 10:20. The gantt chart for this schedule in figure 7.7a/b shows why. In the original schedule, only two jobs run concurrently (A2/A3 and B2). By adjusting the start time of jobs A1 and B2, the optimisation system is able to prevent this, the theoretical lower bound in energy consumption variance is found. This explains the smaller variance reduction, as the schedule can be considered near-optimal to begin with. It should be noted that even when there are no concurrently running jobs, these may still not be the optimal job start times due to the differences in post-job idling currents. It can be assumed however that this will only have a small effect overall, as the manufacturer is unlikely to leave machines idling in a configuration which results in a significantly higher than normal energy demand.

Figures 7.8 to 7.11 show the predicted production line energy profiles for test schedules #3 to #6. The gantt charts for these schedules can be found in appendix F.

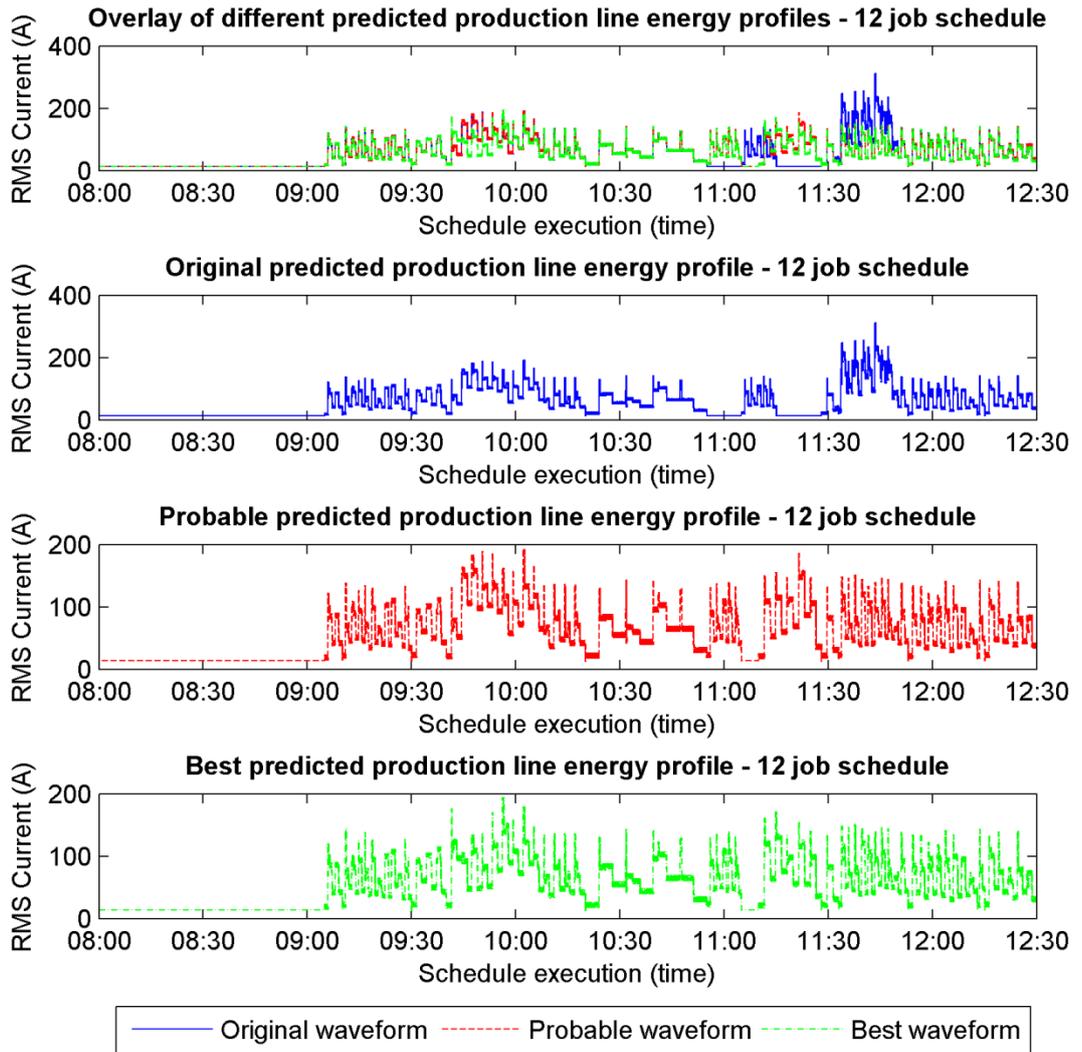


Figure 7.6 - Graphs of individual and combined predicted production line profiles for a 12 job schedule (Test schedule #2).

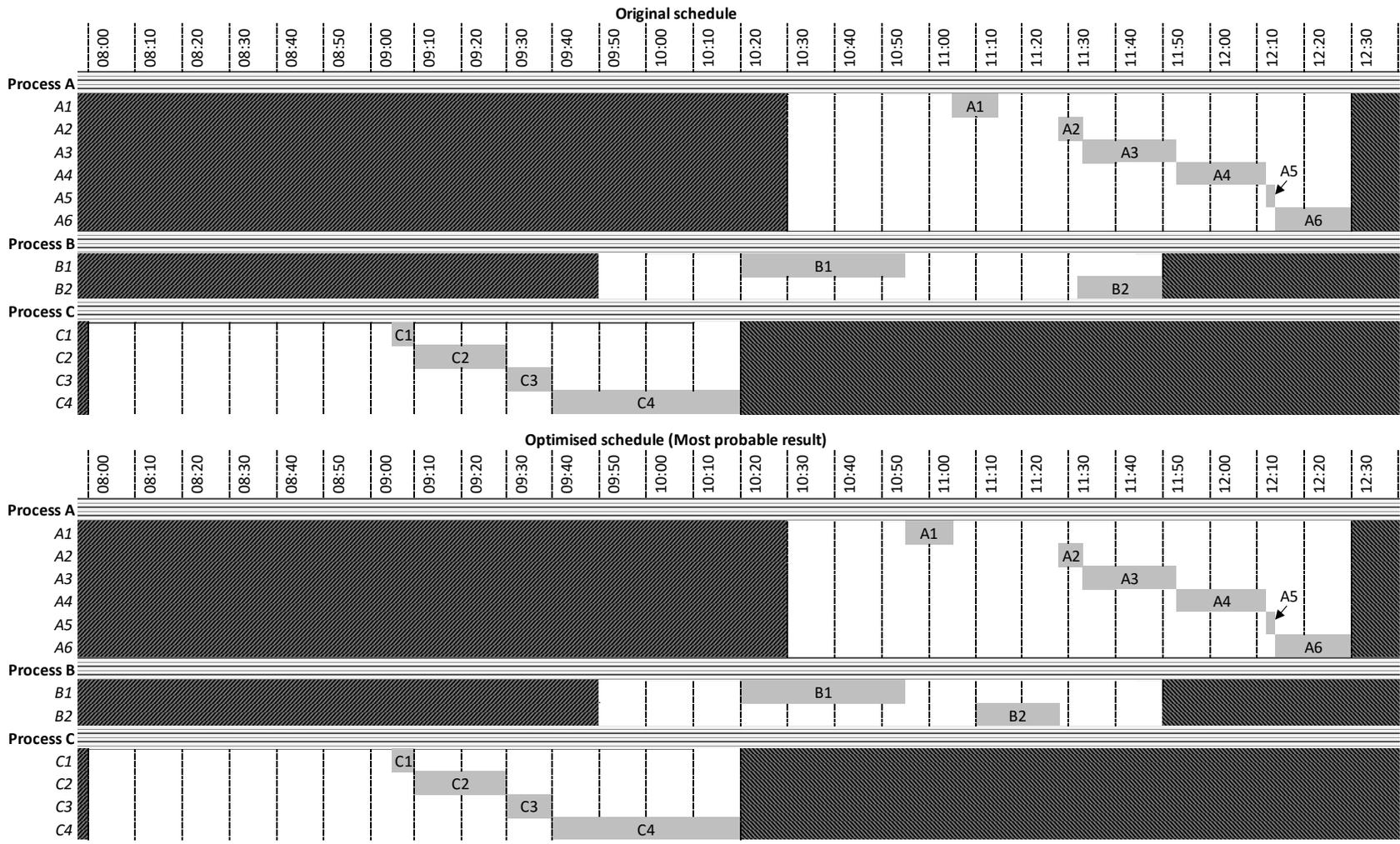


Figure 7.7a - Gantt chart for 12 job schedule (Test schedule #2) showing the job layouts for the original and most probable results.

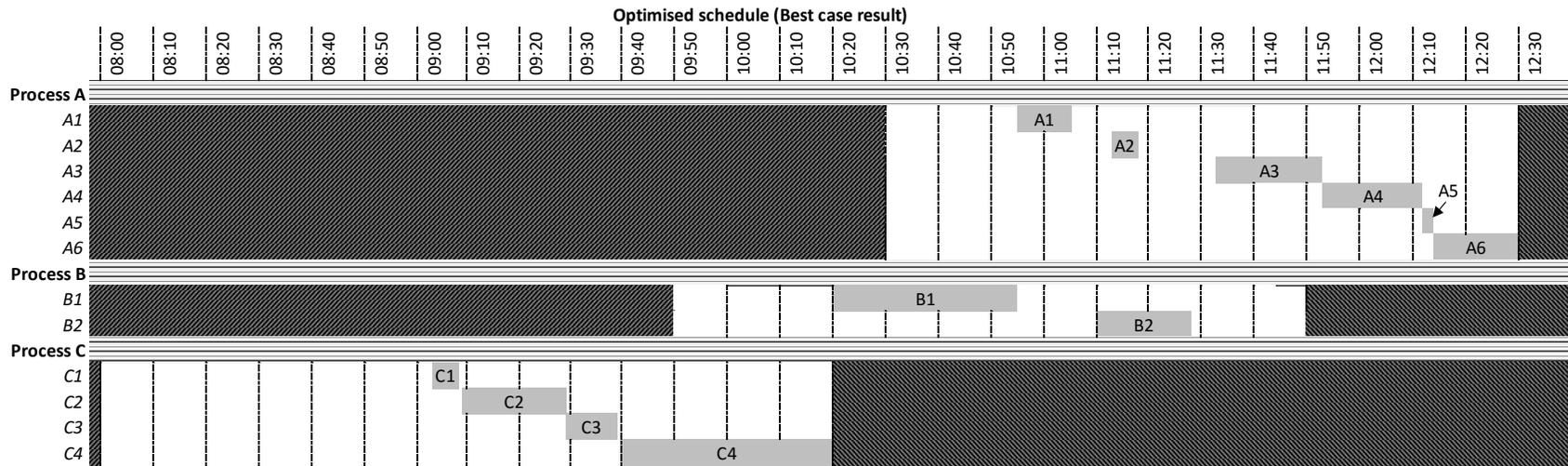


Figure 7.7b - Gantt chart for 12 job schedule (Test schedule #2) showing the job layouts for the best case results.

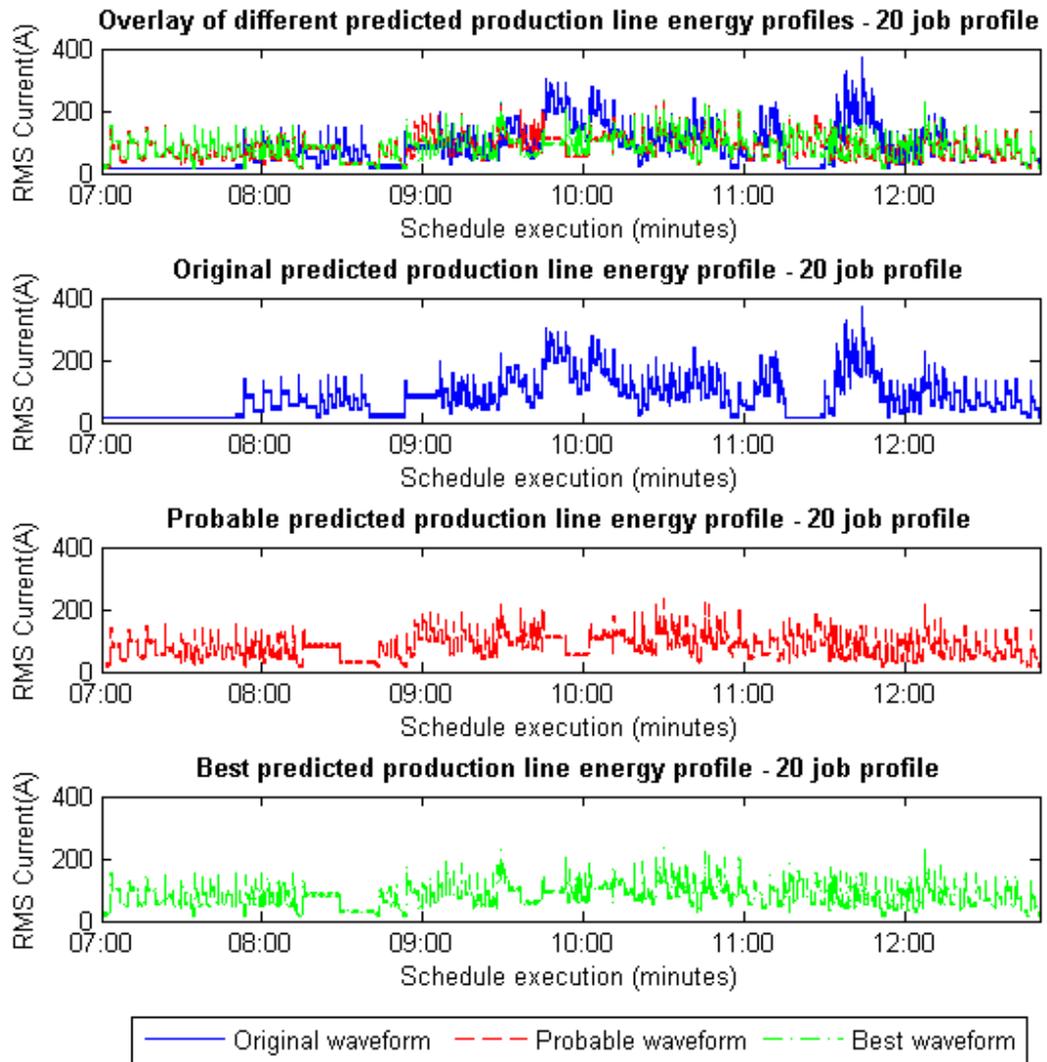


Figure 7.8 - Graphs of individual and combined predicted production line profiles for a 20 job schedule (Test schedule #3).

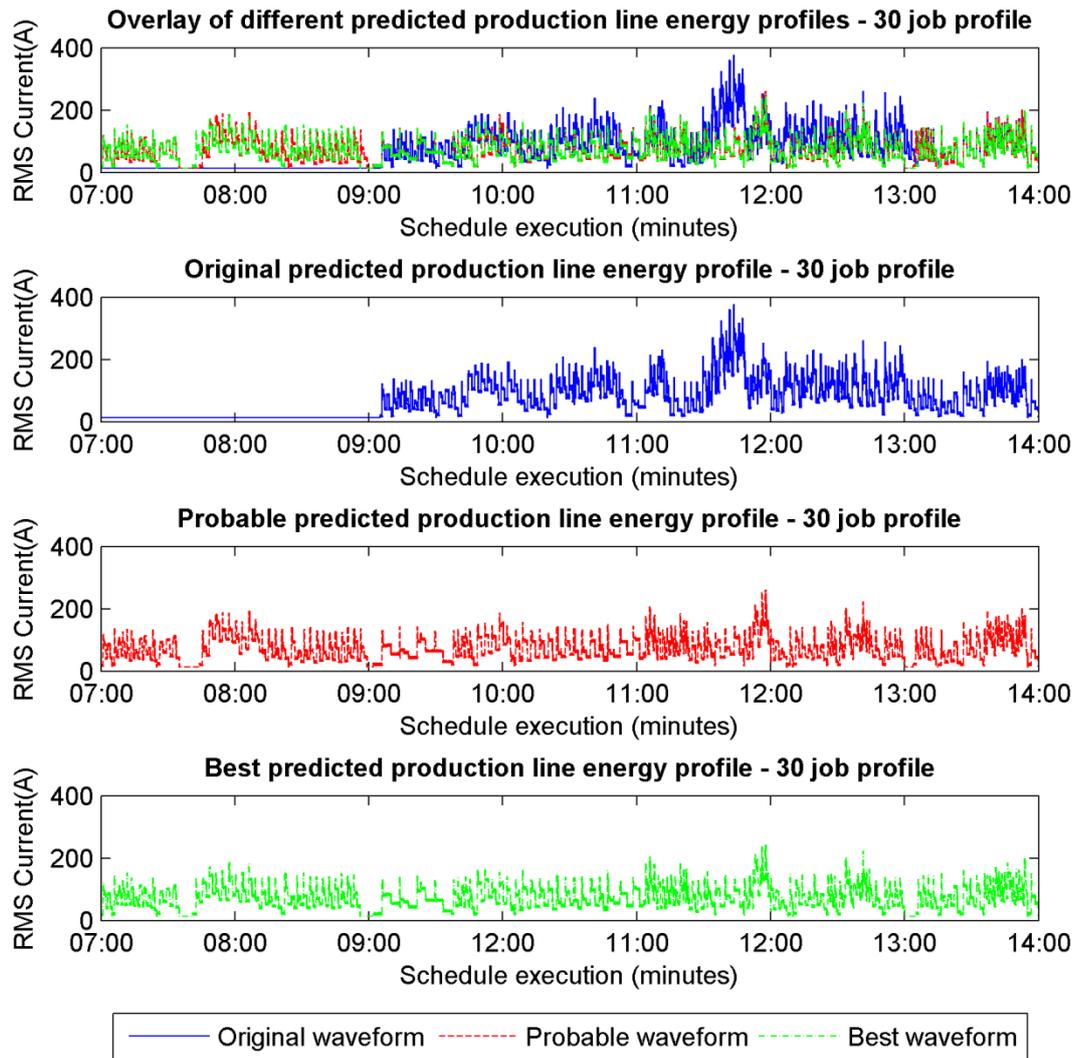


Figure 7.9 - Graphs of individual and combined predicted production line profiles for a 30 job schedule (Test schedule #4).

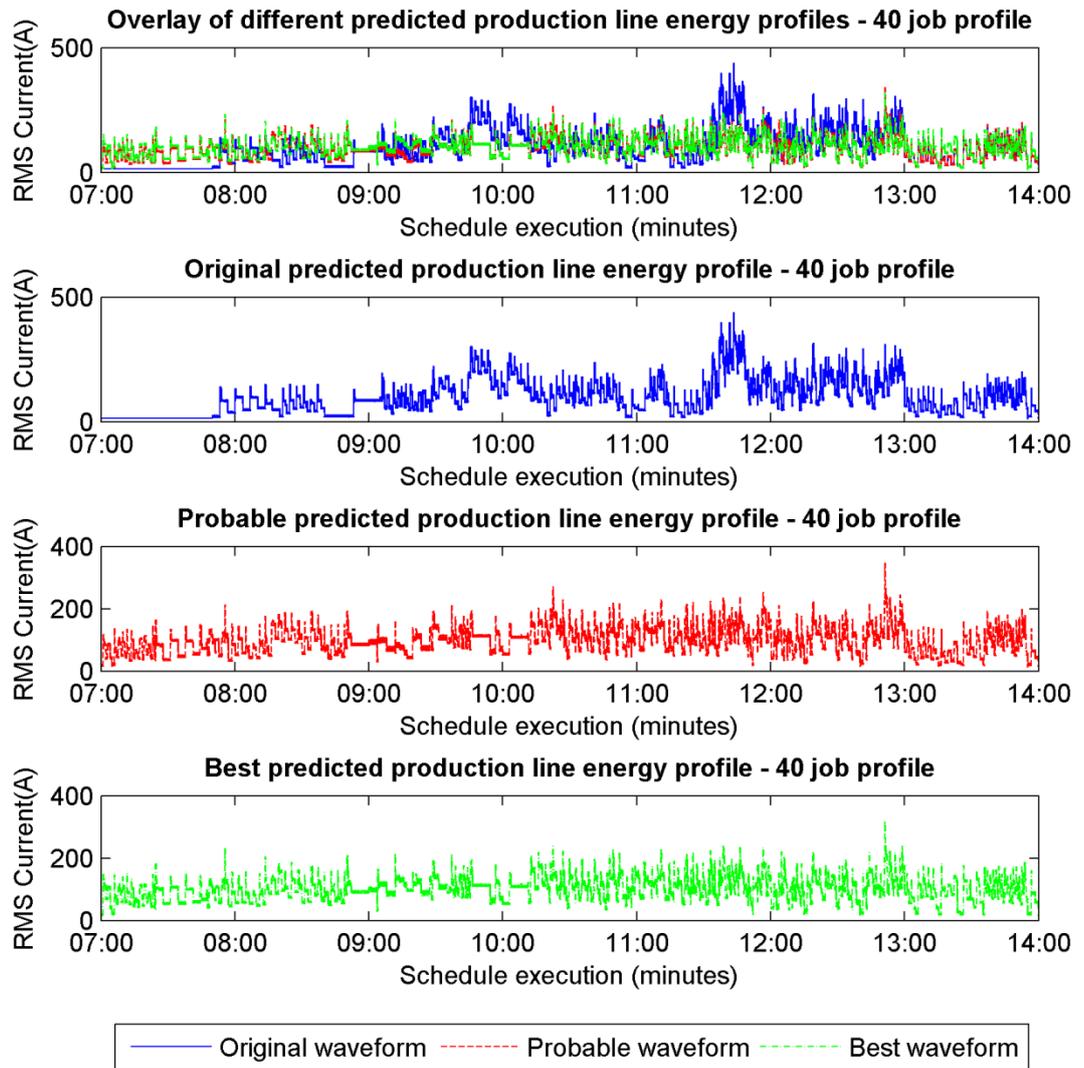


Figure 7.10 - Graphs of individual and combined predicted production line profiles for a 40 job schedule (Test schedule #5).

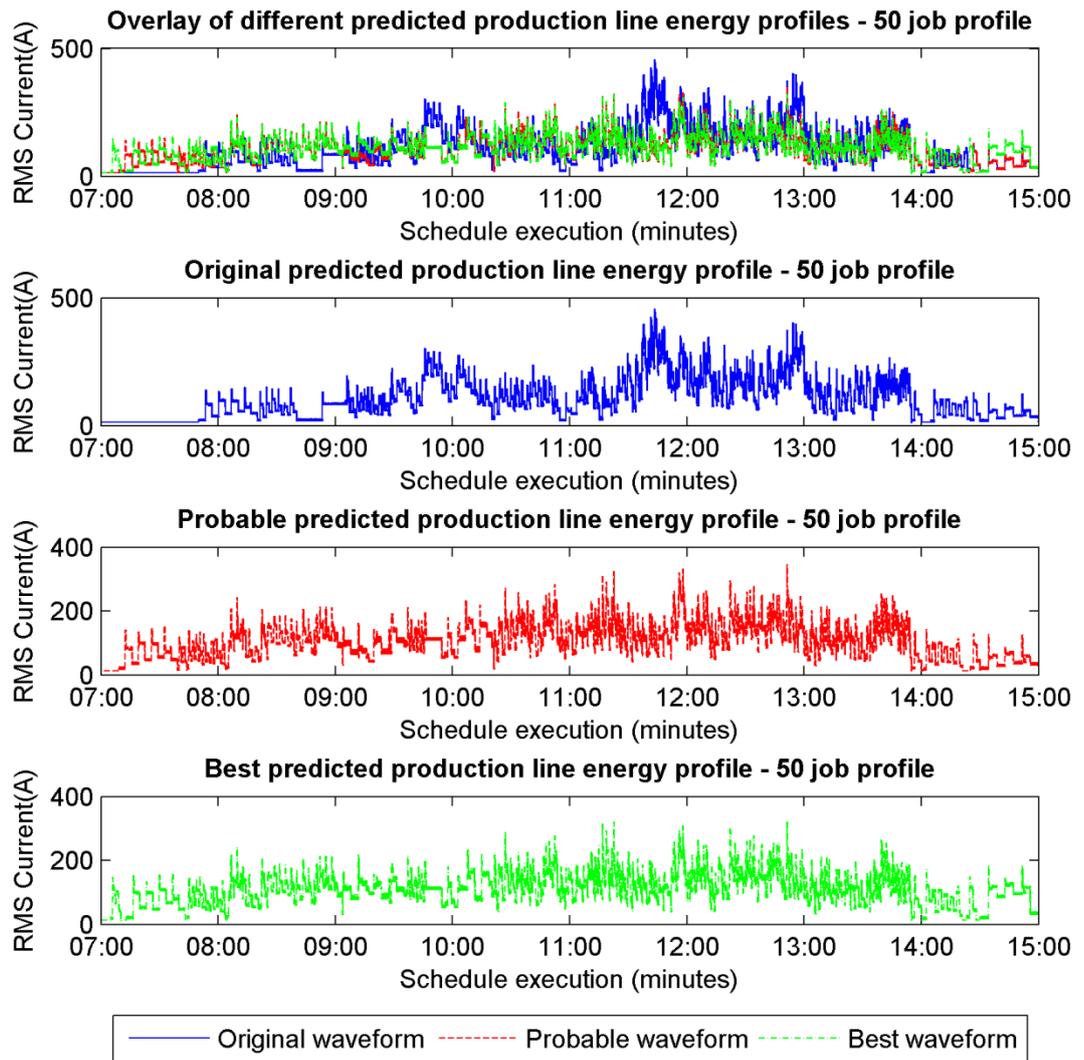


Figure 7.11 - Graphs of individual and combined predicted production line profiles for a 50 job schedule (Test schedule #6).

7.3 Production Schedule Optimisation – Performance Analysis

Overall the most probable and best case results, as seen in tables 7.2a and 7.2b, appear to follow a gradually descending trend as the size of the schedules increase. This is the case for all test schedules with the exception of schedule #2. While this trend could be related to the global optimal values for each of the schedules, this cannot be analysed as the global optimal values are not known. Of course, as the schedules increase in size, the problem’s complexity will increase exponentially. This fact combined with the problem of invalid schedules means the optimisation system may return local minima. While system features such as Genetic Algorithm mutation and the refreshing of the population work to prevent this from occurring, it is believed that the issues with invalid schedules could result in an extremely inhospitable search space. For small schedules (i.e. test schedules #1 and #2), these features are mostly sufficient in preventing the optimisation system from becoming trapped in local minima. As schedule complexity and the search space increases however, this ability cannot be fully maintained. This theory is reinforced by the results in table 7.4. As the schedules increase in size, the optimisation system returns slightly different results over consecutive runs indicating that results returned are local minima. Therefore, the level of potential variance reduction in test schedules #3 to #6 could be further improved.

Table 7.4 - Table showing the range (max. – min.) of results returned by the optimisation system in 5 consecutive runs.

Schedule name	Most probable result		Best case result	
	Range	Percentage of original variance	Range	Percentage of original variance
Test Schedule #1	19.34	0.49%	5.08	0.13%
Test Schedule #2	0.00	0.00%	0.00	0.00%
Test Schedule #3	315.47	8.58%	284.43	7.73%
Test Schedule #4	301.87	9.41%	291.27	9.08%
Test Schedule #5	806.85	17.88%	760.69	16.86%
Test Schedule #6	819.91	14.36%	738.51	12.93%

The reason for the varied results can be explained through the findings presented in figure 7.12. As the schedule complexity increases, the ratio between valid and invalid schedule chromosomes in the Genetic Algorithm’s population shifts significantly in favour of the invalid schedules. This was to be expected to a certain degree. As schedules increase in size, the

probability of a candidate schedule representing a valid schedule will decrease as the validation standard becomes increasingly rigorous thanks to the additional constraints. Unfortunately, this means that in the case of test schedules #4 to #6, the average number of valid schedules in a population is less than 10%. As a result, the Genetic Algorithm will spend a large majority of its time investigating invalid regions of the search space. It is important to note that the actual ratio of valid and invalid schedules within the entire search space cannot be known and as a result, a 10% valid schedule ratio within the population may be a suitably sized representation for these test schedules.

Solving this problem could be as simple as adjusting the optimisation system's termination conditions and allowing it to run for a longer period. However this would obviously increase the optimisation system's runtime. In figure 7.13 it can be seen that the optimisation system has an inconsistent runtime when operating on different schedules. Additionally, the trend over the range of test schedules approximately follows that of the valid ratio percentage in figure 7.12. This can be explained by the design of the fitness function schedule validity checks (see section 5.3). Inside this, if a candidate schedule is invalid, it is assigned a maximum fitness value and spends little time in the fitness function relative to a valid schedule, which triggers the prediction engine to generate a predicted production line energy profile. This means that when the Genetic Algorithm population has a high concentration of invalid candidate schedules, the overall runtime of the optimisation system will be substantially reduced. Table 7.5 helps to contextualise this by demonstrating the difference in fitness function runtime between a valid and invalid schedule. This, combined with the findings from figure 7.12 translates to a significant difference in the overall systems runtime.

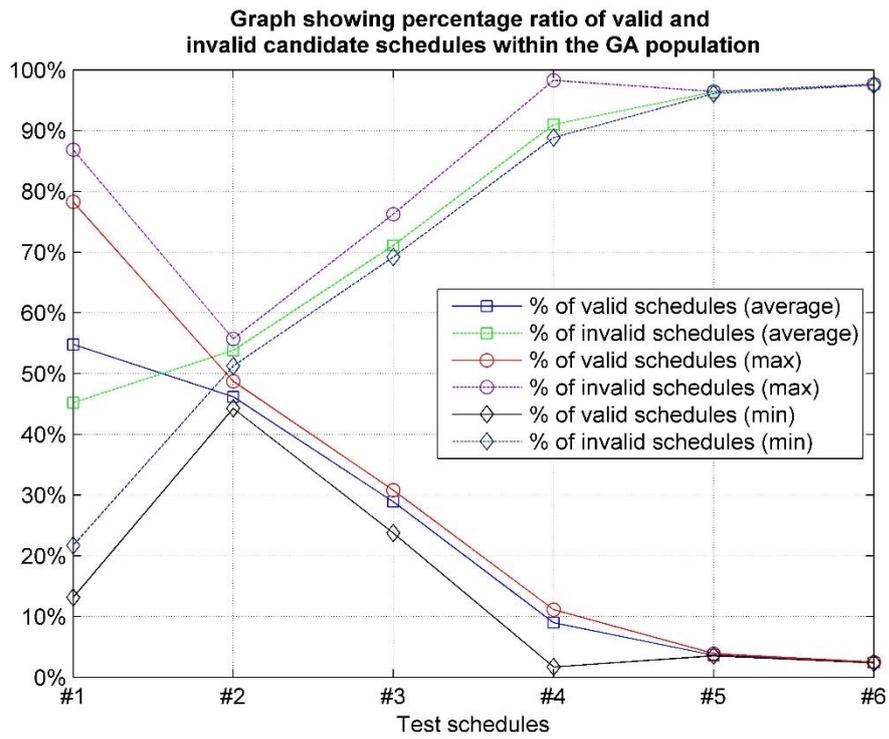


Figure 7.12 - Graph showing the percentage ratio between valid and invalid schedules in the Genetic Algorithm's population during the optimisation of different schedules (Results produced using most probable prediction engine over six consecutive runs).

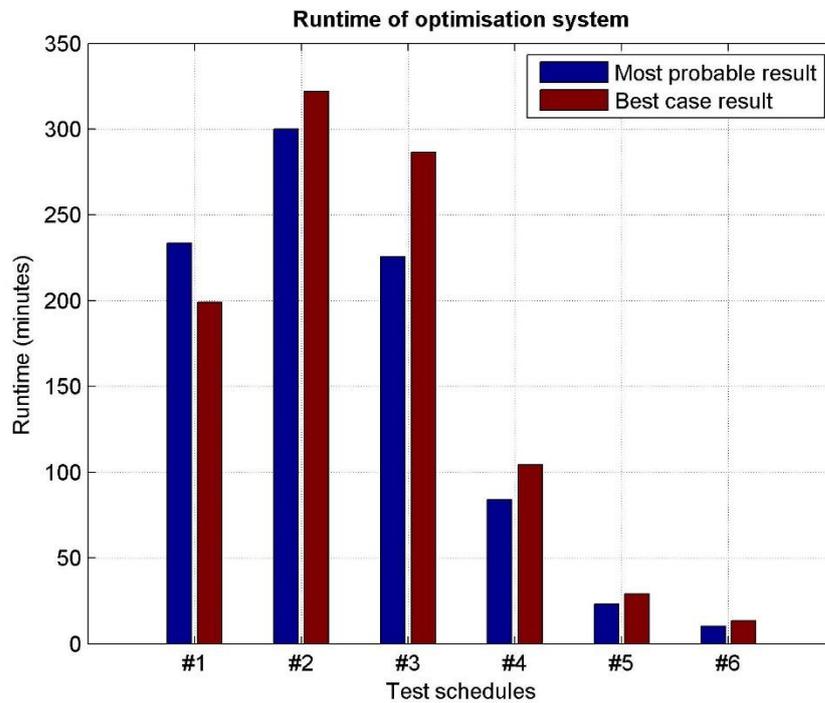


Figure 7.13 - Graph showing the runtime of the optimisation system (Results averaged over 5 consecutive runs).

Table 7.5 - Differences in fitness function evaluation runtime between valid and invalid candidate schedules for the range of test schedules (Results averaged over 100 iterations; Ran on an intel i7 8GB Windows 8.1 computer).

Test schedule	Valid schedule evaluation time (μ s)	Invalid schedule evaluation time (μ s)
#1	5500.29	1.60
#2	11726.03	1.30
#3	15466.02	1.57
#4	19589.72	1.86
#5	20909.70	3.19
#6	26280.67	3.09

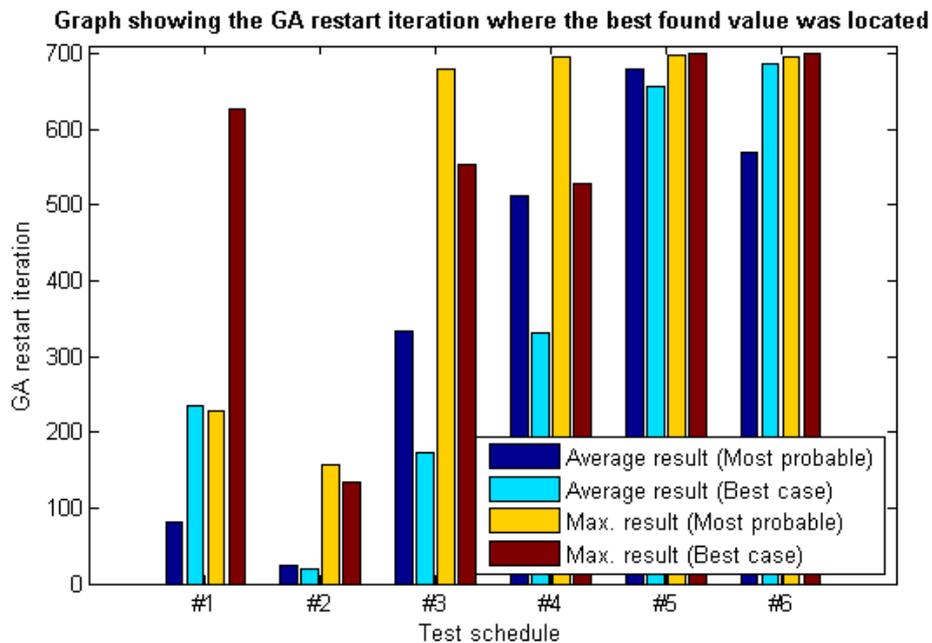


Figure 7.14 - Graph showing the GA restart iteration where the best found value was located (Results averaged and maximum value located through ten consecutive runs).

Figure 7.14 shows that for test schedules #1 to #3, the best found result for both the most probable and best case scenarios are located significantly early in the optimisation system's execution. This can potentially be attributed to the number of jobs within the schedule, as a reduced number of jobs will result in a reduced search space, thereby increasing the probability of the best found result being located early. As such it may be viable to introduce variable termination conditions which are proportionate to the number of jobs. Overall, this would maximise runtime efficiency while minimising the sacrifice to search performance. With that said, figure 7.14 demonstrates that while the average number of iterations required by test schedules #1 to #3 is small in comparison with the current termination conditions, the variability can be significant. This is seen in the best case result for test schedule #1 where the maximum

observed result is approximately 625 restarts of the Genetic Algorithm. This demonstrates that there is the potential for a sacrifice in performance resulting from dynamic termination conditions.

Overall, the average runtime of the optimisation system can be significant with a maximum recorded runtime of approximately 5 hours and 40 minutes for test schedule #2 running on an Intel i5 8GB Windows 7 machine. Table 7.5 shows that this is primarily due to the time required to construct the predicted machine energy profiles and compile them into a production line profile. The granularity of the historical energy profiles, while increasing prediction accuracy, also results in the need to manipulate large amounts of data potentially hundreds of thousands of times during the system's operation. Additional performance issues arise from the use of highly granular data, such as the inefficient use of automatic memory management. As C# is a memory managed programming language, a garbage collection component is used to release allocated memory which is no longer referenced. All allocated memory is divided and placed into one of three 'generations', numbered zero to two, depending on the allocation size and lifespan (Windows Dev Center, No date). In the case of the optimisation system, the constructed machine and production line energy profiles for each candidate schedule can be considered short-lived objects as they only exist within the prediction engine functions. As a result they should be sorted within generation zero of the garbage collector, which is processed frequently and has a fixed size. However any object which is greater than 85,000 bytes is assigned to generation two. This has no size restrictions, is processed less frequently, and has a potentially lengthy processing time relative to the other generations, due to it mainly containing a number of large, long-lived objects.

In the case of this optimisation system, with an historical energy profile granularity of 150ms, even the machine and production line energy profiles for the smallest test schedule, test schedule #1, will consist of 52,000 data points each. As these are represented by single-precision data type arrays, each element of which consists of four bytes (32-bit), the total memory size for each profile will be 208,000 bytes. Hence the related objects will always be

assigned to generation two; the end result being that the overall runtime of the system will be inhibited by the lengthy garbage collection process.

7.4 Comparison of Most Probable and Best Case Results

As stated in section 5.3.2, the purpose of producing a best case result is to approximate the lower bound of the schedules production line energy consumption. This can then be compared against the most probable result to determine how close this is to the approximate lower bound. Table 7.6 shows the differences between a most probable and best case result, using test schedule #3 as an example. The results show that, in this case at least, minimal changes are made to the actual job start times, providing evidence that the most probable start times are already suitably close to their lower bound. On the contrary, significant changes are made to the energy profiles which are actually referenced for each job, with 14 of the 20 jobs having their referenced energy profile changed. As seen in tables 7.2a and 7.2b, for test schedule #3 the average difference between the most probable and best case variance is 3.75% of the original. While this may seem like a minor change, the maximum average change observed in table 7.2a and 7.2b is that for test schedule #2, with a difference of 8.15%.

Table 7.6 - Table comparing the job start times and energy profiles used in the most probable and best case results for test schedule #3. Changes are shown in bold.

Jobs	Most probable result		Best case result	
	Job start time	Energy profile	Job start time	Energy profile
Process A				
A1	09:35	P1A-001-2-11-43	09:52	P1A-001-2-11-43
A2	11:13	P1A-002-13-18	11:13	P1A-002-13-18
A3	11:31	P1A-003-102	11:31	P1A-003-158
A4	11:51	P1A-004-15	11:51	P1A-004-57
A5	12:12	P1A-005-22-31	12:12	P1A-005-22-31
Process B				
B1	10:07	P1B-001-23-70	10:07	P1B-001-23-70
B2	11:15	P1B-002-67-6	11:15	P1B-002-32-192
B3	12:05	P1B-003-44-2	12:05	P1B-003-96-8
B4	12:45	P1B-004-44	12:45	P1B-004-123
Process C				
C1	07:29	P1C-001-24-23	07:29	P1C-001-18-166
C2	07:53	P1C-002-35-15	07:53	P1C-002-96-43
C3	08:42	P1C-003-146	08:42	P1C-003-175
C4	08:52	P1C-004-36-10	08:52	P1C-004-34-461
C5	10:20	P1C-005-2	10:20	P1C-005-2
C6	10:55	P1C-006-70-11	10:55	P1C-006-1-160
Process D				
D1	07:00	P1D-001-20-42	07:00	P1D-001-64-129
D2	07:33	P1D-002-14-3	07:33	P1D-002-78-108
D3	08:02	P1D-003-25	08:02	P1D-003-25
D4	09:19	P1D-004-12	09:19	P1D-004-126
D5	10:42	P1D-005-40-9	10:42	P1D-005-31-31

In the case of an actual manufacturing production line, the differences between these two results could be further analysed to determine how the production line can potentially be operated, such that the most probable result can align more closely with the best case result.

7.5 Analysis of Dynamic Machine Reassignment Influence

As stated at the beginning of this chapter, Dynamic Machine Reassignment (DMR) was initially disabled to produce the baseline results discussed in the previous sections. To evaluate the influence DMR has on the level of potential optimisation within a schedule, each machine in the production line was cloned. This allowed for two of every machine, permitting each job a single alternative should a conflict arise on its default machine. Using this expanded production line, each of the test schedules was evaluated with both DMR disabled and enabled to garner a comparison which can be seen in figure 7.15. Due to the necessary expansion of the production

line, the results presented here will not match the other results discussed thus far in this chapter. As such it is necessary to consider these results in isolation.

Figure 7.15 demonstrates that in the case of test schedules #1 to #5 DMR has little influence on both the most probable and best case results. For schedules #1 and #2, it can be deduced that, based on their Gantt charts (figures 7.5 and 7.7), the time constraints allow for one job to be executed at once, thereby ensuring no machine conflicts arise. In the case of test schedules #3 to #5, it is possible that there is no benefit in terms of variance reduction in reassigning conflicting jobs, as minimising energy consumption variance by its very nature resists running multiple machines concurrently. The only schedule where DMR is seen to have an effect is test schedule #6. As the largest schedule tested, this was the most likely to benefit from DMR. Table 7.7 demonstrates the effect DMR can have on a schedules job-machine assignment. In the case of the original, most probable and best case schedules, processes A to C are unaffected. This is likely due to the fact that the schedule validity checking (see section 5.3) routine checks for machine conflicts in job order beginning with process A. As a result, jobs belonging to earlier processes will likely be able to reserve their default machine.

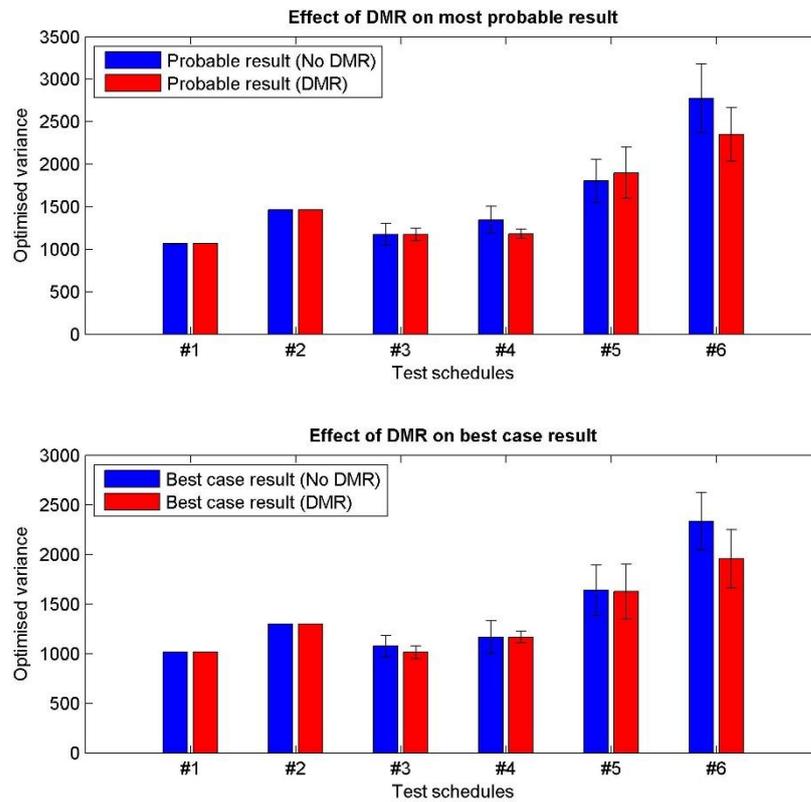


Figure 7.15 - Graph showing influence of DMR on the most probable and best case results (Results averaged over ten consecutive runs).

Table 7.7 - Table showing the job - machine assignments for the original, most probable and best case schedules of test schedule #6. Changes to machine assignment are shown in bold.

Job	Machine assignment		
	Original schedule	Most probable schedule	Best case schedule
Process D			
D1	M1A-001	M1A-001	M1A-001
D2	M1A-002	M1A-012	M1A-012
D3	M1A-009	M1A-009	M1A-009
D4	M1A-010	M1A-020	M1A-020
D5	M1A-007	M1A-007	M1A-007
D6	M1A-004	M1A-004	M1A-004
D7	M1A-001	M1A-001	M1A-001
D8	M1A-008	M1A-008	M1A-008
D9	M1A-010	M1A-020	M1A-020
D10	M1A-008	M1A-008	M1A-008
Process E			
E1	M1A-005	M1A-015	M1A-005
E2	M1A-006	M1A-006	M1A-006
E3	M1A-003	M1A-013	M1A-013
E4	M1A-004	M1A-004	M1A-004
E5	M1A-005	M1A-005	M1A-005
E6	M1A-003	M1A-003	M1A-003
E7	M1A-002	M1A-012	M1A-012
E8	M1A-008	M1A-008	M1A-008
E9	M1A-009	M1A-009	M1A-009
E10	M1A-006	M1A-006	M1A-006

In table 7.7, only six job-machine reassignments occur throughout the most probable and best case predictions. However on average, these result in a significant reduction in returned result. While its effectiveness is clear, the usefulness of DMR will be heavily dependent upon the particular schedules to be optimised and the flexibility and design of the production line.

7.6 Experimental Implementation

To fully test the optimisation system's ability and accuracy, it was applied to a small-scale schedule in a lab based production line.

7.6.1 Implementation Setup

The implementation involved the use of an original ten job schedule along with four devices, listed below, to represent the energy consumption for each job.

- A 550W 230V AC motor,
- An industrial dust collector containing a 750W AC motor,
- A 24 litre air compressor
- A 400W floodlight

While not manufacturing machines, they would generate suitable energy profiles which could be associated with each job in the schedule. These particular devices were selected because from an energy perspective, they closely resemble the components and load types of modern manufacturing machinery. Three of the four devices utilise an AC motor – the most common component of manufacturing machinery. The floodlight represents a resistive load, which is also comparable to a heat-based process.

As with the simulation based experiments, the schedule was originally produced using production planning software before being associated with pre-recorded energy profiles and given to the optimisation system. Both the original and most probable optimised schedules were

then executed with the energy profiles of the machines being recorded via the custom monitoring system. A best case result could not be produced, as there was only one energy profile available for each job. Therefore in this case it can be construed that the optimised schedule was both the most probable, and the best case. The results are shown in the next section. The full schedules along with their Gantt charts can be found in appendix G.

7.6.2 Implementation Results

Figure 7.16 shows a visual comparison between the predicted and recorded energy profiles for the original and optimised schedules.

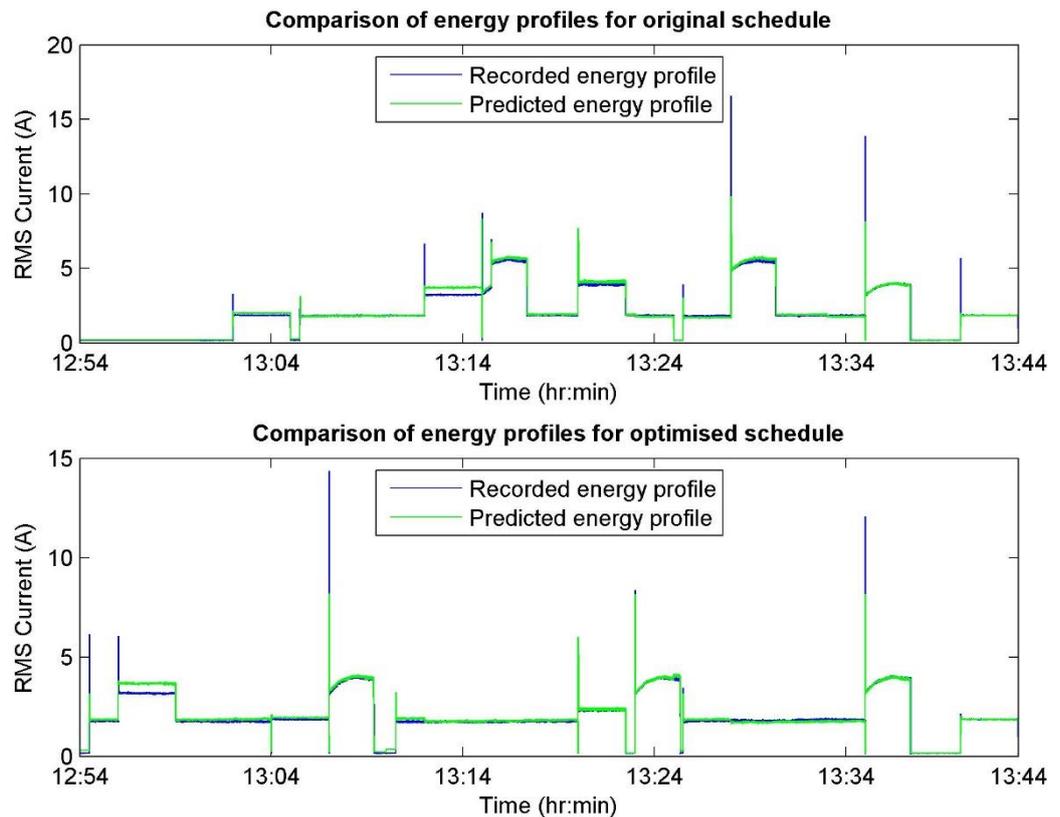


Figure 7.16 - Comparison between predicted and recorded energy profiles for the original and optimised evaluation schedule.

As can be seen in both the original and optimised schedules, the predicted and recorded profiles match each other to within an acceptable degree of tolerance. This is further confirmed by the

results in both table 7.8 and 7.9, where only small errors are seen. There are only two noticeable differences between the predicted and recorded values:

1. The peak values recorded during the inrush peaks – this can be explained by the fact that the custom energy monitoring system, while possessing a fast reporting rate, still may miss the peak values of short lived inrush periods. This results in the large maximum error seen in table 7.9.

Table 7.8 - Comparison between predicted and recorded variance values for the evaluated schedule.

Schedule	Predicted Variance	Recorded Variance	Error (%)
Original	2.495	2.323	7.404
Optimised	1.022	0.960	6.458

Table 7.9 - Error values between the predicted and recorded energy profiles.

Schedule	Absolute mean error (Arms)	Max Error (Arms)	RMS error (Arms)
Original	0.110	14.850	0.274
Optimised	0.102	14.70	0.260

2. There is a consistent error which occurs in both schedules, being seen at around 13:14 in the original schedule profile and at around 12:56 in the optimal schedule profile. An analysis of the Gantt charts reveal that this error coincides with the running of the floodlight however it only occurs when the floodlight is operated consecutively with the AC motor. This error is never seen again in the profiles despite the floodlight and AC motor being used twice again, which indicates that the recorded profiles are accurate. However, the floodlight and AC motor are never used at the same time again, meaning the error maybe a result of the two running simultaneously. In figure 7.17, a Kyoritsu KEW 6315 power quality analyser was used to monitor the power factor while the floodlight and AC motor were ran independent and simultaneously. As can be seen, when ran individually, the motor and floodlight have two very different power factors. The motor, representing an inductive load, lowers the power factor to approximately 0.5 while the light, a resistive load, has little influence. When ran together, the overall power factor is approximately 0.9. This is believed to be due to the fact that the light may represent a small capacitive load; hence the power factor not being one when it is

ran. As a result of all this, the power factor when both the motor and floodlight are running is better than when the motor is running solely. A better power factor results in a reduced current consumption which explains why the recorded current consumption is less than that of the predicted when the floodlight and motor are running. As the energy models used for the prediction do not account for power factor, the optimisation system is unable to account for this.

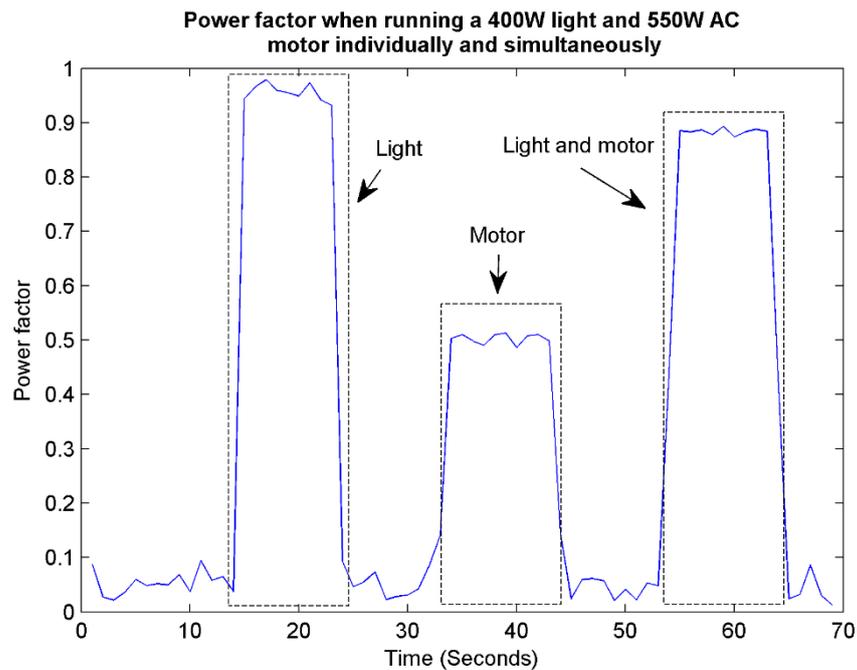


Figure 7.17 - Comparison of power factor when using a floodlight and AC motor independently and simultaneously. (Note: When no devices are using power, the meter is unable to measure power factor and hence produce erroneous results around zero).

Overall, this experimental implementation proves that the optimisation system works in a real world scenario and is able to predict the energy consumption of a production line to a high degree of accuracy. The main disadvantage to the accuracy is that within the scope of this project, the models do not account for power factor. However, this is a limitation of the energy monitoring setup and not of the optimisation system.

7.7 Coarse Prediction – Results and Analysis

As discussed previously and seen in figure 7.13, the runtime of the overall optimisation system can be long. It is for this reason that a series of coarse accuracy methodologies, introduced in section 6.3, were developed for the purpose of producing an approximate result within a reduced time frame. The three methods are listed in table 7.10.

Table 7.10 - Table discussing the influence on accuracy and runtime of the coarse prediction methodologies.

Coarse prediction method	Effect on runtime	Effect on accuracy
Reduce granularity of historical energy profile using custom reduction algorithm.	Reduces the data size and as a result, the compilation time for machine and production line energy profiles.	By using reduced profiles, the final result will not be as accurate compared to using full length profiles.
Reduce Genetic Algorithm population size	Reduces the number of candidate schedule chromosomes to consider.	By considering a smaller population, the algorithm may return a shallow local optimum.
Combination of above methodologies	A combination of the above two effects.	A combination of the above two effects.

The implementation of these methodologies using the current optimisation system was simple and achieved with little overhead. Where necessary, the energy profile prediction engines were instructed to source their historical data from a new library, generated by applying the profile reduction algorithm introduced in section 6.3.1, to a copy of the existing profile library. As for reducing the population size, this was simply a matter of adjusting a single program constant. Initial experimentation focused on how using reduced length profiles would influence the runtime. For this, the library of original profiles was reduced by a factor of ten. Table 7.11 shows the fitness function evaluation time for the series of test schedules using these reduced profiles.

Table 7.11 - Differences in fitness function evaluation runtime between valid and invalid candidate schedules using reduced energy profiles. (Results averaged over 100 iterations; Ran on an Intel i7 8GB Windows 8.1 computer).

Test schedule	Valid schedule evaluation time (µs)	Invalid schedule evaluation time (µs)	Valid evaluation time reduction compared to table 7.5 (Factor of)
#1	121.50	1.31	45.27
#2	242.72	1.72	48.31
#3	337.28	1.76	45.86
#4	434.48	1.69	45.29
#5	509.28	1.65	41.06
#6	644.31	2.40	40.79

In comparison with table 7.5, the results in table 7.11 demonstrate the significant influence the lengths of the historical profiles have on the valid schedule evaluation time. As expected, the invalid schedule evaluation time is immune to this influence as the historical profiles are not considered for invalid schedules. Curiously, the reduction in valid schedule evaluation time is four times the reduction ratio of the historical profiles. This demonstrates that reducing the profile lengths has more than just a singular effect on the evaluation time. One potential explanation for this is that using the reduced length profiles also reduces the memory resources required to hold each of the machine and production line profiles. Taking test schedule #6, the schedule with the longest time span as an example, the optimisation system originally required 768,000 bytes of system memory per profile. By using profiles reduced by a factor of ten, the memory requirements will also subsequently drop by a factor of ten, and fall under the 85,000 byte limit of the generation zero garbage collector (Windows Dev Center, No date). As discussed in section 7.3, this can result in a decreased runtime.

Table 7.12 - Table showing empirical reduction factors in optimisation system runtime measured for the three coarse prediction methods on all test schedules with most probable result (Results averaged over ten consecutive runs).

Test schedule	Coarse prediction method		
	Reduced waveform (Factor of)	Reduced population (Factor of)	Both methods (Factor of)
#1	15.62	11.15	166.84
#2	22.24	12.02	249.72
#3	29.08	10.30	297.88
#4	32.98	10.09	286.94
#5	17.72	7.45	110.54
#6	12.85	4.32	57.19
Average:	20.31	9.31	183.61

For a full evaluation, each of the three coarse methods was applied to the six test schedules. The results and runtimes can be seen in table 7.12 and figures 7.18 to 7.23, followed by discussions of the individual methodology results.

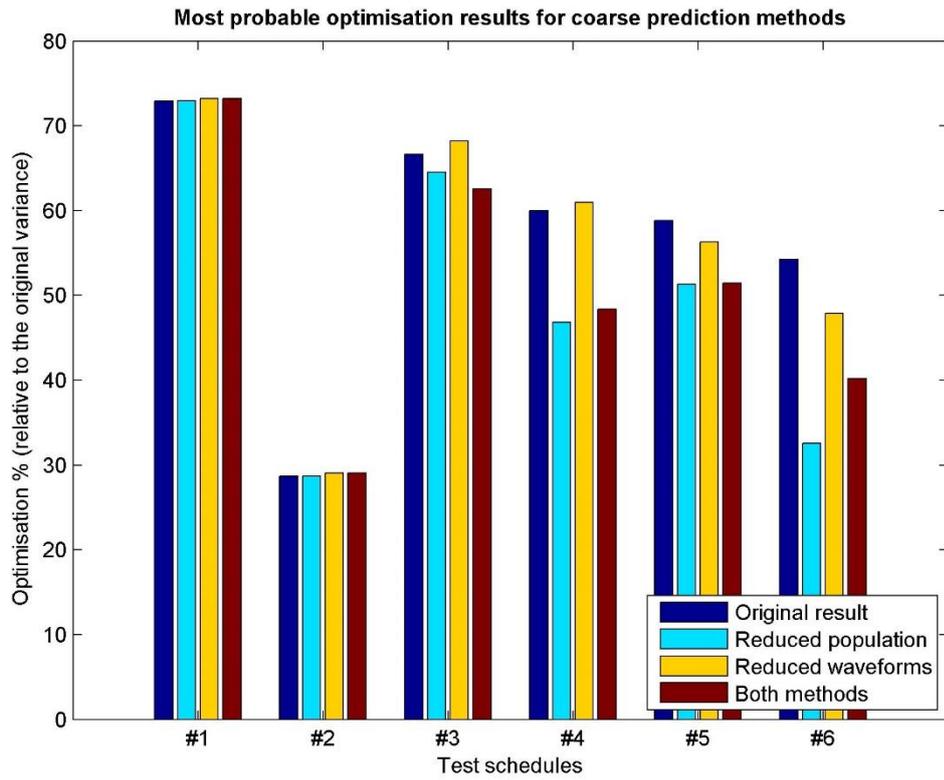


Figure 7.18 - Graph comparing most probable results from original and coarse prediction methods.

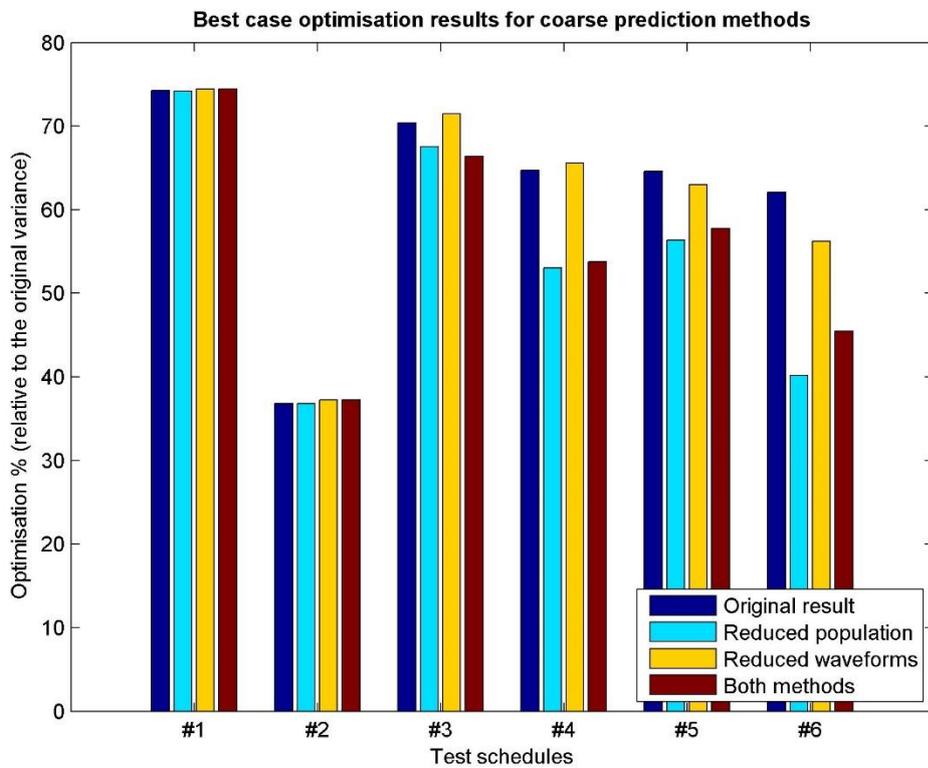


Figure 7.19 - Graph comparing best case results from original and coarse prediction methods.

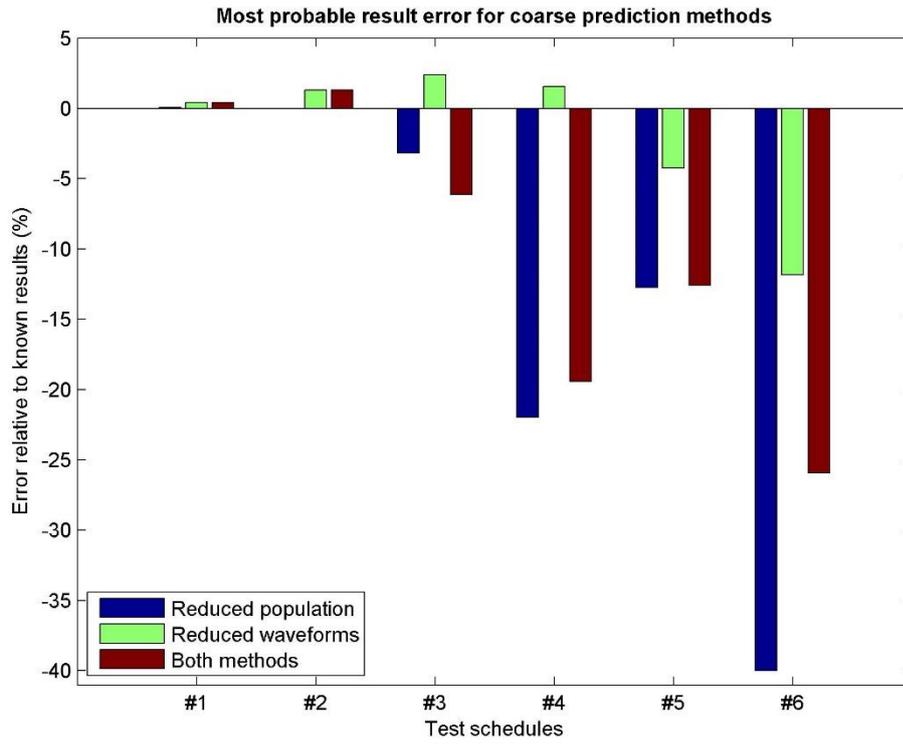


Figure 7.20 - Graph showing most probable coarse prediction methods error % compared to original results.

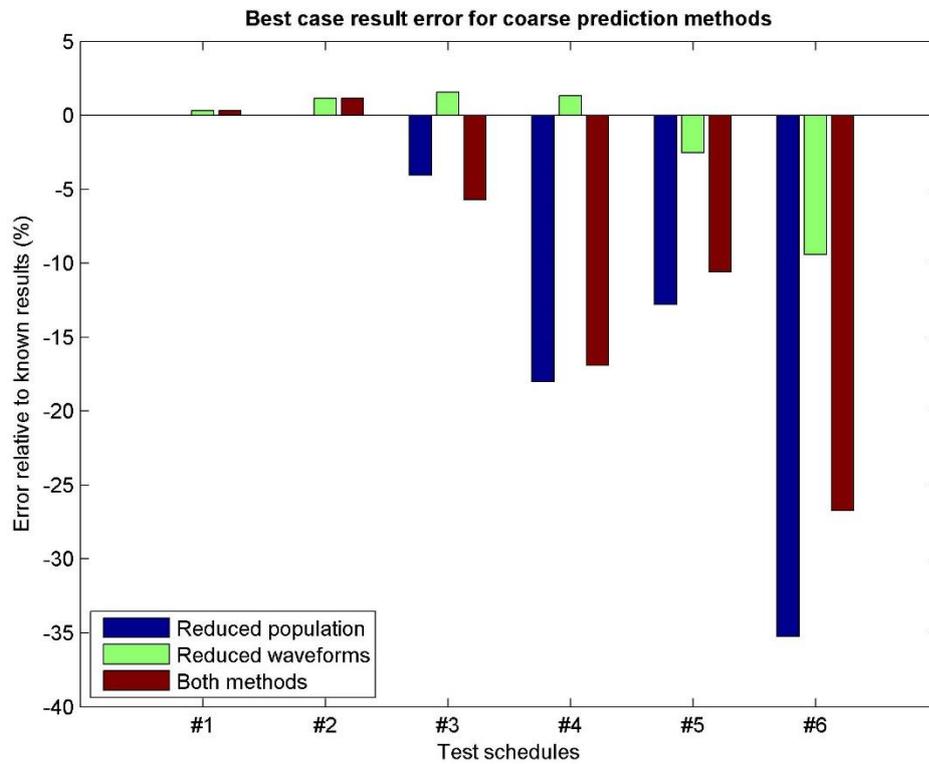


Figure 7.21 - Graph showing best case coarse prediction methods error % compared to original results.

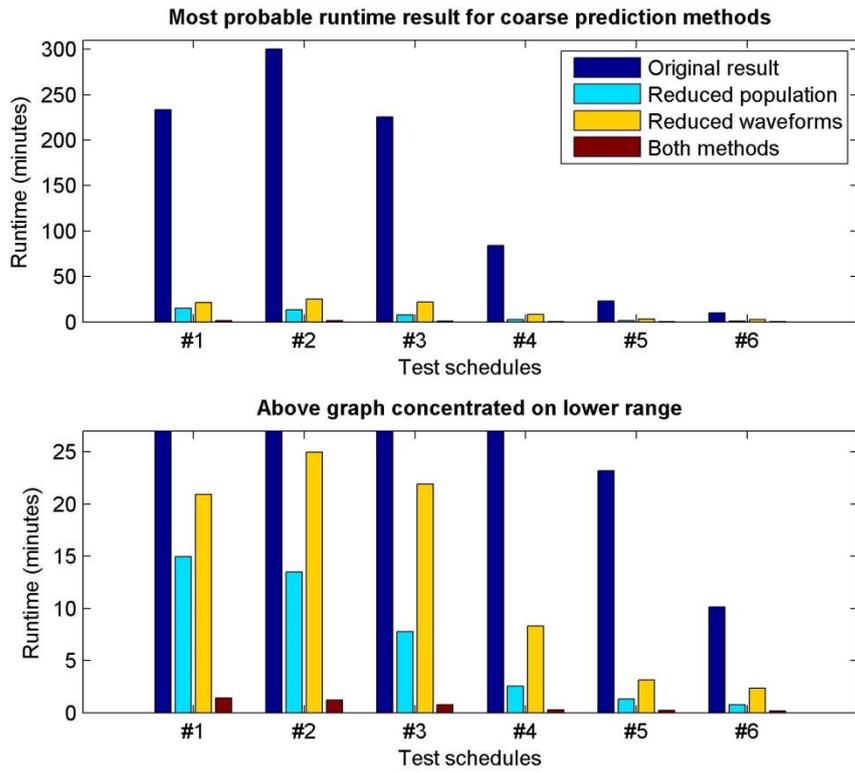


Figure 7.22 - Graph comparing most probable runtimes for original and coarse prediction methods.

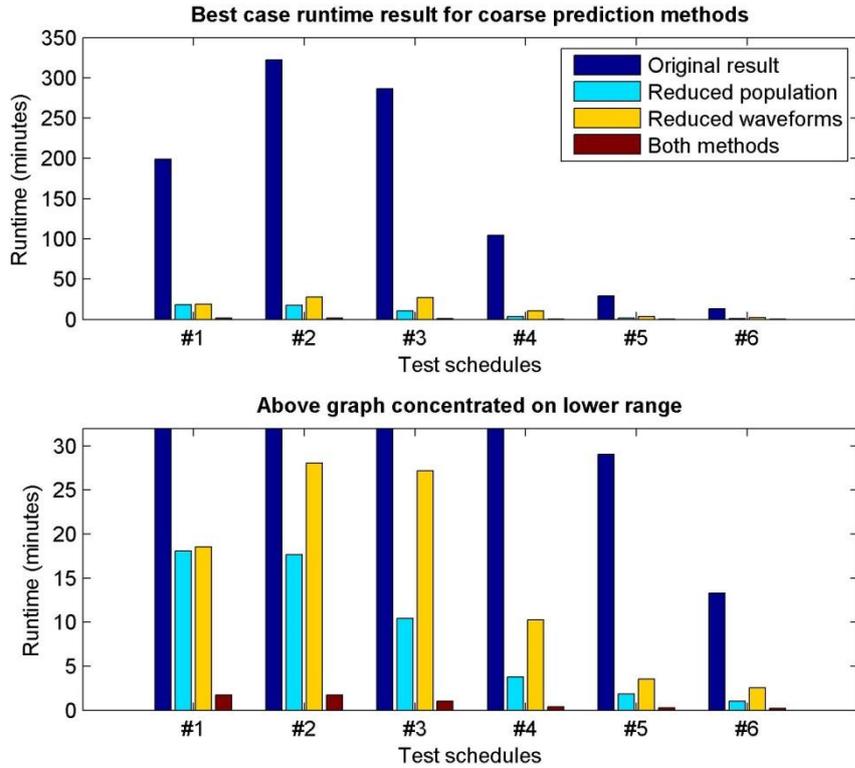


Figure 7.23 - Graph comparing best case runtimes for original and coarse prediction methods.

Reduced waveform methodology

In general, this methodology is seen to produce the most accurate predictions with a maximum recorded error of approximately -10%. Due to the information loss during the waveform reduction process, small positive errors can occur in the prediction. The error percentage will be directly related to the shape of the historical energy profiles and their information densities. However by utilising waveforms reduced by a factor of ten, this methodology was successful in reducing the optimisation systems runtime by a factor of 20.31 averaged over all test schedules.

These results indicate that using highly granular energy profiles is not beneficial, as they significantly increase the runtime while providing only marginal increases to result accuracy. However it is important to note that the reduced length profiles used here are a compressed form of the highly granular profiles and are produced via the custom compression algorithm (see section 6.3.1). As this attempts to minimise the loss of short lived waveform features (relative to the reporting rate), there is the potential that the reduced length profiles will be a more accuracy representation of the energy consumption than a profile directly recorded at a similarly reduced reporting rate.

Reduced population methodology

Contrary to the reduced waveform methodology, reducing the population size by a factor of ten produced the largest error range, with a maximum error of approximately -40%. Through test schedules #3 to #6, the negative error demonstrates that the optimisation system in its coarse prediction configuration returned a result less optimal than the original full accuracy version, as reducing the population restricts the Genetic Algorithms search time. The only case where this is not true is with test schedules #1 and #2, where the error is minimal. This is likely due to the fact that both of these schedules are the smallest out of all the test schedules, both in terms of the number of constraints and overall schedule timespan. Therefore it is plausible to conclude that the optimisation system even with a reduced population, is able to successfully locate a suitably optimal result within the allowed iterations. While the same cannot be said for the other test schedules, the accuracy loss is rewarded by a decrease in runtime. By reducing the

population size by a factor of ten, the runtime decreases by a factor of 9.31, averaged over all test schedules. Table 7.12 shows that for this method, the observed reduction is close to or less than the specified reduction factor, indicating that this method is already losing effectiveness. As this is mainly observed in the larger schedules, it is believed that this is due to the fact that, as figure 7.12 shows, when optimising these schedules, the Genetic Algorithm's population mainly consists of invalid candidates. Therefore reducing the population only reduces the number of invalid candidates which are already quickly evaluated and dismissed in the fitness function.

These results also show that for schedules where the Genetic Algorithm's population typically contains a large percentage of invalid candidates, reducing the population size does not benefit the returned solution. While in theory this should allow for valid candidates to have more influence on the search direction, in practice, the results in figures 7.20 and 7.21 demonstrate that the opposite effect occurs. This presents the case that while of no use to the final result, invalid candidate schedules do heavily influence the returned result.

Combination of both methodologies

As was to be expected, combining the two methodologies produces an amalgamation of the aforementioned advantages and disadvantages. In terms of the error, in all schedules but test schedule #3, the combined method is flanked on either side by the low erroneous reduced waveform methodology, and the high erroneous reduced population method. This indicates that the errors produced by both methods are negated to a certain degree. In terms of cost verses benefit, this method is highly appealing. As it utilises two independent methods, both of which are configured to theoretically reduce optimisation system runtime by a factor of ten, this method is theoretically capable of reducing the runtime by a factor of 100. However as seen in figures 7.22 and 7.23, and table 7.12, the overall reduction in runtime, averaged over the range of test schedules, is seen to a factor of 183.61.

Overall the results show that using a combination of both methods produces the best cost to benefit ratio. The runtime can be significantly decreased, far in comparison to the other

methods, while distributing the resulting negative effects to ensure the accuracy of the system remains within an acceptable range. Furthermore, the error of the coarse prediction does not seem to directly relate to the schedule's complexity, with predictions for test schedule #5 being far less erroneous than the smaller test schedule #4. However from a broad perspective it can be seen in that more complex schedules do have a higher potential error. As is shown in table 7.4, a schedule with more jobs is less likely to return a consistent value over consecutive runs. This is exacerbated by using a reduced population as it impedes the Genetic Algorithm's search performance. Furthermore as a larger schedule will involve a larger number of energy profiles, there will be a higher error due to the information loss in profiles.

7.8 Summary

In this chapter, the extensive optimisation system introduced in the previous chapters is tested and evaluated to ascertain its effectiveness at solving the applied problem. For all test schedules, the system is successfully able to reduce the energy consumption variance by a high degree, and where prudent, utilise features such as DMR to further the level of optimisation. The systems performance is dependent upon the schedule under consideration and there are cases where the schedules cannot be optimised by a significant degree. Furthermore, the systems performance does degrade slightly as schedule complexity increase with the system returning inconsistent results over repetitive runs indicating that the result is a local minima. However this issue needs to be considered in light of the problem's complexity where it can be deduced that the optimisation system performance scales acceptably. To address the lengthy runtime of the system, three coarse prediction methodologies are evaluated. While errors are present in the returned result, two of the three methods demonstrate better than expected reductions in runtime, therefore offering an appealing cost to benefit ratio.

CHAPTER 8

OPTIMISATION SYSTEM EXPANDABILITY

The optimisation system introduced in chapter 5 can be generalised as a generic optimisation system for production schedule consumables. While in this particular case, the consumable is electrical energy and the optimisation objective is energy consumption variance, this can easily be modified. In both the most probable and best case fitness functions, the calculation of energy consumption variance can be considered a very small part which occurs at the very end of the procedure and can be considered completely independent from the rest of the function. As a result, while they are by no means novel goals, the optimisation system can be adjusted and used for alternate purposes with only minor changes.

8.1 Optimisation of Minimum Peak Energy Consumption

As an initial demonstration of the alternative uses, the optimisation was modified to solely reduce peak energy consumption. In this case, the fitness value for a candidate schedule was assigned as the associated energy profile's peak energy consumption, instead of its variance. This was done by replacing equation (5.7) with (8.1) and assigning the result as the candidate schedules fitness. Table 8.1 shows the level of reduction possible and compares it with the peak consumption observed when a schedule is optimised for energy consumption variance.

$$\max \{ profile_{Predicted}^{Prod}(t) \} \quad (8.1)$$

Table 8.1 - Table demonstrating the level of peak energy consumption reduction compared to the peak consumption when a schedule is optimised for energy consumption variance.

Test schedule	Original peak energy consumption	Reduced peak energy consumption results (A RMS)			
		Schedule optimised for peak energy consumption		Schedule optimised for energy consumption variance	
		Most probable	Best case	Most probable	Best case
#1	297.17	186.93	171.83	186.93	171.83
#2	312.13	191.33	181.43	192.43	194.53
#3	374.33	215.51	214.18	233.43	233.43
#4	376.36	240.03	229.85	260.83	243.33
#5	439.56	301.19	298.78	346.56	321.36
#6	456.76	325.75	310.89	346.56	321.58

The results in table 8.1 demonstrate that the modified system is able to successfully reduce the peak consumption. It is also seen that while optimising for minimal energy consumption variance does reduce the peak consumption, some higher peaks can remain. In the case of test schedule #1, figure 7.5 shows that this schedule can be optimised such that no two jobs are in concurrent execution. Therefore the minimal variance and peak consumption is observed. For the most complex test schedules, large differences are observed between the two result sets.

Depending on the manufacturer's preferences, these results show that the optimisation system can be easily employed to either minimise energy consumption variance or peak consumption.

8.2 Demand Response for Renewable Energy Generation

Along with the minimisation of peak consumption, the optimisation system can also be used to adapt the energy demand in response to the supply from renewable energy resources, in a technique similar to Demand Side Management (DSM) (Samadi et al, 2011). Where production lines are partly or entirely powered from renewable resources, the optimisation system could modify a schedule in an attempt to fit the production lines energy consumption within the predicted supply response. In this case, the optimisation system is tasked with minimising (8.2), where $profile_{forecast}$ is the energy supply forecast from the renewable resource.

$$\sum_{t=s_e}^{D_{Max}} profile_{Predicted}^{Prod}(t) - profile_{Forecast}(t) \quad (8.2)$$

Examples of this modification are shown in figures 8.1 and 8.2, where test schedule #6 has been optimised for two different forecast profiles. Numerical results are shown in table 8.2.

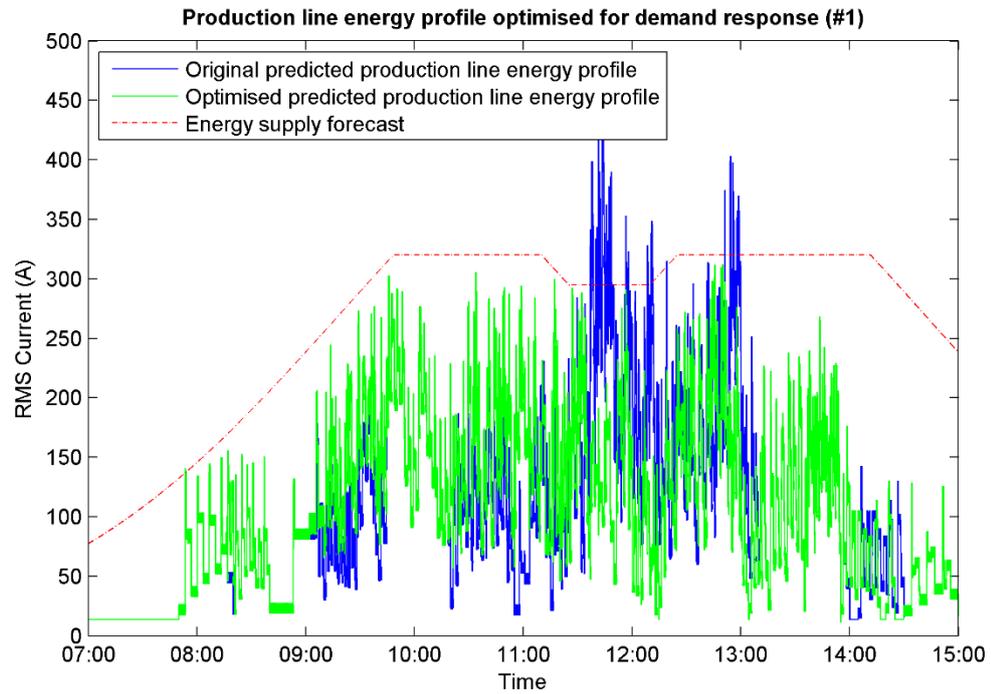


Figure 8.1 – Example of production line energy profile which is optimised for demand response with renewable energy sources.

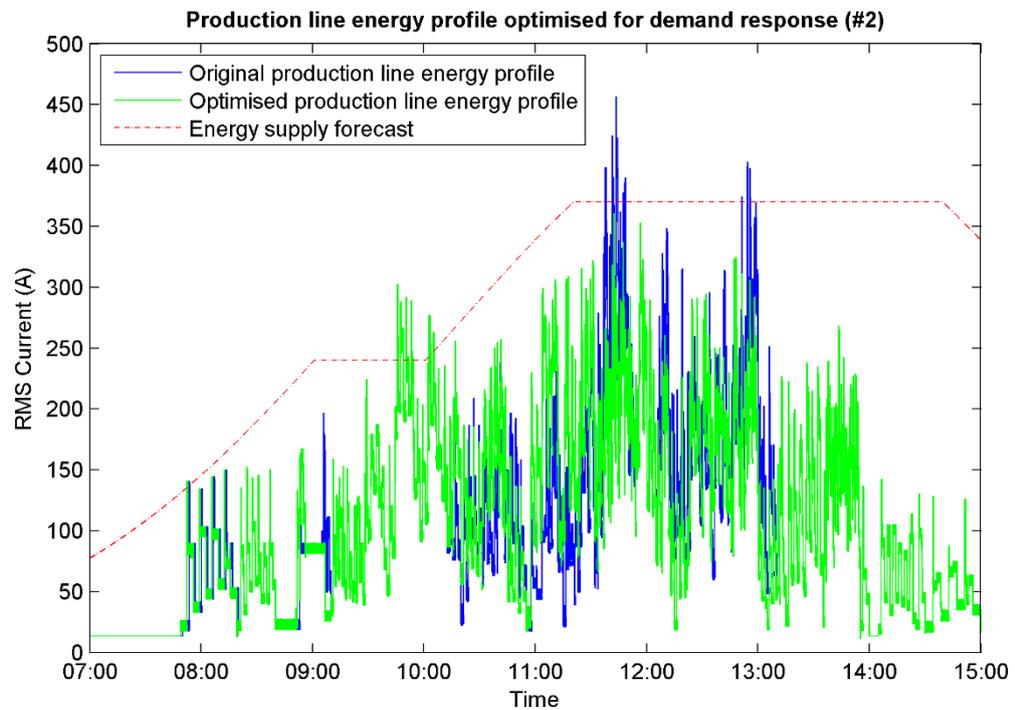


Figure 8.2 - Second example of production line energy profile which is optimised for demand response with renewable energy sources.

Table 8.2 - Results of reduction to forecast-demand breach experiment.

	Original energy breaching forecast (% of energy profile)	Optimised energy breaching forecast (% of energy profile)	Reduction (%)
First example (Figure 8.1)	0.65	0.0005	99.92
Second example (Figure 8.2)	0.25	0.17	34.36

Due to the need to maintain scheduling constraints, the modified optimisation system does not guarantee that the production line energy consumption will completely fit within the supply response. However, the system has been proven useful for this application. This application also offers dynamic termination conditions, as the optimisation system can stop once the value calculated in (8.2) becomes non-positive. While this may not produce the most optimal solution, this is not the goal in this particular application.

Further, minor modifications allow the system to account for energy storage systems. In (8.2), for every value of t , if the supply $profile_{Forecast}(t)$, outweighs the demand $profile_{predicted}^{prod}(t)$, excess energy $profile_{excess}(t)$ can be diverted into an energy storage system whose current storage level is represented by $B_{lvl}(t)$. When this is at capacity B_{Cap} , any remaining excess can be exported to the National Grid for profit. When demand outweighs supply, stored energy can be tapped, $profile_{Required}(t)$, in an attempt to meet demand levels. In the case where stored energy is insufficient, or the energy store is depleted, energy can be sourced from the National Grid at cost, $profile_{Import}(t)$. See (8.3). The fitness function should therefore minimise the total energy sourced from the National Grid as seen in (8.4).

$$\begin{aligned}
& profile_{Excess}(t) = profile_{Forecast}(t) - profile_{predicted}^{prod}(t) \\
& \text{IF } profile_{Excess}(t) > 0 \\
& \quad \text{IF } (B_{Cap} - B_{lvl}(t-1)) \geq profile_{Excess}(t) \\
& \quad \quad B_{lvl}(t) = B_{lvl}(t-1) + profile_{Excess}(t) \\
& \quad \text{ELSE} \\
& \quad \quad profile_{Export}(t) = profile_{Excess}(t) - (B_{Cap} - B_{lvl}(t-1)) \\
& \quad \quad B_{lvl}(t) = profile_{Excess}(t) - profile_{Export}(t)
\end{aligned} \tag{8.3}$$

$$\begin{aligned}
& \text{ELSE IF } profile_{Excess}(t) < 0 \\
& \quad profile_{required}(t) = profile_{Excess}(t) \\
& \quad \text{IF } profile_{required}(t) \leq B_{lvl}(t-1) \\
& \quad \quad B_{lvl}(t) = B_{lvl}(t-1) - profile_{required}(t) \\
& \quad \text{ELSE} \\
& \quad \quad profile_{import}(t) = profile_{required}(t) - B_{lvl}(t-1) \\
& \quad \quad \min \{ profile_{import}(t) \}
\end{aligned} \tag{8.4}$$

As an example, figure 8.3 shows a fictitious predicted energy consumption profile for a production line powered by a photovoltaic (PV) array, complimented with a 20Ah battery¹. Here, the production line requires power from the National Grid at around 12:00 and 13:00. In figure 8.4, the schedule is optimised allowing the production line to be powered entirely from the PV array and the battery system.

¹ The batteries in this example are assumed to have a discharge cut-off of 50% of the total battery capacity, along with an 80% charging efficiency. To simplify the representation, the 100% full to 50% discharge cut-off is represented as 100% to 0%.

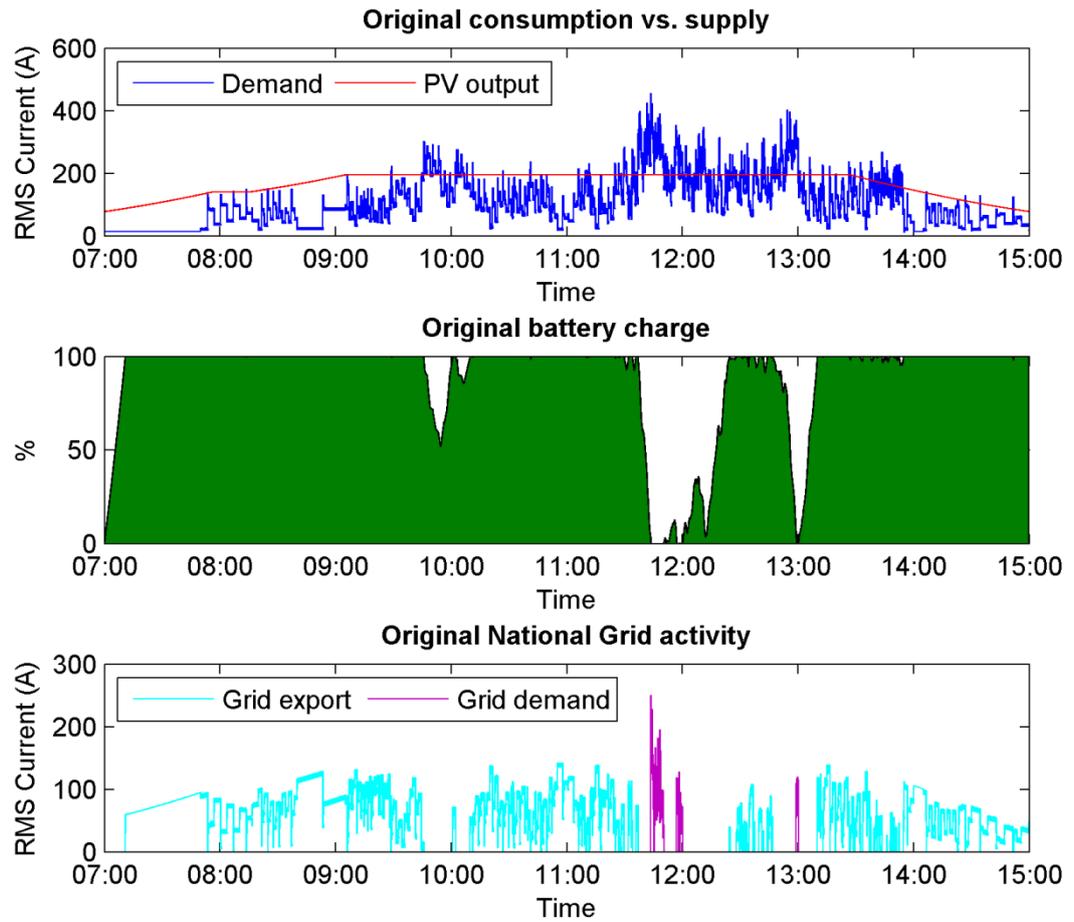


Figure 8.3 - Example prediction of production line powered from a PV array with battery storage. A backup connection to the National Grid is available for exporting excess power and demanding power when demand exceeds local availability. Note: Battery percentage shown is measured between the fully charged and the acceptable discharge cut-off level.

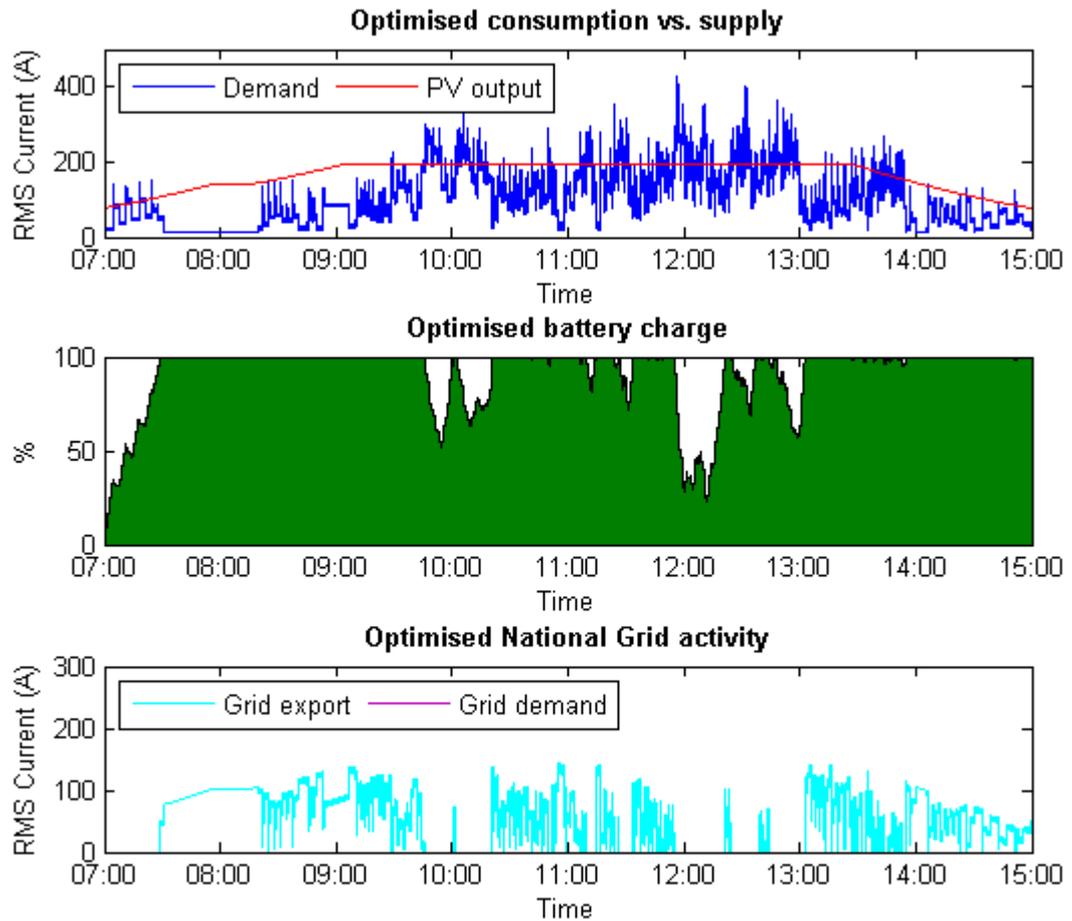


Figure 8.4 - Optimised version of data shown in figure 8.3.

Again, between figures 8.3 and 8.4 the system is shown to be capable of both solving this category of problem, while accounting for additional information such as the level of electricity exported to the National Grid. This means that the system could also be used in smart grid applications or from a financial viewpoint, furthered if the varying cost of energy were considered also.

8.3 Other Potential Expansions

Outside the realms of optimisation, the system also has additional uses. A manufacturer can use the system to predict the energy consumption profile of their manufacturing line for any schedule prior to it being executed, so long as the historical energy profiles are available. While this will not account for the total facility's energy consumption, it could be used for predictive

billing purposes. In the long term, being able to predict the future energy consumption to a high granularity could aid in contract negotiations with the energy supplier. This is similar to the 24 hour forecasting by Pechman et al (2011; 2012).

While electrical current is considered in this implementation, other forms of energy could also be used, either solely or in combination, through modifications to the fitness functions and prediction engines.

Overall, the system developed can be seen as a highly adaptable piece of software that can be applied to both commonly used and novel optimisation situations both within and outside the energy domain.

CHAPTER 9

CONCLUSION

This thesis presents a novel production schedule optimisation system for the minimisation of production line energy consumption variance. This work commenced with a substantial investigation into current and state-of-the-art methodologies for the consideration of process energy consumption in manufacturing environments. Research findings identified that there is a discontinuity between currently employed methods and those proposed by researchers. Furthermore, the consideration of energy consumption is seldom with many approaches viewing it purely from an energy price viewpoint with a highly abstract representation of the energy consumption itself thanks to the current state of energy monitoring devices. To that end, this work proposed and successfully implemented several novel strategies both specifically for considering energy consumption variance and for energy in manufacturing in general. These are:

- The development of a Genetic Algorithm based production schedule optimisation system, which is proven to minimise the energy consumption variance generated through schedule execution.
- The development of the Intelligent Historical Library for Manufacturing Energy Prediction which is able to produce predictions which are independent of influence from the machines mechanical conditions.
- Predictions generated from time-series energy consumption profiles with a high (sub-second) temporal granularity.
- The ability to generate synthetic energy profiles for a particular job when existing data is insufficient.
- A schedule optimisation system which is proven to be readily expandable into other energy-based optimisation applications.

In this chapter the top-level objectives and findings are condensed and evaluated in relation with the original research aim and objectives presented in chapter 1. Finally, recommendations based on retrospective analysis are given before concluding this work with a short discussion as to the applicability of the work in an actual manufacturing production line.

9.1 Addressing the Research Aim and Objectives

Achieving the aim of research as seen in chapter 1, required substantial background investigations which can be segregated into three distinct but related topics – industry energy monitoring, modelling, and energy optimisation; as dictated by the initial three accompanying research objectives. Progress in each topic is constructed atop of the previous topic. The conclusion from the findings of these three aims will follow a similar structure, beginning with the findings from investigations into industrial energy monitoring and ending with a foundation upon which the novel system is developed to address the research aim.

“One of the key prerequisites for any energy-based optimisation in the manufacturing sector is to understand how energy is distributed and consumed in production lines”

The above phrase introduces the chapter on energy monitoring principles (section 2.2) and denotes the critical link between this work and energy monitoring. Put simply, you cannot optimise what you cannot see. Current industry-grade energy monitoring equipment while apt at recording multivariate consumption levels to a high degree of accuracy, lack the reporting rates needed to provide data at a high temporal granularity. While providing this high level of granularity will call for an increase in computational processing power, this is not considered to be the limiting factor given the abundance of low cost high power embedded microcontrollers. This is further reinforced by the development of a custom energy monitoring system which is able to record seven current channels plus voltage at a reporting rate of 150ms. While the measurement specification of the custom energy monitoring system was much simpler than that of a commercially available system, this does not allow for the significantly higher reporting rate as the cost-to-benefit ratios are disproportionate. Necessitating the calculation of supply

frequency, power factor and the related power measures cannot delay the reporting of data by several minutes. This is especially given the level of modern electronics and the UK supply frequency of 50Hz. All this serves as the first, and perhaps the only needed indicator that energy related objectives are not highly considered in industry.

Similar findings are seen throughout this work. Chapter 3 shows that a significant amount of research has been conducted into predicting and modelling energy consumption with respect to manufacturing environments. However the overall diversity of methodologies is limited, although this could be a consequence of the problems approachability. Despite providing high levels of accuracy with some works attaining greater than 95%, the applicability of these methods is brought into question. Most methods are developed around a single type of machine, and in some cases, the values of required attributes are difficult or impractical to know in working production lines. Furthermore as manufacturing production lines are highly dynamic environments, while dependent upon the attributes required, there is the possibility that their values may change over time. This phenomenon is not addressed in all but a few works, further bringing into question the long term accuracy of the proposed methods. Manufacturing machines are by no means fixed, tools blunt, machines get older, are maintained, and parts are changed or upgraded. Simply put, it is believed that no fixed modelling methodology can accurately account for this dynamic environment over time. The granularity of the calculated models is also poor with highly abstract data values such as average and maximum consumption used. This may potentially be a consequence of the poor reporting rates of energy monitoring systems. Manufacturing jobs can consist of many elemental operations each with its own unique energy consumption profile. By restricting the representation to a single value, the overall loss of information is extreme.

All this directly influences current efforts into energy optimisation. While there are numerous examples of works to optimise energy consumptions for both individual machine and entire production lines, these again only reference abstract values which as a consequence, limit approachability. The two umbrella methodology seen are machine-centric, which aim to reduce the overall energy consumption of individual machines, and schedule-centric which aims to

operate on the entire production line in accordance with a particular energy-based objective while considering traditional constraints. For this approach, a popular energy-based objective is the minimisation of peak energy consumption. Abstract values such as peak consumption for individual jobs may seem beneficial for approaches like this. Data requirements are reduced, requiring only a single value for each job which will result in decreased processing time compared to time-series models. However, results from using the custom high granularity energy monitor show that peak consumption values are a poor representation of the energy consumption for a typical manufacturing job. As these are seen to contain large peaks relative to the rest of the consumption levels, optimising an entire production line knowing only a single energy value will likely result in less-than optimal solutions. Despite all this, the amount of research into energy optimisation in manufacturing lines demonstrates there is an avid interest in the subject.

Concluding the research aim required all of the foreseen shortcomings to be addressed and improved upon in a manner representative of the topics relationship with each other. Beginning at the data level, a custom energy monitoring system with a temporal granularity of 150ms is developed to provide data at the necessary level of information. While the measured values are limited to RMS voltage and current, this is due to the inability to regulate the power factor of the supply and not related to the temporal granularity.

The availability of high granularity energy consumption data subsequently permits high accuracy consumption modelling. However there does not seem to be any single methodology which can a) produce high granularity models, b) be universally applicable to all manufacturing machines, and c) be easily and quickly applied to modern, working production lines. In this work, the novel Intelligent Historical Library is proposed to overcome these limitations by directly utilising recorded historical data. By recording and storing multiple profiles for identical jobs along with associated machine-related metadata, predictions can compensate for machine-related changes to the jobs energy consumption. Furthering the originality, when historical data is lacking for a particular job, the library is able to generate temporary synthetic

profiles by analysing how the metadata influences the energy consumption. These synthetic profiles are then generated based on the probability of them being requested for a prediction.

All this sets the foundations for the main original contribution – the production schedule optimisation system, which directly addresses the research aim. Literature in the area shows that minimising energy consumption variance has not seen a significant research focus. Methods such as demand side management attempt to reduce peak demand but focus solely on the electrical distribution network and are unable to finely control the enormity of end users. The approach documented here looks at a more manageable problem of controlling the variance of a production facility. A Genetic Algorithm is used to search for and select a suitable solution to the problem while maintaining hard and soft constraints, along with additional components which interface between the Genetic Algorithm, the production schedule and the real-world. Energy consumption variance predictions are made using highly granular time-series energy profiles provided by the Intelligent Historical Library. With multiple profiles available for each job, two results can be searched for and returned – one based on the most likely energy consumption to be encountered, and another representing the results estimate lower bound, known as the best case result.

Many features are integrated into the schedule optimisation system to ensure a valid and highly optimal result is returned. These can include active features such as Dynamic Machine Reassignment (DMR), or passive features like the Genetic Algorithms periodic refresh of the population, and the optimisation of internal parameters. Validity of candidate schedules is confirmed through checking each job for compliance with process deadlines and start times, resource conflicts, and job overlap, and each check is sequential allowing for easy integration of additional checks.

The operation and performance of the schedule optimisation system is proven through extensive testing with six different production schedules. Each schedule was originally produced via a traditional production planning tool and the subsequent data was inserted into custom schedule templates which allow for the accountability of energy consumption. For all but one of the test schedules, the original energy consumption variance was decreased by a minimum of 50%,

however this does come as a cost of extending the total processing time, typically to the maximum allowed time. This can be controlled however by artificially adjusting the constraints via assigning process priorities. Where the constraints limit the amount of flexibility within the schedule, or the schedule is considered already mostly optimised by default, the optimisation system effectiveness is limited, as is shown in one of the test schedules where only a 29% decrease in variance is observed. Simulated results are further reinforced with an actual implementation. These help to demonstrate the accuracy and applicability of the optimisation system when applied to a real-world scenario, with an error in parts of the time-series prediction being sourced to factors outside the consideration of the system.

The entire system is evaluated both as a whole and separately where necessary, to determine the level of influence each feature has on the overall result. It is seen that the performance is dependent upon the schedule under consideration, with features such as DMR only providing a useful influence for the largest schedule. Furthermore it is seen in experimentation that the optimal value of internal parameters also changes depending on the schedule. As a result, parameter values were chosen based on universal schedule application over individual optimality.

Finally, when the runtime of the system becomes an issue, three methodologies are presented to minimise runtime and produce a coarse prediction. Results show that loss in accuracy is maintained within acceptable levels while producing a more than specified reduction in runtime.

9.2 Future Work Recommendations

The system developed, while proven working to a high standard does have a small number of limitations. First and foremost, experimentation has shown that the optimal values of some of the Genetic Algorithm's internal parameters are schedule dependent. In the current configuration, these values are fixed. However if the relationship between the parameter values and the schedules could be mapped, the schedule optimisation system could optimise its search performance based on the schedule it is tasked with optimising. The main disadvantage of this is

that it would require significant and time consuming experimentation to investigate the possible range of parameter values at a suitable resolution.

Similarly, the termination conditions could be altered based on the schedule. Experiment results show that for simpler schedules, the optimal result is found in earlier iterations of the Genetic Algorithms restart. Again, understanding this relationship should allow for dynamic termination conditions which would directly affect the runtime of the system. However this does run the risk of the system terminating prematurely.

All these can be considered as non-significant limitations and were not implemented in this work due to the large time commitment required for the experimentation.

One further limitation is that the prediction does not account for energy consumption which cannot be scheduled. This includes machinery such as air compressors, which activate automatically based on air pressure. While compressed air consumption can be measured for a known job, in reality, compressed air systems will likely contain leaks resulting in the compressors activation time being unknown. Similarly, the energy consumed for none manufacturing tasks such as lighting, HVAC and general office equipment cannot be scheduled. With that said, it is believed that in typical production lines, the consumption rate of machines will far outweigh the energy consumed by none manufacturing equipment. While not considered at current, it may be possible for the optimisation system to account for it in the future via a fixed assumed consumption level or a more complex model.

One feature which could be implemented into the schedule optimisation system would be the intelligent machine shutdown ability, as seen in section 4.1.1. This would prevent machines from idling unnecessarily and contributing to the production lines energy consumption.

9.3 Applications in Real World Manufacturing Production Lines

The applicability of the entire developed system in a real world production line will likely be directly dependent upon the nature of individual manufacturing lines. An actual implementation

of the schedule optimisation system proves that the prediction is accurate, although purely considering current consumption can lead to errors due to power factor. However this is a limitation of the modelling data and not of the optimisation system. While these results along with the ones from simulations show that the system possesses a large potential it does make a number of assumptions about the control features of the production line:

1. Machines remain idling when not in use.
2. All jobs are executed on CNC machinery ensuring identical executions for repeated jobs.
3. All jobs are to commence at the time specified by the schedule.
4. All jobs to be executed are scheduled by the optimisation system.
5. All machines contain some form of networked energy monitoring device.

Most of the assumptions are based around exact compliance with the schedule. In the case where human intervention is a necessary part of the jobs execution, some degree of error should be anticipated in the prediction accuracy. As manufacturing lines can unfortunately encounter unexpected delays and other issues, the schedule will have to be modified and re-optimised accordingly.

After this point, individual applicability primarily comes down to the manufacturers particular requirements. Do additional constraints need to be accounted for? Can the manufacturer afford to allow the total makespan to be expanded? The entire system has been designed to be application generic, meaning it can be applied to any type of production environment. As stated in the previous section, and in chapter 8, the developed system can be easily expanded to include additional constraints, and can also be applied to a number of applications, not solely within the energy domain.

9.4 Final Word

All of the discussed features and associated results demonstrate the proficiency of the developed system for solving the research aim. Ultimately, this work presents a novel contribution and an alternative schedule strategy to manufacturers who wish to increase the efficiency of their utilisation of the local power distribution network. Original contributions are made both in energy modelling and manufacturing energy optimisation, with focus on the latter. Extensive experimentation demonstrates that this approach can offer significant benefits to manufacturers.

List of References

- Ailva, A. and Falcão. (2008). Fundamentals of Genetic Algorithms. In: K. Lee and M. El-Sharkawi, eds., *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*, 1st ed. New York City: Wiley-IEEE Press, pp. 25 – 42.
- Al-Naymat, G. and Taheri, J. (2008). Effects of Dimensionality Reduction Techniques on Time Series Similarity Measurements. In: *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*. New York City: IEEE, pp. 188 – 195.
- Artigues, C., Lopez, P. and Haït, A. (2009). Scheduling Under Energy Constraints. In: *International Conference on Industrial Engineering and Systems Management*. [online] New York: IEEE. Available at: <https://hal.archives-ouvertes.fr/hal-00367614/document> [cited 17th December 2015].
- Artigues, C., Lopez, P. and Haït, A. (2013). The Energy Scheduling Problem: Industrial Case-Study and Constraint Propagation Techniques. *International Journal of Production Economics*, **143** (1), pp. 13 – 23.
- Balogun, V. and Mativenga, P. (2013). Modelling of Direct Energy Requirements in Mechanical Machining Processes. *Journal of Cleaner Production*, **41**, pp. 179 – 186.
- Berthold, T., Heinz, S. and Schulz, J. (2011). Approximative Criterion for the Potential of Energetic Reasoning. In: a. Marchetti-Spaccamela and M. Segal, eds., *Theory and Practice of Algorithms in (Computer) Systems: First International ICST Conference, TAPAS 2011, Rome, Italy, April 18-20, 2011. Proceedings*, 1st ed. Berlin: Springer-Verlag, pp. 229 – 239.
- Bierwirth, C. and Mattfeld, D. (1999). Production Scheduling and Rescheduling with Genetic Algorithms. *Evolutionary Computation*, **7** (1), pp. 1 – 17.
- Blickle, T. and Thiele, L. (1996). A Comparison of Selection Schemes used in Genetic Algorithms. *International Journal of Evolutionary Computation*, **4** (4).

- Boger, Z. and Guterman, H. (1997). Knowledge Extraction from Artificial Neural Network Models. In: *Systems, Man and Cybernetics. Computational Cybernetics and Simulation. 1997 IEEE International Conference on*. New York: IEEE, pp, 3030 – 3035 (vol. 4).
- Brown, G. (2010). Ensemble Learning. In: C. Sammut and G. Webb, eds. *Encyclopaedia of Machine Learning*, 1st ed. New York City: Springer US, pp. 312 – 320.
- Brown, N., Greenough, R., Vikorev, K. and Khattak, S. (2012). Precursors to using Energy Data as a Manufacturing Process Variable. In: *Digital Ecosystems Technologies (DEST), 6th International Conference on*. New York City: IEEE, pp. 1 – 6.
- Brozzone, A., Anghinolfi, D., Paolucci, M. and Tonelli, F. (2012). Energy-Aware Scheduling for Improving Manufacturing Process Sustainability: A Mathematical Model for Flexible Flow Shops. *CIRP Annals – Manufacturing Technology*, **61** (1), pp. 459 – 462.
- Burtini, G., Fazackerly, S. and Lawrence, R. (2013). Time Series Compression for Adaptive Chart Generation. In: *Electrical and Computer Engineering (CCECE), 2013 26th Annual IEEE Canadian Conference on*. New York City: IEEE, pp. 1 – 6.
- Butters, T. (2014). *Time Series Reduction*. [online] Sabisu. Available from: <http://www.sabisu.co/resources/> [cited 17th December 2015].
- Chen, G., Zhang, L., Arinez, J. and Biller, S. (2012). Energy-Efficient Production Systems through Schedule-Based Operations. *Automation Science and Engineering, IEEE Transactions on*, **10** (1), pp. 27 – 37.
- Committee on Climate Change (No date). *The Climate Change Act and UK Regulations*. [Online] Committee on Climate Change. Available from: <https://www.theccc.org.uk/> [cited 14th December 2015].
- Corder, G. and Foreman, D. (2014). *Nonparametric Statistics: A Step-by-Step Approach*. 2nd ed. Hoboken, NJ: Wiley, pp. 1 – 3, 69 – 80.
- Cooper, J. and Hinde, C. (2003). Improving Genetic Algorithms' Efficiency using Intelligent Fitness Functions. In: P. Chung, C. Hinde and M. Ali, eds., *Developments in Applied*

Artificial Intelligence: 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert System, IEA/AIE 2003, Loughborough, UK, June 23-26, 2003, Proceedings, 1st ed. Berlin: Springer-Verlag, pp. 636 – 643.

Current Cost (No date). ENviR. [online] Current Cost. Available from: <http://www.currentcost.com/product-envir.html> [cited 15th December 2015].

Dang, T. and Ringland, K. (2012), Optimal Load Scheduling for Residential Renewable Energy Integration. In: *IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. New York City: IEEE, pp. 516 – 521.

Dietmair, A., Eberspaecher, P. and Verl, A. (2009). Predictive Simulation for Model Based Energy Consumption Optimisation in Manufacturing System and Machine Control. In: *Flexible Automation and Intelligent Manufacturing 2009, 19th International Conference on*. Red Hook, NY: Curran Associates Inc. pp. 226 – 233.

Dietmair, A. and Verl, A. (2009). Energy Consumption Forecasting and Optimisation for Tool Machines. *Modern Machinery (MM) Science Journal*, (March), pp. 62 – 67.

Dietterich, T. (2000). Ensemble Methods in Machine Learning. In: J. Kittler and F. Roli, ed., *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, 1st ed. New York: Springer-Verlag, pp. 1 – 15.

Duerden, C., Shark, LK., Hall, G. and Howe, J. (2014a), Genetic Algorithm based Modification of Production Schedule for Variance Minimisation of Energy Consumption, In: *Proceedings of the World Congress on Engineering and Computer Science 2014 Vol I*. Hong Kong: IAENG, pp. 370 – 375.

Duerden, C., Shark, L-K., Hall, G. and Howe, J. (2014b). Minimisation of Energy Consumption Variance for Multi-Process Manufacturing Lines Through Genetic Algorithm Manipulation of Production Schedule. *Engineering Letters*, **23** (1), pp. 40 – 48.

Duerden, C., Shark, L-K., Hall, G. and Howe, J. (2015). Genetic Algorithm for Energy Consumption Variance Minimisation in Manufacturing Production Lines Through

- Schedule Manipulation. In: H. Kim, M. Amouzegar, and S. Ao, eds., *Transactions on Engineering Technologies: World Congress on Engineering and Computer Science 2014*, 1st ed. Netherlands: Springer-Netherlands, 2015. pp. 1 – 13.
- Duerden, C., Shark, L-K., Hall, G. and Howe, J., Genetic Algorithm based Production Schedule Optimization System for Highly Granular Energy Consumption Variance Minimization. In print: *IEEE International Conference on Systems, Man and Cybernetics 2015*.
- Duerden, C., Shark, L-K., Hall, G. and Howe, J., Genetic Algorithm based Production Schedule Optimization System for Highly Granular Energy Consumption Variance Minimization. *Accepted for publication: 50th Annual Conference on Information Sciences and Systems (CISS)*.
- Duflou, J., Sutherland, J., Dornfeld, D., Herrmann, C., Jeswiet, J., Kara, S., Hauschild, and M. Kellens, K., 2012. Towards Energy and Resource Efficiency Manufacturing: A Processes and Systems Approach. *CIRP Annals – Manufacturing Technology*. **61** (2), pp. 587 – 609.
- Eberspächera, P. and Verla, A. (2013). Realizing Energy Reduction of Machine Tools through a Control-Integrated Consumption Graph-Based Optimization Method. *Procedia CIRP*, **7**, PP. 640 – 645.
- Emec, S., Kuschke, M., Chemnitz, M. and Strunz, K. (2013). Potential for Demand Side Management in Automotive Manufacturing. In: *Smart Grid Technologies Europe (ISGT EUROPE)*, 4th IEEE/PES Innovative. New York City: IEEE, pp. 1 – 5.
- Environmental Taxes Reliefs and Schemes for Businesses, (2015), *Climate Change Levy* [Online]. Available from: <https://www.gov.uk/green-taxes-and-reliefs/climate-change-levy> [cited 14th December 2015].
- E-PPS. (No date). Dortmund: Transfact.
- Fang, K., Uhan, N., Zhao, F. and Sutherland, J. (2013). Flow Shop Scheduling with Peak Power Consumption Constraints. *Annals of Operations Research*, **206** (1), pp. 115 – 145.

- Fang, K., Uhan, N., Zhao, F. and Sutherland, J. (2011). A New Shop Scheduling Approach in Support of Sustainable Manufacturing. in Hasselbach, J. Herrmann, C. eds, *Glocalized Solutions for Sustainability in Manufacturing*, 1st ed. Berlin: Springer-Verlag, 2010. pp. 305 – 310.
- Fausett, L. (1993). *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Cambridge: Pearson, pp. 3.
- Filho, M., Barco, C. and Neto, R. (2012). Using Genetic Algorithms to Solve Scheduling Problems on Flexible Manufacturing Systems (FMS): A Literature Survey, Classification and Analysis. *Flexible Services and Manufacturing Journal*, **26** (3), pp. 408 – 431.
- Fink, E. and Gandhi, H. (2011). Compression of Time Series by Extracting Major Extrema. *Journal of Experimental & Theoretical Artificial Intelligence*, **23** (2), pp. 255 – 270.
- Fink, E. and Pratt, K. (2004). Indexing of Compressed Time Series. In: M. Last, A. Kandel, and H. Bunke, eds., *Data Mining in Time Series Databases*, 1st ed. Singapore: World Scientific Publishing, pp. 43 – 66.
- Frepple. (2015). Belgium: FrePPLLe bvba.
- Fu, T., Chung, F., Lam, C., Luk, R. and Ng, C. (2005). Adaptive Data Delivery Framework for Financial Time Series Visualization. In: *Mobile Business, 2005. ICMB 2005. International Conference on*. New York City: IEEE, pp. 267 – 273.
- Gilani, S., Windmann, S., Pethig, F., Kroll, B. and Niggermann, O. (2013). The Importance of Model-Learning for the analysis of the Energy Consumption of Production Plants. In: *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, New York: IEEE, pp. 1 – 8.
- Gonçalves, J., Mendes, J. and Resende, M. (2008). A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem. *European Journal of Operational Research*, **189** (3), pp. 1171 – 1190.

- Gong, Y. and Ma, L. (2011). Research on Estimation of Energy Consumption in Machining Process based on CBR. In: *Industrial Engineering and Engineering Management (IE&EM), 2011 IEEE 18th International Conference on*, New York: IEEE, pp. 334 – 338.
- Gong, Y. and Ma, L. (2011). Research on Estimation of Energy Consumption in Machining Process based on CBR. In: *Industrial Engineering and Engineering Management (IE&EM), 2011 IEEE 18th International Conference on*, New York: IEEE, pp. 334 – 338.
- Gunn, S. (1998). *Support Vector Machines for Classification and Regression*. 1st ed. [PDF] Southampton: University of Southampton. Available from: <http://users.ecs.soton.ac.uk/srg/publications/pdf/SVM.pdf> [cited 16th December 2015].
- Gutowski, T., Dahmus, J. and Thiriez, A. (2006). Electrical Energy Requirements for Manufacturing Processes. In: *Life Cycle Engineering, 13th CIRP International Conference on*. [online] Paris: CIRP, pp. 623 – 628.
- He, Y., Liu, B., Zhang, X., Gao, H. and Liu, X. (2012). A Modelling Method of Task-Oriented Energy Consumption for Machining Manufacturing System. *Journal of Cleaner Production*, **23** (1), pp. 167 – 174.
- Herrera, F., Lozano, M. and Verdegay, J. (1998). Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*, **12** (4), pp. 265 – 319.
- Herrmann, C., Thiede, S., Kara, S. and Hesselbach, J. (2011). Energy Oriented Simulation of Manufacturing Systems – Concepts and Application. *CIRP Annals – Manufacturing Technology*, **60** (1), pp. 45 – 48.
- Ingarao, G., Vanhove, H., Kellens, K. and Doflou, J. (2014). A Comprehensive Analysis of Electric Energy Consumption Single Point Incremental Forming Processes. *Journal of Cleaner Production*, **67**, pp. 173 – 186.
- Kara, S. and Li W. (2011). Unit Process Energy Consumption Models for Material Removal Processes. *CIRP Annals – Manufacturing Technology*, **60** (1), pp. 37 – 40.

- Karaboga, D. (2000). *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. New York City: Springer-Verlag.
- Karger, D., Stein, C. and Wein, J. (2009). Scheduling Algorithms. In: M. Atallah and M. Blanton, eds., *Algorithms and Theory of Computation Handbook*, 2nd ed. London: Chapman and Hall, pp. 20.1 – 20.34.
- Keevey, S. and Smyth, K. (2015). *Time Series Simplification in kdb+: A Method for Dynamically Shrinking Big Data*. [online] Newry, NI: First Derivatives plc. Available from: <http://www.firstderivatives.com/> [cited 17th December 2015].
- Keogh, E. and Pazzani, M. (2003). A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases. *Knowledge Discovery and Data Mining. Current Issues and New Applications: 4th Pacific-Asia Conference, PAKDD 2000 Kyoto, Japan, April 18-20, 2000 Proceedings*, **1805**, pp. 122 – 133.
- Kibriya, A. and Frank, E. (2007). An Empirical Comparison of Exact Nearest Neighbour Algorithms. In: J. Kok, J. Koronacki, R. Mantaras, S. Matwin, D. Mladenič and A. Skowron, eds. *Knowledge Discovery in Databases: PFDD 2007*, 1st ed. Berlin: Springer-Verlag, pp. 140 – 151.
- Law, M. (2011) A Simply Introduction to Support Vector Machines.
- Lin, W., Lee, W. and Hong, T. (2003). Adapting Crossover and Mutation Rates in Genetic Algorithms. *Journal of Information Science and Engineering*, **19**, pp. 889 – 903.
- Liu, F., Xie, J. and Liu, S. (2015). A Method for Predicting the Energy Consumption of the Main Driving System of a Machine Tool in a Machining Process. *Journal of Cleaner Production*, **105**, pp. 171 – 177.
- Maxino, T. and Kooperman, P. (2009). The Effectiveness of Checksums for Embedded Control Networks. *Dependable and Secure Computing, IEEE Transactions on*, **6** (1), pp. 59 – 72.
- Microchip. (No date) PIC18F87J72 Energy Monitoring PICtail Plus Daughter Board. [online] Microchip. Available from: <http://www.microchip.com/> [cited 15th December 2015].

- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. Cambridge, MA: The MIT Press.
- Moulin, E. (2002). Measuring Reactive Power in Energy Meters. *Metering International*, (1), pp. 52 – 54.
- Mouzon, G., Yildirim, M. and Twomey, J. (2007). Operational Methods for Minimization of Energy Consumption of Manufacturing Equipment. *International Journal of Production Research*, **45** (18-19), pp. 4247 – 4271.
- National Grid. (No date) *Frequency Response Services*. [online] Available from: <http://www2.nationalgrid.com/uk/> [cited 15th December 2015].
- Newtons4th Ltd. (2012). *3 Phase 2 Wattmeter Power Measurements Explained*. [PDF] Newtons4th Ltd. Available from: <https://www.newtons4th.com> [cited 14th January 2016].
- nPower,(No date) *The Guide to Reactive Power: An npower briefing note*. [online] Available from: <http://npowerbrighterfutures.com/> [cited 15th December 2015].
- Orio, G., Candido, G., Barata, J., Bittencourt, J. and Bonfeld, R. (2013). Energy Efficiency in Machine Tools – A Self-Learning Approach. In: *Systems, Man and Cybernetics (SMC), 2013 IEEE International Conference on*. New York: IEEE, pp. 4878 – 4883.
- Pach, C., Berger, T., Sallez, Y., Bonte, Y., Adam, E. and Trentesaux, D. (2014). Reactive and Energy-Aware Scheduling of Flexible Manufacturing Systems using Potential Fields. *Computers in Industry*, **65** (3), pp. 434 – 448.
- Palpanas, T., Vlachos, M., Keogh, E., Gunopulos, D. and Truppel, W. (2004). Online Amnesic Approximation of Streaming Time Series. In: *Data Engineering, 2004. Proceedings 20th International Conference on*. New York City: IEEE, pp. 339 – 349.
- Pechmann, A. and Schöler, I. (2011). Optimizing Energy Costs by Intelligent Production Scheduling. In: J. Hesselbach and C. Herrmann, eds., *Glocalized Solutions for Sustainability in Manufacturing: Proceedings of the 18th CIRP International Conference on Life Cycle Engineering, Technische Universität Braunschweig, Braunschweig, Germany, May 2nd-4th 2011*, 1st ed. Berlin: Springer-Verlag, pp. 293 – 298.

- Pechmann, A., Schöler, I. and Hackmann, R. (2012). Energy Efficient and Intelligent Production Scheduling – Evaluation of a New Production Planning and Scheduling Software. In: D. Dornfeld and B. Linke, eds. *Leveraging Technology for a Sustainable World: Proceedings of the 19th CIRP Conference on Life Cycle Engineering, University of California at Berkeley, Berkeley, USA, May 23-25, 2012*, 1st ed. Berlin: Springer-Verlag, pp. 491 – 496.
- Peng, T., Xu, X. and Wang, L. (2014). A Novel Energy Demand Modelling Approach for CNC Machining based on Function Blocks. *Journal of Manufacturing Systems*, **33** (1), pp. 196 – 208.
- Pinedo, M. (2012). *Scheduling: Theory, Algorithms and Systems*. 4th ed. New York: Springer-Verlag, pp. 1.
- Polikar, R. (2006). Ensemble based Systems in Decision Making. *Circuits and System Magazine, IEEE*, **6** (3), pp. 21 – 45.
- Povinelli, R. (2000). Improving Computational Performance of Genetic Algorithms: A Comparison of Techniques. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Burlington, MA: Morgan Kaufmann, pp. 297 -302.
- Povinelli, R. and Feng, X. (1999). Improving Genetic Algorithms Performance by Hashing Fitness Values. In: *Proceedings Artificial Neural Network in Engineering*, pp. 399 – 404.
- Qiu, H., Zhou, W. and Wang, H. (2009). A Genetic Algorithms-Based Approach to Flexible Job-Shop Scheduling Problem. In: *Natural Computation, 2009. ICNI '09. Fifth International Conference on*. New York: IEEE, pp. 81 – 85.
- Qureshi, F., Li, W., Kara, S. and Herrmann, C. (2012). Unit Process Energy Consumption Models for Material Addition Processes: A Case of the Injection Molding Process. In: Dornfeld, D. and Linke, B. eds., *Leveraging Technology for a Sustainable World: Proceedings of the 19th CIRP Conference on Life Cycle Engineering, University of*

California at Berkeley, Berkeley, USA, May 23-25, 2012, 1st ed. Berlin: Springer-Verlag, pp. 269 – 274.

Rager, M., Gahm, C. and Denz, F. (2015). Energy-Oriented Scheduling based on Evolutionary Algorithms. *Computers & Operations Research*, **54**, pp. 218 – 231.

Ringwelski, M., Renner, C., Reinhardt, A., Weigel, A. and Turau, V. (2012). The Hitchhiker's Guide to Choosing the Compression Algorithm for your Smart Meter Data. In: *Energy Conference and Exhibition (ENERGY CON), 2012 IEEE International*, New York City: IEEE, pp. 935 – 940.

Roth A. Giffi, and Chaudhuri, A. (2010). *2010 Global Manufacturing Competitiveness Index* [Online]. London: Deloitte Ltd. pp. 7. Available from: <http://www2.deloitte.com/> [cited 14th December 2015].

S. M. (2012). myPowerMoitor. [online] NI Community. Available from: <https://decibel.ni.com/content/docs/DOC-23214> [cited 15th December 2015].

Shevade, K., Keerthi, S., Bhattacharyya, C. and Murthy, K. (2000). Improvements to the SMO Algorithms for SVM Regression. *Neural Networks, IEEE Transactions on*, **11** (5), 1188 – 1193.

Shin, S-J., Woo, J. and Rachuri, S. (2014). Predictive Analytics Model for Power Consumption in Manufacturing. *Procedia CIRP*, **15**, pp. 153 – 158.

Shrouf, F., Ordieres-Meré, J., García-Sánchez, A. and Ortega-Mier, M. (2014). Optimizing the Production Scheduling of a Single Machine to Minimize Total Energy Consumption Costs. *Journal of Cleaner Production*, **67**, pp. 197 – 207.

Shrivastava, A., Overcash, M. and PfefferKorn, F. (2015). Prediction of Unit Process Life Cycle Inventory (UPLCI) Energy Consumption in a Friction Stir Weld. *Journal of Manufacturing Processes*, **18**, pp. 46 – 54.

- Standards & Technical Regulations Directorate, (2005). *Electrical Supply Tolerances and Electrical Appliance Safety*. [online] Available from: <https://www.gov.uk/> [cited 15th December 2015].
- Sung, M. and Ko, Y. (2015). Machine-Learning-Integrated Load Scheduling for Reduced Peak Power Demand. *IEEE Transactions on Consumer Electronics*, **61** (2), pp. 167 – 174.
- Tektronix (No date). *The Fundamentals of Three-Phase Power Measurements: Application Note*. [online] Tektronix. Available from: www.tek.com/dl/55W_28943_0_HR_Letter.pdf [cited 15th December 2015].
- United Nations Framework Convention on Climate Change (No date). *Kyoto Protocol*. [Online]. Available from: http://unfccc.int/kyoto_protocol/items/2830.php [cited 14th December 2015].
- Unterweger, A. and Engel, D. (2015). Resumable Load Data Compression in Smart Grids. *Smart Grid, IEEE Transactions on*, **6** (2), pp. 919 – 929.
- Velchev, S., Kolev, I., Ivanov, K. and Gechevski, S. (2014). Empirical Models for Specific Energy Consumption and Optimization of Cutting Parameters for Minimizing Energy Consumption during Turning. *Journal of Cleaner Production*, **80**, pp. 139 – 149.
- Ware, J. (2006). *Power Factor Correction*. [online] Available from: electrical.theiet.org/wiring-matters/18/power-factor.cfm?type=pdf [cited 15th December 2015].
- Weinert, N., Chiotellis, S. and Seliger, G. (2011). Methodology for Planning and Operating Energy-Efficient Production Systems. *CIRP Annals – Manufacturing Technology*, **60** (1), pp. 41 – 44.
- Weka 3: Data Mining Software in Java. (2015). Hamilton: University of Waikato.
- Windows Dev Center (No date) *Improving Garbage Collection Performance (Windows Store Apps using C#/VB)*. [online] Available from: <https://msdn.microsoft.com/en-us> [cited 17th December 2015].

- Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* – Chapter 4.7. 2nd ed. Massachusetts: Morgan Kaufmann, pp. 128 – 136.
- Xu, F., Weng, W. and Fujimura, S. (2014). Energy-Efficient Schedule for Flexible Flow Shops by using MIPS. In: *IIE Annual Conference and Expo 2014. Institute of Industrial Engineers*. Norcross, GA: Institute of Industrial Engineers, pp. 1040 – 1048.
- Yang, X. (2014). *Nature-Inspired Optimization Algorithms*. Amsterdam, Elsevier.
- Yildirim, M. and Mouzon, G. (2012). Single-Machine Sustainable Production Planning to Minimize Total Energy Consumption and Total Completion Time Using a Multiple Objective Genetic Algorithm. *Engineering Management, IEEE Transactions on*. **59** (4), pp. 585 – 597.
- Zaniolo, C. (2008) *Mining Time Series Data (CS240B)*.
- Zanoni, S., Bettoni, L. and Glock, C. (2014). Energy Implications in a Two-Stage Production System with Controllable Production Rates. *International Journal of Production Economics*, **149**, pp. 164 – 171.

APPENDIX

Appendix A: Custom Energy Monitor Schematic

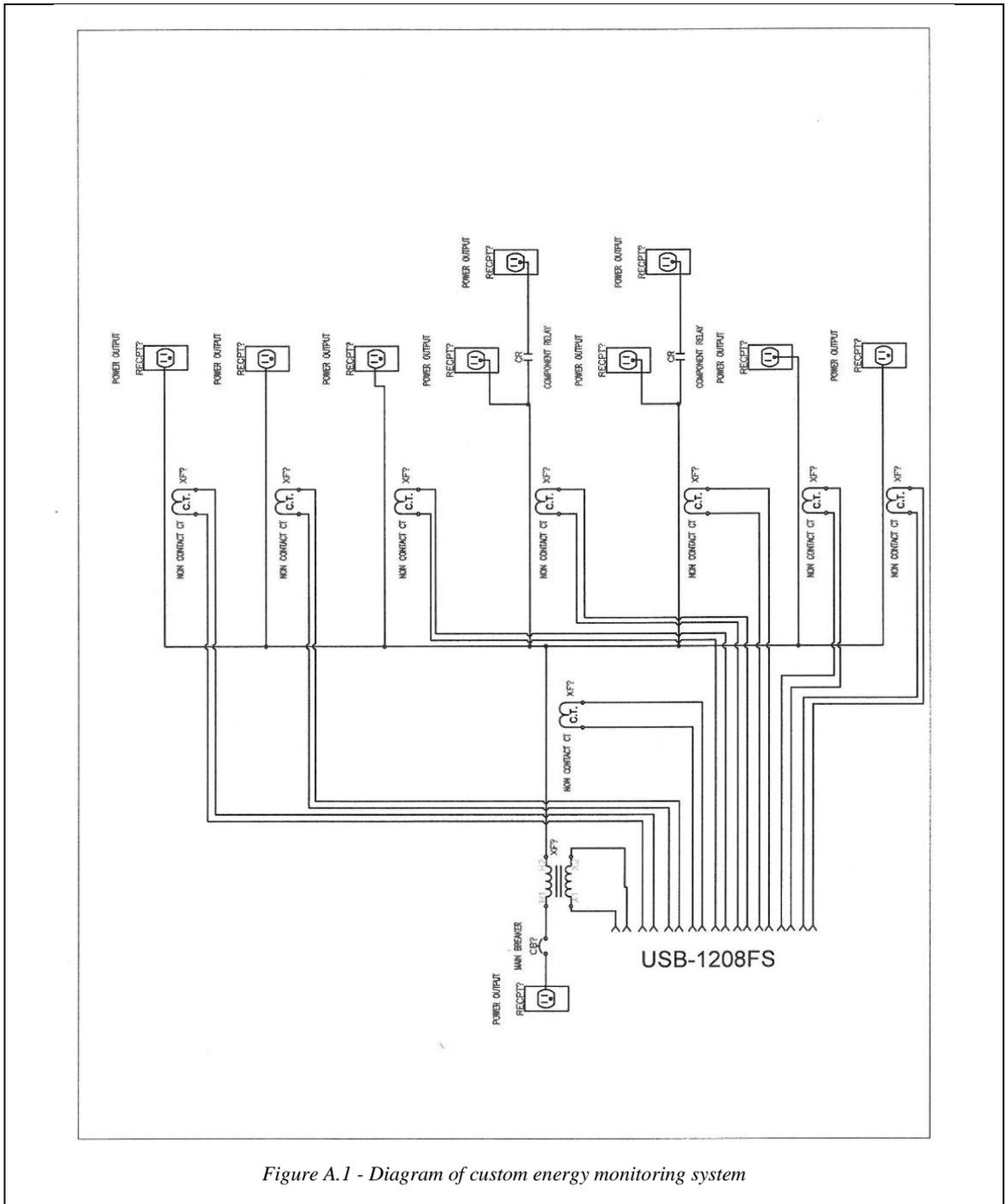


Figure A.1 - Diagram of custom energy monitoring system

Appendix B: Parameters for Evaluated Machine Learning Algorithms

Standalone Multilayer Perceptron (Standalone & as base learner)

- Learning rate: 0.05 (standalone), 0.3 (base learner)
- Momentum: 0.2
- Training time: 500 epochs
- Validation threshold: 20

Standalone Support Vector Regression (Standalone & as base learner)

- Complexity: 1
- Kernel parameters
 - Polynomial kernel
 - Cache size: 250007
 - Exponent value: 1
 - Pearson VII Universal kernel
 - Cache size: 250007
 - Omega: 1
 - Sigma: 1
- Training algorithm
 - Epsilon value: 1^{-12}
 - Parameter of the epsilon intensive loss function: 0.001
 - Tolerance value for the checking stopping criterion: 0.001

Linear Regression (Standalone & as base learner)

- Attribute selection method: M5 method
- Ridge: 1^{-8}

Bagging (Bootstrap aggregating)

- Size of bag: 100% of training set
- Number of iterations: 10
- Random number seed: 1

Stacking (Stacked generalization)

- Number of folds for cross-validation: 10
- Random number seed: 1

Appendix C: Full Results from Evaluating Machine Learning Algorithms to Generate Synthetic Energy Profiles

Name of algorithm	Abbreviation used
Support Vector Regression with a Polynomial kernel	SVR w/ Poly
Support Vector Regression with a Pearson VII kernel	SVR w/ PUK
Bagging ensemble with Multilayer Perceptron base learners	Bagging w/ MLP
Bagging ensemble with Support Vector Regression with a Polynomial kernel as base learners	Bagging w/ SVR w/ Poly
Bagging ensemble with Support Vector Regression with a Pearson VII kernel as base learners	Bagging w/ SVR w/ PUK
Bagging ensemble with Linear Regression base learners	Bagging w/ Linear reg.
Stacking ensemble with Multilayer Perceptron base learners	Stacking w/ MLP
Stacking ensemble with Support Vector Regression with a Polynomial kernel as base learners	Stacking w/ SVR w/ Poly
Stacking ensemble with Support Vector Regression with a Pearson VII kernel as base learners	Stacking w/ SVR w/ PUK
Stacking ensemble with Linear Regression base learners	Stacking w/ Linear reg.

Training set size	Algorithm	Training					Testing		
		Correlation coefficient	Mean error (RMS A)	RMS error (RMS A)	Relative absolute error (%)	Root relative squared error (%)	Correlation coefficient	Mean absolute error (RMS A)	RMS error (RMS A)
5	Multilayer Perceptron	0.161	1.723	2.692	18.470	24.327	0.293	0.871	1.040
10		0.343	0.360	0.494	5.988	6.734	0.310	0.693	0.816
20		0.345	0.273	0.385	4.463	5.134	0.347	0.236	0.274
100		0.348	0.188	0.279	2.939	3.648	0.349	0.283	0.307
5	SVR w/ Poly	0.212	1.377	2.200	18.453	23.133	0.320	0.550	0.664
10		0.327	0.254	0.430	4.249	5.476	0.332	0.158	0.186
20		0.331	0.151	0.274	2.665	3.768	0.332	0.120	0.153
100		0.332	0.148	0.243	2.366	3.241	0.333	0.312	0.323
5	SVR w/ PUK	-0.235	2.300	3.206	25.203	29.978	0.223	1.443	1.717
10		0.319	0.497	0.731	8.242	9.799	0.241	1.215	1.599
20		0.324	0.350	0.535	6.234	7.951	0.272	0.561	0.985
100		0.331	0.181	0.278	2.965	3.781	0.291	0.671	1.040
5	Linear	-0.214	3.122	2.652	33.333	33.333	0.275	0.722	0.875

10	Regression	0.328	0.281	0.405	3.950	4.469	0.319	0.374	0.459
20		0.330	0.179	0.282	3.007	3.781	0.331	0.175	0.216
100		0.332	0.153	0.237	2.452	3.140	0.333	0.282	0.296
5	Bagging w/ MLP	0.226	1.372	2.256	19.354	24.534	0.264	0.930	1.110
10		0.327	0.453	0.549	6.599	6.705	0.287	0.712	0.867
20		0.329	0.267	0.375	4.442	5.090	0.328	0.241	0.299
100		0.331	0.180	0.265	2.945	3.574	0.332	0.298	0.318
5	Bagging w/ SVR w/ Poly	0.144	1.489	2.580	19.764	25.961	0.312	0.971	1.131
10		0.329	0.310	0.447	4.805	5.446	0.332	0.222	0.267
20		0.331	0.139	0.267	2.394	3.578	0.333	0.119	0.137
100		0.332	0.148	0.243	2.362	3.238	0.333	0.313	0.325
5	Bagging w/ SVR w/ PUK	-0.235	2.300	3.206	25.203	26.377	0.163	2.463	2.727
10		0.299	0.855	1.269	13.538	16.660	0.217	1.276	1.689
20		0.319	0.457	0.727	8.074	10.737	0.252	0.725	1.163
100		0.331	0.182	0.275	2.986	3.725	0.287	0.670	1.050
5	Bagging w/ Linear Reg.	-0.214	3.197	3.664	33.786	33.616	0.314	0.715	0.862
10		0.330	0.269	0.360	3.866	4.114	0.324	0.310	0.378
20		0.330	0.179	0.281	2.971	3.736	0.331	0.178	0.225
100		0.332	0.155	0.238	2.480	3.156	0.333	0.286	0.302
5	Stacking w/ MLP	0.039	3.480	3.851	43.588	38.197	0.188	3.856	4.606
10		0.315	0.618	0.787	10.748	11.369	0.289	0.731	0.874
20		0.326	0.363	0.485	6.299	6.937	0.328	0.472	0.550
100		0.330	0.281	0.374	4.444	5.008	0.332	0.427	0.478
5	Stacking w/ SVR w/ Poly	-0.020	3.089	3.854	36.970	37.081	0.320	3.792	1.290
10		0.328	0.310	0.447	5.279	5.966	0.333	0.199	0.233
20		0.331	0.154	0.273	2.757	3.886	0.332	0.129	0.175
100		0.332	0.148	0.244	2.367	3.247	0.333	0.317	0.328
5	Stacking w/ SVR w/ PUK	-0.028	2.925	3.346	37.479	34.078	-0.054	2.484	2.889
10		0.163	1.356	2.033	23.672	28.720	0.235	1.302	1.539
20		0.322	0.456	0.613	7.940	8.710	0.273	0.569	0.958
100		0.331	0.192	0.289	3.119	3.919	0.292	0.652	1.017

5	Stacking w/ Linear Reg.	-0.217	3.808	5.427	48.123	52.303	0.076	3.770	4.949
10		0.328	0.312	0.446	4.551	5.140	0.319	0.408	0.502
20		0.330	0.176	0.282	2.962	3.786	0.331	0.179	0.224
100		0.332	0.153	0.237	2.449	3.331	0.333	0.283	0.296

Appendix D: Waveforms used during Evaluation of Time-Series Compression Algorithms

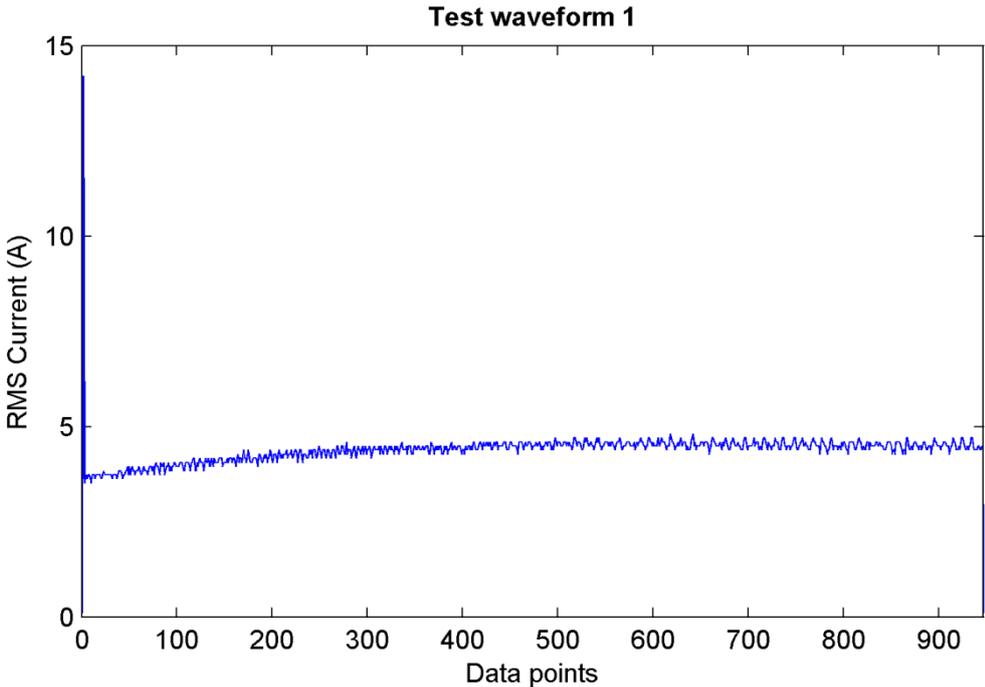


Figure D.1 - Compression algorithms test waveform 1

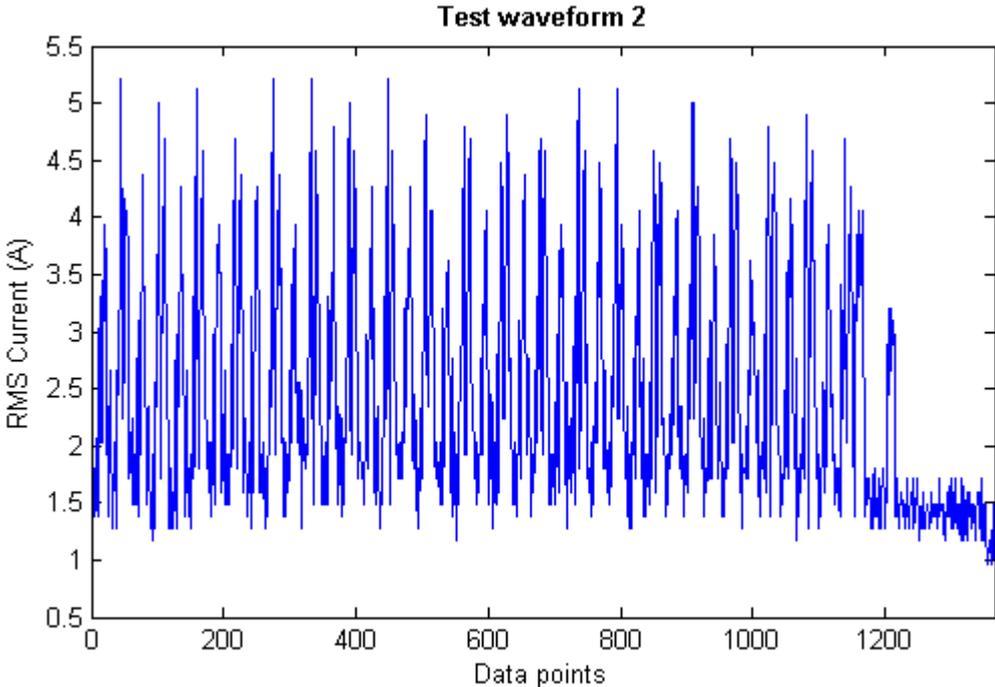


Figure D.2 - Compression algorithms test waveform 2

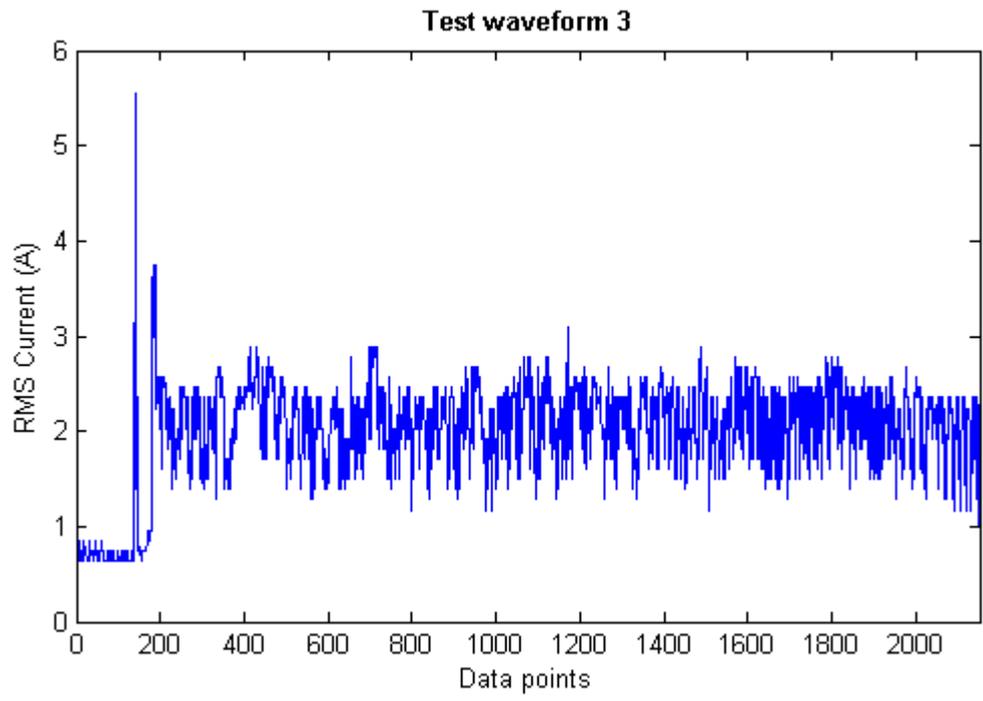


Figure D.3 - Compression algorithms test waveform 3

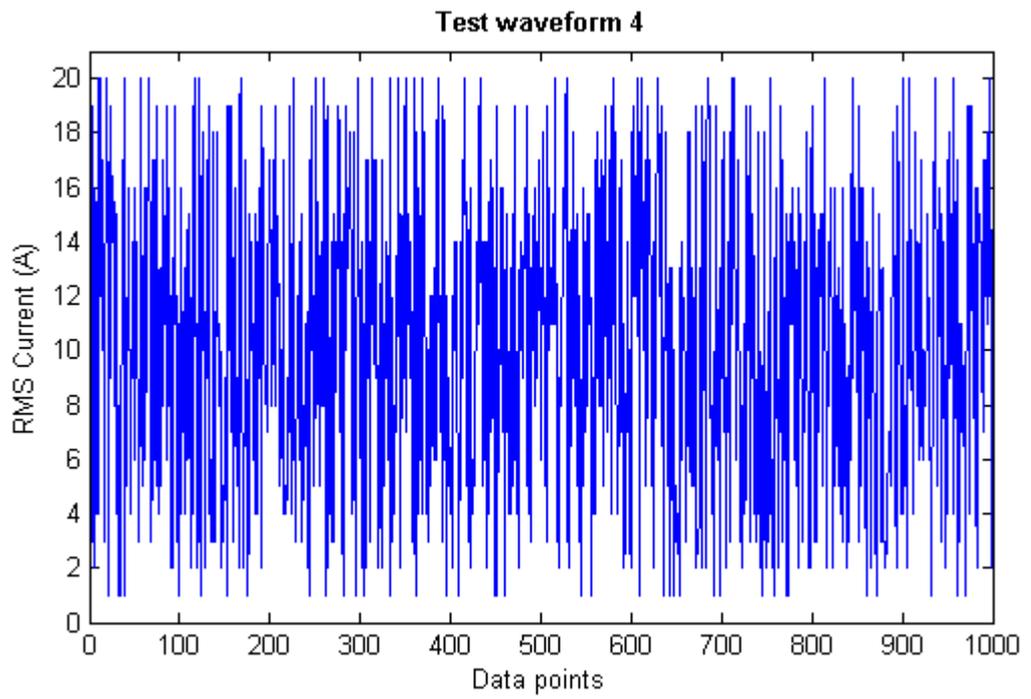


Figure D.4 - Compression algorithms test waveform 4

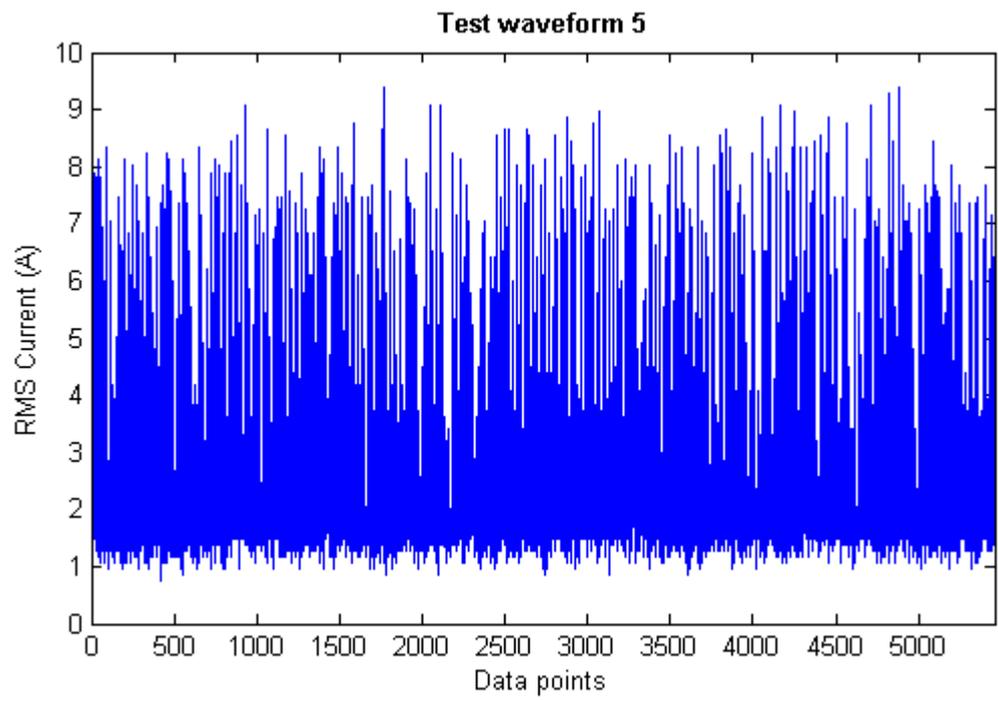


Figure D.5 - Compression algorithms test waveform 5

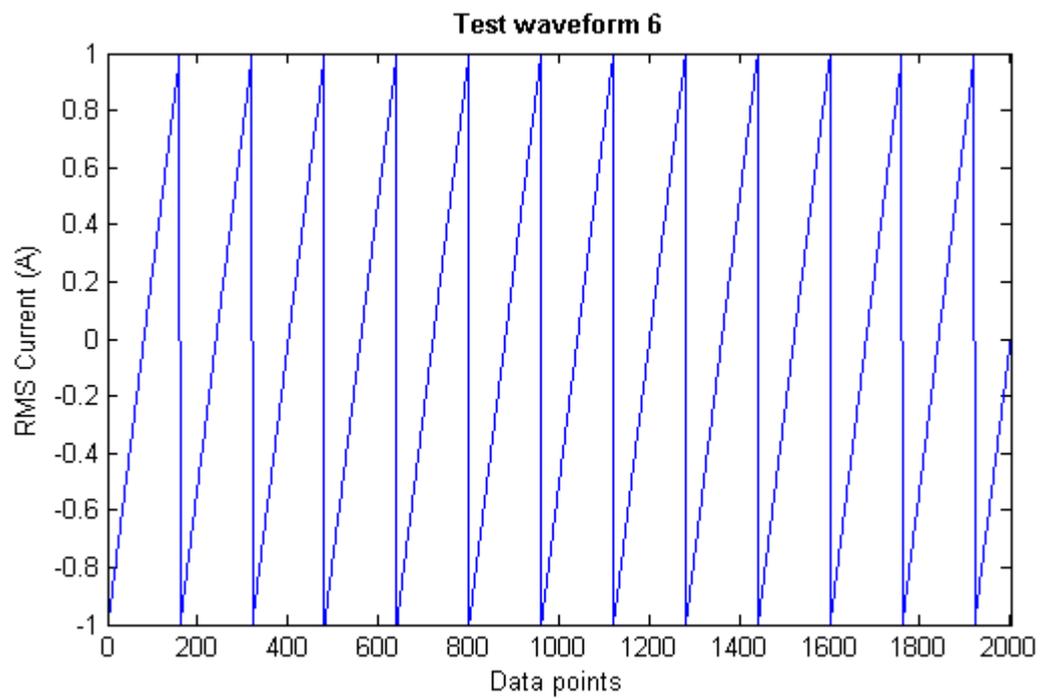


Figure D.6 - Compression algorithms test waveform 6

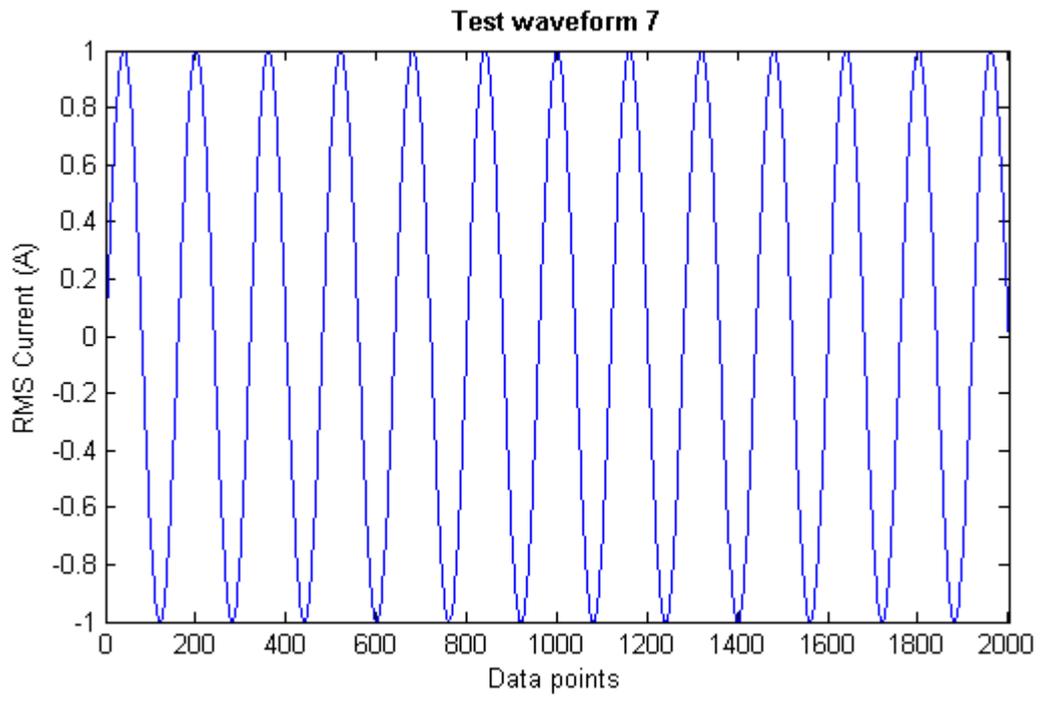


Figure D.7 - Compression algorithms test waveform 7

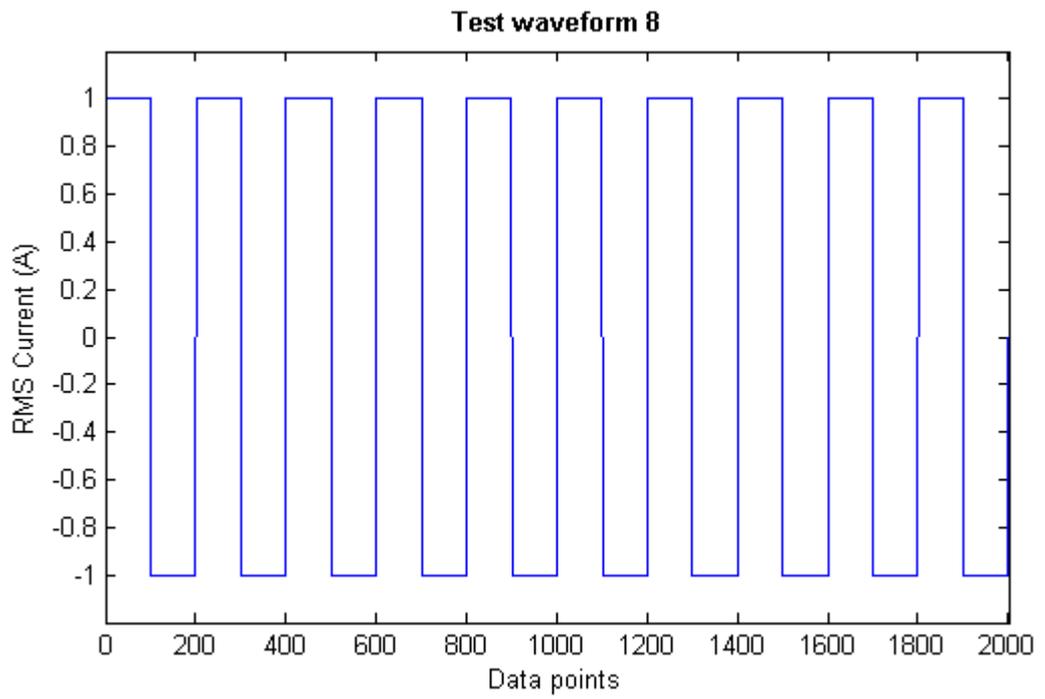


Figure D.8 - Compression algorithms test waveform 8

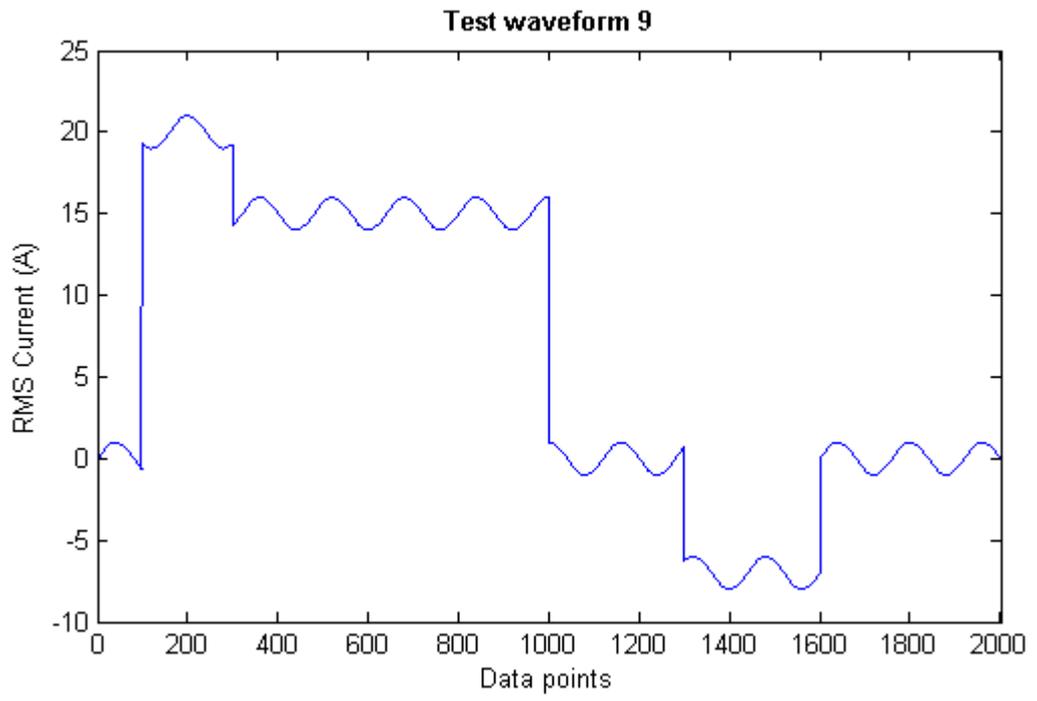


Figure D.9 - Compression algorithms test waveform 9

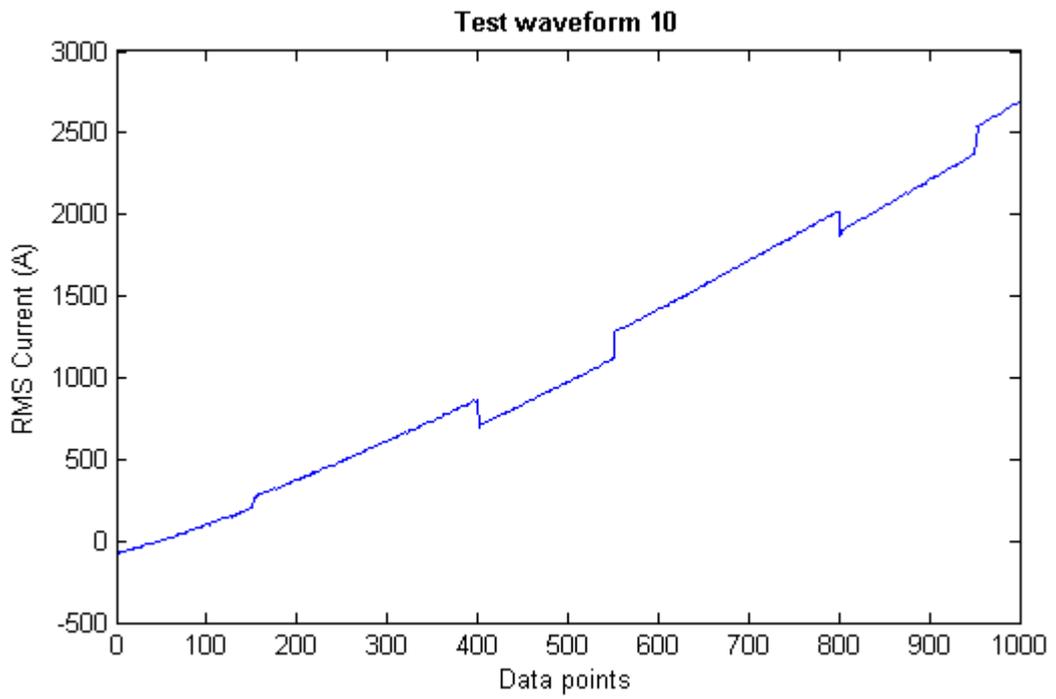


Figure D.10 - Compression algorithms test waveform 10

Appendix E: Details of Test Schedules

Test Schedule #1

Process ID	A	Process release date	11:30
Process priority	1	Process Deadline	12:50
Job ID	P1A-001	Job start time	12:15
Job Duration	00:10	Prerequisite jobs	-
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-002	Job start time	12:25
Job Duration	00:05	Prerequisite jobs	P1A-001
Default machine	M1A-001	Alternative machines	-
Job ID	P1A-003	Job start time	12:30
Job Duration	00:20	Prerequisite jobs	P1A-002
Default machine	M1A-010	Alternative machines	-
Process ID	B	Process release date	12:00
Process priority	1	Process Deadline	13:10
Job ID	P1B-001	Job start time	12:17
Job Duration	00:35	Prerequisite jobs	-
Default machine	M1A-004	Alternative machines	-
Job ID	P1B-002	Job start time	12:52
Job Duration	00:18	Prerequisite jobs	P1B-001
Default machine	M1A-005	Alternative machines	-
Process ID	C	Process release date	11:00
Process priority	1	Process Deadline	13:00
Job ID	P1C-001	Job start time	12:05
Job Duration	00:05	Prerequisite jobs	-
Default machine	M1A-008	Alternative machines	-
Job ID	P1C-002	Job start time	12:10
Job Duration	00:20	Prerequisite jobs	P1C-001
Default machine	M1A-002	Alternative machines	-
Job ID	P1C-003	Job start time	12:42
Job Duration	00:10	Prerequisite jobs	P1C-002
Default machine	M1A-009	Alternative machines	-

Test Schedule #2

Process ID	A	Process release date	10:30
Process priority	1	Process Deadline	12:30
Job ID	P1A-001	Job start time	11:05
Job Duration	00:10	Prerequisite jobs	-
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-002	Job start time	11:28

	Job Duration	00:05	Prerequisite jobs	P1A-001
	Default machine	M1A-001	Alternative machines	-
	Job ID	P1A-003	Job start time	11:33
	Job Duration	00:20	Prerequisite jobs	P1A-002
	Default machine	M1A-010	Alternative machines	-
	Job ID	P1A-004	Job start time	11:53
	Job Duration	00:20	Prerequisite jobs	P1A-003
	Default machine	M1A-006	Alternative machines	-
	Job ID	P1A-005	Job start time	12:13
	Job Duration	00:02	Prerequisite jobs	P1A-004
	Default machine	M1A-001	Alternative machines	-
	Job ID	P1A-006	Job start time	12:15
	Job Duration	00:15	Prerequisite jobs	P1A-005
	Default machine	M1A-003	Alternative machines	-
	Process ID	B	Process release date	09:50
	Process priority	1	Process Deadline	11:50
	Job ID	P1B-001	Job start time	10:20
	Job Duration	00:35	Prerequisite jobs	-
	Default machine	M1A-004	Alternative machines	-
	Job ID	P1B-002	Job start time	11:32
	Job Duration	00:18	Prerequisite jobs	P1B-001
	Default machine	M1A-005	Alternative machines	-
	Process ID	C	Process release date	08:00
	Process priority	1	Process Deadline	10:20
	Job ID	P1C-001	Job start time	09:05
	Job Duration	00:05	Prerequisite jobs	-
	Default machine	M1A-008	Alternative machines	-
	Job ID	P1C-002	Job start time	09:10
	Job Duration	00:20	Prerequisite jobs	P1C-001
	Default machine	M1A-002	Alternative machines	-
	Job ID	P1C-003	Job start time	09:30
	Job Duration	00:10	Prerequisite jobs	P1C-002
	Default machine	M1A-009	Alternative machines	-
	Job ID	P1C-004	Job start time	09:40
	Job Duration	00:40	Prerequisite jobs	P1C-003
	Default machine	M1A-007	Alternative machines	-

Test Schedule #3

Process ID	A	Process release date	09:30
Process priority	1	Process Deadline	12:15
Job ID	P1A-001	Job start time	11:05
Job Duration	00:10	Prerequisite jobs	-
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-002	Job start time	11:28
Job Duration	00:05	Prerequisite jobs	P1A-001
Default machine	M1A-001	Alternative machines	-
Job ID	P1A-003	Job start time	11:33
Job Duration	00:20	Prerequisite jobs	P1A-002
Default machine	M1A-010	Alternative machines	-
Job ID	P1A-004	Job start time	11:53
Job Duration	00:20	Prerequisite jobs	P1A-003
Default machine	M1A-006	Alternative machines	-
Job ID	P1A-005	Job start time	12:13
Job Duration	00:02	Prerequisite jobs	P1A-004
Default machine	M1A-001	Alternative machines	-
Process ID	B	Process release date	09:00
Process priority	1	Process Deadline	12:50
Job ID	P1B-001	Job start time	10:20
Job Duration	00:35	Prerequisite jobs	-
Default machine	M1A-004	Alternative machines	-
Job ID	P1B-002	Job start time	11:32
Job Duration	00:18	Prerequisite jobs	P1B-001
Default machine	M1A-005	Alternative machines	-
Job ID	P1B-003	Job start time	12:05
Job Duration	00:40	Prerequisite jobs	P1B-002
Default machine	M1A-002	Alternative machines	-
Job ID	P1B-004	Job start time	12:45
Job Duration	00:05	Prerequisite jobs	P1B-003
Default machine	M1A-010	Alternative machines	-
Process ID	C	Process release date	07:00
Process priority	1	Process Deadline	11:15
Job ID	P1C-001	Job start time	09:05
Job Duration	00:05	Prerequisite jobs	-
Default machine	M1A-008	Alternative machines	-
Job ID	P1C-002	Job start time	09:10
Job Duration	00:20	Prerequisite jobs	P1C-001
Default machine	M1A-002	Alternative machines	-

Job ID	P1C-003	Job start time	09:30
Job Duration	00:10	Prerequisite jobs	P1C-002
Default machine	M1A-009	Alternative machines	-
Job ID	P1C-004	Job start time	09:40
Job Duration	00:40	Prerequisite jobs	P1C-003
Default machine	M1A-007	Alternative machines	-
Job ID	P1C-005	Job start time	10:20
Job Duration	00:35	Prerequisite jobs	P1C-004
Default machine	M1A-010	Alternative machines	-
Job ID	P1C-006	Job start time	10:55
Job Duration	00:20	Prerequisite jobs	P1C-005
Default machine	M1A-004	Alternative machines	-
Process ID	D	Process release date	07:00
Process priority	1	Process Deadline	11:53
Job ID	P1D-001	Job start time	07:50
Job Duration	00:30	Prerequisite jobs	-
Default machine	M1A-001	Alternative machines	-
Job ID	P1D-002	Job start time	08:20
Job Duration	00:20	Prerequisite jobs	P1D-002
Default machine	M1A-002	Alternative machines	-
Job ID	P1D-003	Job start time	08:40
Job Duration	00:40	Prerequisite jobs	P1D-003
Default machine	M1A-009	Alternative machines	-
Job ID	P1D-004	Job start time	09:20
Job Duration	01:00	Prerequisite jobs	P1D-004
Default machine	M1A-010	Alternative machines	-
Job ID	P1D-005	Job start time	11:34
Job Duration	00:19	Prerequisite jobs	P1D-005
Default machine	M1A-007	Alternative machines	-

Test Schedule #4

Process ID	A	Process release date	09:00
Process priority	1	Process Deadline	13:00
Job ID	P1A-001	Job start time	11:05
Job Duration	00:10	Prerequisite jobs	-
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-002	Job start time	11:28
Job Duration	00:05	Prerequisite jobs	P1A-001
Default machine	M1A-001	Alternative machines	-
Job ID	P1A-003	Job start time	11:33
Job Duration	00:20	Prerequisite jobs	P1A-002

	Default machine	M1A-010	Alternative machines	-
	Job ID	P1A-004	Job start time	11:53
	Job Duration	00:20	Prerequisite jobs	P1A-003
	Default machine	M1A-006	Alternative machines	-
	Job ID	P1A-005	Job start time	12:13
	Job Duration	00:02	Prerequisite jobs	P1A-004
	Default machine	M1A-001	Alternative machines	-
	Job ID	P1A-006	Job start time	12:15
	Job Duration	00:15	Prerequisite jobs	P1A-005
	Default machine	M1A-003	Alternative machines	-
	Job ID	P1A-007	Job start time	12:30
	Job Duration	00:05	Prerequisite jobs	P1A-006
	Default machine	M1A-007	Alternative machines	-
	Job ID	P1A-008	Job start time	12:35
	Job Duration	00:05	Prerequisite jobs	P1A-007
	Default machine	M1A-008	Alternative machines	-
	Job ID	P1A-009	Job start time	12:40
	Job Duration	00:10	Prerequisite jobs	P1A-008
	Default machine	M1A-009	Alternative machines	-
	Job ID	P1A-010	Job start time	12:50
	Job Duration	00:10	Prerequisite jobs	P1A-009
	Default machine	M1A-001	Alternative machines	-
	Process ID	B	Process release date	09:00
	Process priority	1	Process Deadline	14:00
	Job ID	P1B-001	Job start time	10:20
	Job Duration	00:35	Prerequisite jobs	-
	Default machine	M1A-004	Alternative machines	-
	Job ID	P1B-002	Job start time	11:32
	Job Duration	00:18	Prerequisite jobs	P1B-001
	Default machine	M1A-005	Alternative machines	-
	Job ID	P1B-003	Job start time	12:05
	Job Duration	00:40	Prerequisite jobs	P1B-002
	Default machine	M1A-002	Alternative machines	-
	Job ID	P1B-004	Job start time	12:45
	Job Duration	00:05	Prerequisite jobs	P1B-003
	Default machine	M1A-010	Alternative machines	-
	Job ID	P1B-005	Job start time	12:50
	Job Duration	00:15	Prerequisite jobs	P1B-004
	Default machine	M1A-007	Alternative machines	-
	Job ID	P1B-006	Job start time	13:05

Job Duration	00:10	Prerequisite jobs	P1B-005
Default machine	M1A-009	Alternative machines	-
Job ID	P1B-007	Job start time	13:15
Job Duration	00:10	Prerequisite jobs	P1B-006
Default machine	M1A-006	Alternative machines	-
Job ID	P1B-008	Job start time	13:25
Job Duration	00:10	Prerequisite jobs	P1B-007
Default machine	M1A-004	Alternative machines	-
Job ID	P1B-009	Job start time	13:35
Job Duration	00:20	Prerequisite jobs	P1B-008
Default machine	M1A-005	Alternative machines	-
Job ID	P1B-010	Job start time	13:55
Job Duration	00:05	Prerequisite jobs	P1B-009
Default machine	M1A-002	Alternative machines	-
Process ID	C	Process release date	07:00
Process priority	1	Process Deadline	12:00
Job ID	P1C-001	Job start time	09:05
Job Duration	00:05	Prerequisite jobs	-
Default machine	M1A-008	Alternative machines	-
Job ID	P1C-002	Job start time	09:10
Job Duration	00:20	Prerequisite jobs	P1C-001
Default machine	M1A-002	Alternative machines	-
Job ID	P1C-003	Job start time	09:30
Job Duration	00:10	Prerequisite jobs	P1C-002
Default machine	M1A-009	Alternative machines	-
Job ID	P1C-004	Job start time	09:40
Job Duration	00:40	Prerequisite jobs	P1C-003
Default machine	M1A-007	Alternative machines	-
Job ID	P1C-005	Job start time	10:20
Job Duration	00:35	Prerequisite jobs	P1C-004
Default machine	M1A-010	Alternative machines	-
Job ID	P1C-006	Job start time	10:55
Job Duration	00:20	Prerequisite jobs	P1C-005
Default machine	M1A-004	Alternative machines	-
Job ID	P1C-007	Job start time	11:15
Job Duration	00:05	Prerequisite jobs	P1C-006
Default machine	M1A-003	Alternative machines	-
Job ID	P1C-008	Job start time	11:20
Job Duration	00:15	Prerequisite jobs	P1C-007
Default machine	M1A-006	Alternative machines	-

Job ID	P1C-009	Job start time	11:35
Job Duration	00:15	Prerequisite jobs	P1C-008
Default machine	M1A-001	Alternative machines	-
Job ID	P1C-010	Job start time	11:50
Job Duration	00:10	Prerequisite jobs	P1C-009
Default machine	M1A-005	Alternative machines	-

Test Schedule #5

Process ID	A	Process release date	09:00
Process priority	1	Process Deadline	13:00
Job ID	P1A-001	Job start time	11:05
Job Duration	00:10	Prerequisite jobs	-
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-002	Job start time	11:28
Job Duration	00:05	Prerequisite jobs	P1A-001
Default machine	M1A-001	Alternative machines	-
Job ID	P1A-003	Job start time	11:33
Job Duration	00:20	Prerequisite jobs	P1A-002
Default machine	M1A-010	Alternative machines	-
Job ID	P1A-004	Job start time	11:53
Job Duration	00:20	Prerequisite jobs	P1A-003
Default machine	M1A-006	Alternative machines	-
Job ID	P1A-005	Job start time	12:13
Job Duration	00:02	Prerequisite jobs	P1A-004
Default machine	M1A-001	Alternative machines	-
Job ID	P1A-006	Job start time	12:15
Job Duration	00:15	Prerequisite jobs	P1A-005
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-007	Job start time	12:30
Job Duration	00:05	Prerequisite jobs	P1A-006
Default machine	M1A-007	Alternative machines	-
Job ID	P1A-008	Job start time	12:35
Job Duration	00:05	Prerequisite jobs	P1A-007
Default machine	M1A-008	Alternative machines	-
Job ID	P1A-009	Job start time	12:40
Job Duration	00:10	Prerequisite jobs	P1A-008
Default machine	M1A-009	Alternative machines	-
Job ID	P1A-010	Job start time	12:50
Job Duration	00:10	Prerequisite jobs	P1A-009
Default machine	M1A-001	Alternative machines	-

Process ID	B	Process release date	09:00
Process priority	1	Process Deadline	14:00
Job ID	P1B-001	Job start time	10:20
Job Duration	00:35	Prerequisite jobs	-
Default machine	M1A-004	Alternative machines	-
Job ID	P1B-002	Job start time	11:32
Job Duration	00:18	Prerequisite jobs	P1B-001
Default machine	M1A-005	Alternative machines	-
Job ID	P1B-003	Job start time	12:05
Job Duration	00:40	Prerequisite jobs	P1B-002
Default machine	M1A-002	Alternative machines	-
Job ID	P1B-004	Job start time	12:45
Job Duration	00:05	Prerequisite jobs	P1B-003
Default machine	M1A-010	Alternative machines	-
Job ID	P1B-005	Job start time	12:50
Job Duration	00:15	Prerequisite jobs	P1B-004
Default machine	M1A-007	Alternative machines	-
Job ID	P1B-006	Job start time	13:05
Job Duration	00:10	Prerequisite jobs	P1B-005
Default machine	M1A-009	Alternative machines	-
Job ID	P1B-007	Job start time	13:15
Job Duration	00:10	Prerequisite jobs	P1B-006
Default machine	M1A-006	Alternative machines	-
Job ID	P1B-008	Job start time	13:25
Job Duration	00:10	Prerequisite jobs	P1B-007
Default machine	M1A-004	Alternative machines	-
Job ID	P1B-009	Job start time	13:35
Job Duration	00:20	Prerequisite jobs	P1B-008
Default machine	M1A-005	Alternative machines	-
Job ID	P1B-010	Job start time	13:55
Job Duration	00:05	Prerequisite jobs	P1B-009
Default machine	M1A-002	Alternative machines	-
Process ID	C	Process release date	07:00
Process priority	1	Process Deadline	12:00
Job ID	P1C-001	Job start time	09:05
Job Duration	00:05	Prerequisite jobs	-
Default machine	M1A-008	Alternative machines	-
Job ID	P1C-002	Job start time	09:10
Job Duration	00:20	Prerequisite jobs	P1C-001
Default machine	M1A-002	Alternative machines	-

Job ID	P1C-003	Job start time	09:30
Job Duration	00:10	Prerequisite jobs	P1C-002
Default machine	M1A-009	Alternative machines	-
Job ID	P1C-004	Job start time	09:40
Job Duration	00:40	Prerequisite jobs	P1C-003
Default machine	M1A-007	Alternative machines	-
Job ID	P1C-005	Job start time	10:20
Job Duration	00:35	Prerequisite jobs	P1C-004
Default machine	M1A-010	Alternative machines	-
Job ID	P1C-006	Job start time	10:55
Job Duration	00:20	Prerequisite jobs	P1C-005
Default machine	M1A-004	Alternative machines	-
Job ID	P1C-007	Job start time	11:15
Job Duration	00:05	Prerequisite jobs	P1C-006
Default machine	M1A-003	Alternative machines	-
Job ID	P1C-008	Job start time	11:20
Job Duration	00:15	Prerequisite jobs	P1C-007
Default machine	M1A-006	Alternative machines	-
Job ID	P1C-009	Job start time	11:35
Job Duration	00:15	Prerequisite jobs	P1C-008
Default machine	M1A-001	Alternative machines	-
Job ID	P1C-010	Job start time	11:50
Job Duration	00:10	Prerequisite jobs	P1C-009
Default machine	M1A-005	Alternative machines	-
Process ID	D	Process release date	07:00
Process priority	1	Process Deadline	13:00
Job ID	P1D-001	Job start time	07:50
Job Duration	00:30	Prerequisite jobs	-
Default machine	M1A-001	Alternative machines	-
Job ID	P1D-002	Job start time	08:20
Job Duration	00:20	Prerequisite jobs	P1D -001
Default machine	M1A-002	Alternative machines	-
Job ID	P1D-003	Job start time	08:40
Job Duration	00:40	Prerequisite jobs	P1D-002
Default machine	M1A-009	Alternative machines	-
Job ID	P1D-004	Job start time	09:20
Job Duration	01:00	Prerequisite jobs	P1D-003
Default machine	M1A-010	Alternative machines	-
Job ID	P1D-005	Job start time	11:34
Job Duration	00:19	Prerequisite jobs	P1D-004
Default machine	M1A-007	Alternative machines	-

Job ID	P1D-006	Job start time	10:53
Job Duration	00:15	Prerequisite jobs	P1D-005
Default machine	M1A-004	Alternative machines	-
Job ID	P1D-007	Job start time	12:08
Job Duration	00:05	Prerequisite jobs	P1D-006
Default machine	M1A-001	Alternative machines	-
Job ID	P1D-008	Job start time	12:18
Job Duration	00:02	Prerequisite jobs	P1D-007
Default machine	M1A-008	Alternative machines	-
Job ID	P1D-009	Job start time	12:20
Job Duration	00:25	Prerequisite jobs	P1D-008
Default machine	M1A-010	Alternative machines	-
Job ID	P1D-010	Job start time	11:45
Job Duration	00:15	Prerequisite jobs	P1D-009
Default machine	M1A-008	Alternative machines	-

Test Schedule #6

Process ID	A	Process release date	09:00
Process priority	1	Process Deadline	13:00
Job ID	P1A-001	Job start time	11:05
Job Duration	00:10	Prerequisite jobs	-
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-002	Job start time	11:28
Job Duration	00:05	Prerequisite jobs	P1A-001
Default machine	M1A-001	Alternative machines	-
Job ID	P1A-003	Job start time	11:33
Job Duration	00:20	Prerequisite jobs	P1A-002
Default machine	M1A-010	Alternative machines	-
Job ID	P1A-004	Job start time	11:53
Job Duration	00:20	Prerequisite jobs	P1A-003
Default machine	M1A-006	Alternative machines	-
Job ID	P1A-005	Job start time	12:13
Job Duration	00:02	Prerequisite jobs	P1A-004
Default machine	M1A-001	Alternative machines	-
Job ID	P1A-006	Job start time	12:15
Job Duration	00:15	Prerequisite jobs	P1A-005
Default machine	M1A-003	Alternative machines	-
Job ID	P1A-007	Job start time	12:30
Job Duration	00:05	Prerequisite jobs	P1A-006
Default machine	M1A-007	Alternative machines	-

Job ID	P1A-008	Job start time	12:35
Job Duration	00:05	Prerequisite jobs	P1A-007
Default machine	M1A-008	Alternative machines	-
Job ID	P1A-009	Job start time	12:40
Job Duration	00:10	Prerequisite jobs	P1A-008
Default machine	M1A-009	Alternative machines	-
Job ID	P1A-010	Job start time	12:50
Job Duration	00:10	Prerequisite jobs	P1A-009
Default machine	M1A-001	Alternative machines	-
Process ID	B	Process release date	09:00
Process priority	1	Process Deadline	14:00
Job ID	P1B-001	Job start time	10:20
Job Duration	00:35	Prerequisite jobs	-
Default machine	M1A-004	Alternative machines	-
Job ID	P1B-002	Job start time	11:32
Job Duration	00:18	Prerequisite jobs	P1B-001
Default machine	M1A-005	Alternative machines	-
Job ID	P1B-003	Job start time	12:05
Job Duration	00:40	Prerequisite jobs	P1B-002
Default machine	M1A-002	Alternative machines	-
Job ID	P1B-004	Job start time	12:45
Job Duration	00:05	Prerequisite jobs	P1B-003
Default machine	M1A-010	Alternative machines	-
Job ID	P1B-005	Job start time	12:50
Job Duration	00:15	Prerequisite jobs	P1B-004
Default machine	M1A-007	Alternative machines	-
Job ID	P1B-006	Job start time	13:05
Job Duration	00:10	Prerequisite jobs	P1B-005
Default machine	M1A-009	Alternative machines	-
Job ID	P1B-007	Job start time	13:15
Job Duration	00:10	Prerequisite jobs	P1B-006
Default machine	M1A-006	Alternative machines	-
Job ID	P1B-008	Job start time	13:25
Job Duration	00:10	Prerequisite jobs	P1B-007
Default machine	M1A-004	Alternative machines	-
Job ID	P1B-009	Job start time	13:35
Job Duration	00:20	Prerequisite jobs	P1B-008
Default machine	M1A-005	Alternative machines	-
Job ID	P1B-010	Job start time	13:55
Job Duration	00:05	Prerequisite jobs	P1B-009
Default machine	M1A-002	Alternative machines	-

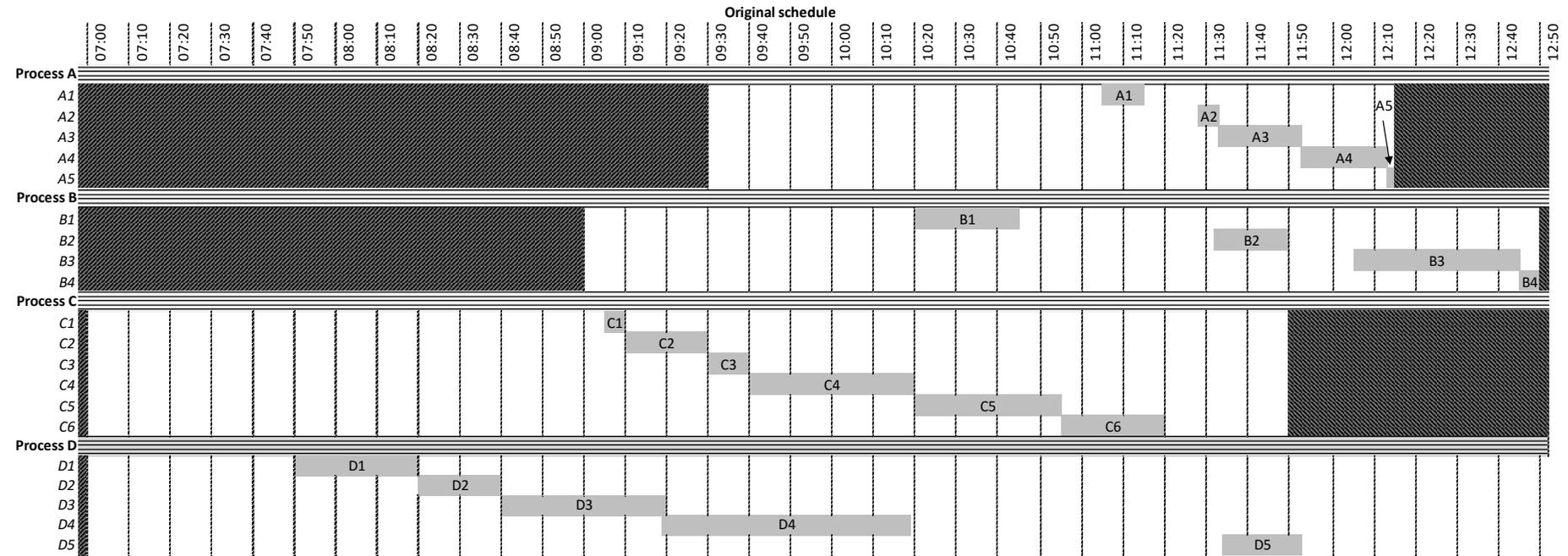
Process ID	C	Process release date	07:00
Process priority	1	Process Deadline	12:00
Job ID	P1C-001	Job start time	09:05
Job Duration	00:05	Prerequisite jobs	-
Default machine	M1A-008	Alternative machines	-
Job ID	P1C-002	Job start time	09:10
Job Duration	00:20	Prerequisite jobs	P1C-001
Default machine	M1A-002	Alternative machines	-
Job ID	P1C-003	Job start time	09:30
Job Duration	00:10	Prerequisite jobs	P1C-002
Default machine	M1A-009	Alternative machines	-
Job ID	P1C-004	Job start time	09:40
Job Duration	00:40	Prerequisite jobs	P1C-003
Default machine	M1A-007	Alternative machines	-
Job ID	P1C-005	Job start time	10:20
Job Duration	00:35	Prerequisite jobs	P1C-004
Default machine	M1A-010	Alternative machines	-
Job ID	P1C-006	Job start time	10:55
Job Duration	00:20	Prerequisite jobs	P1C-005
Default machine	M1A-004	Alternative machines	-
Job ID	P1C-007	Job start time	11:15
Job Duration	00:05	Prerequisite jobs	P1C-006
Default machine	M1A-003	Alternative machines	-
Job ID	P1C-008	Job start time	11:20
Job Duration	00:15	Prerequisite jobs	P1C-007
Default machine	M1A-006	Alternative machines	-
Job ID	P1C-009	Job start time	11:35
Job Duration	00:15	Prerequisite jobs	P1C-008
Default machine	M1A-001	Alternative machines	-
Job ID	P1C-010	Job start time	11:50
Job Duration	00:10	Prerequisite jobs	P1C-009
Default machine	M1A-005	Alternative machines	-
Process ID	D	Process release date	07:00
Process priority	1	Process Deadline	13:00
Job ID	P1D-001	Job start time	07:50
Job Duration	00:30	Prerequisite jobs	-
Default machine	M1A-001	Alternative machines	-
Job ID	P1D-002	Job start time	08:20
Job Duration	00:20	Prerequisite jobs	P1D -001
Default machine	M1A-002	Alternative machines	-

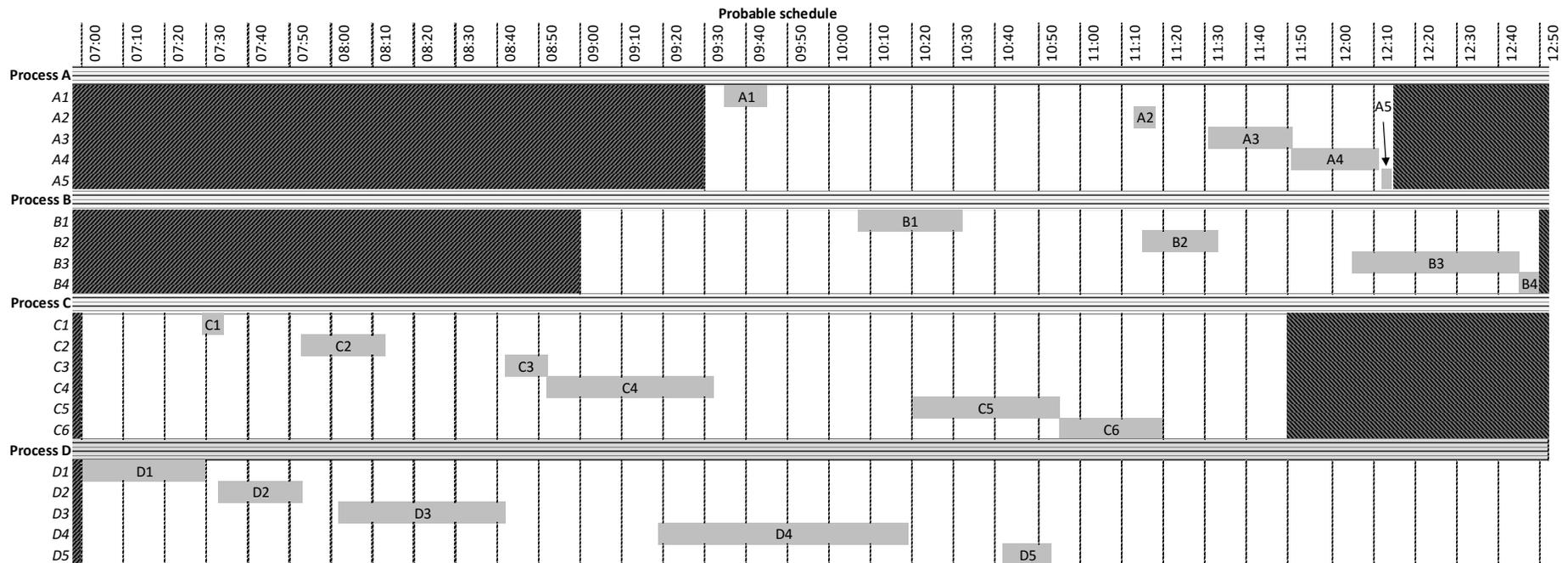
Job ID	P1D-003	Job start time	08:40
Job Duration	00:40	Prerequisite jobs	P1D-002
Default machine	M1A-009	Alternative machines	-
Job ID	P1D-004	Job start time	09:20
Job Duration	01:00	Prerequisite jobs	P1D-003
Default machine	M1A-010	Alternative machines	-
Job ID	P1D-005	Job start time	11:34
Job Duration	00:19	Prerequisite jobs	P1D-004
Default machine	M1A-007	Alternative machines	-
Job ID	P1D-006	Job start time	10:53
Job Duration	00:15	Prerequisite jobs	P1D-005
Default machine	M1A-004	Alternative machines	-
Job ID	P1D-007	Job start time	12:08
Job Duration	00:05	Prerequisite jobs	P1D-006
Default machine	M1A-001	Alternative machines	-
Job ID	P1D-008	Job start time	12:18
Job Duration	00:02	Prerequisite jobs	P1D-007
Default machine	M1A-008	Alternative machines	-
Job ID	P1D-009	Job start time	12:20
Job Duration	00:25	Prerequisite jobs	P1D-008
Default machine	M1A-010	Alternative machines	-
Job ID	P1D-010	Job start time	11:45
Job Duration	00:15	Prerequisite jobs	P1D-009
Default machine	M1A-008	Alternative machines	-
Process ID	E	Process release date	10:00
Process priority	1	Process Deadline	15:00
Job ID	P1E-001	Job start time	11:14
Job Duration	00:18	Prerequisite jobs	-
Default machine	M1A-005	Alternative machines	-
Job ID	P1E-002	Job start time	11:40
Job Duration	00:05	Prerequisite jobs	P1E -001
Default machine	M1A-006	Alternative machines	-
Job ID	P1E-003	Job start time	11:45
Job Duration	00:30	Prerequisite jobs	P1E-002
Default machine	M1A-003	Alternative machines	-
Job ID	P1E-004	Job start time	12:40
Job Duration	00:30	Prerequisite jobs	P1E-003
Default machine	M1A-004	Alternative machines	-
Job ID	P1E-005	Job start time	13:10
Job Duration	00:05	Prerequisite jobs	P1E-004
Default machine	M1A-005	Alternative machines	-

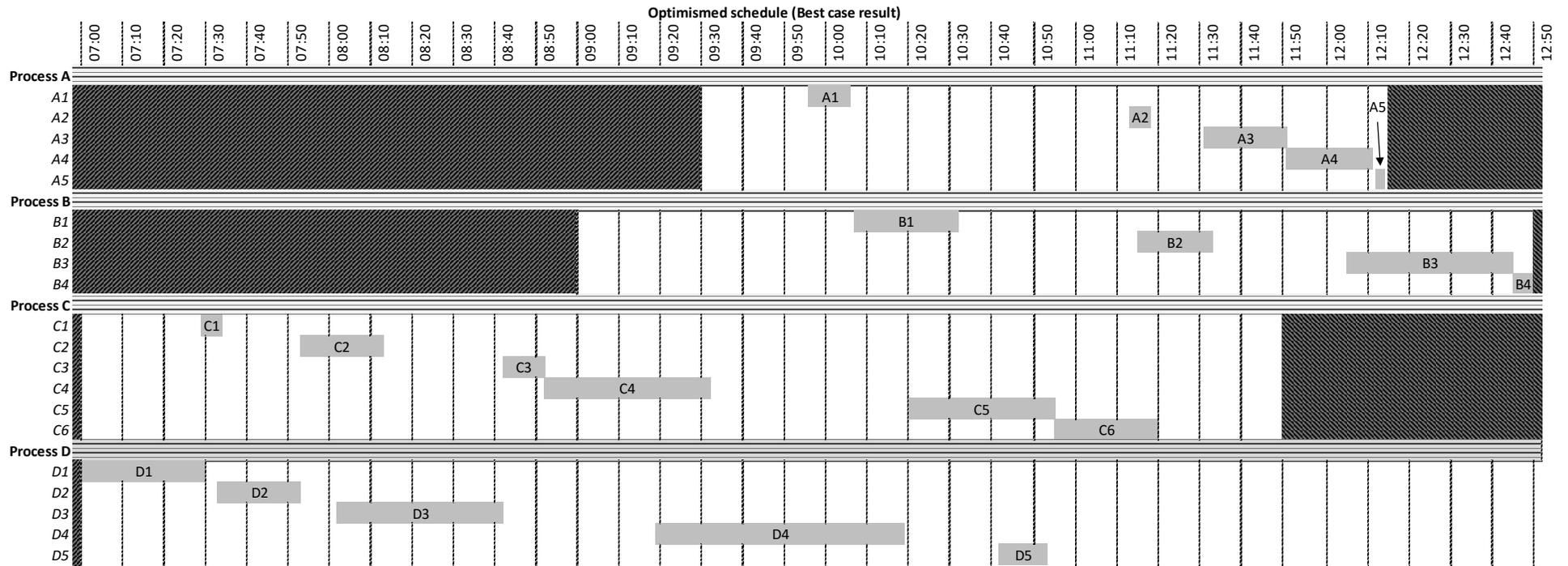
Job ID	P1E-006	Job start time	13:15
Job Duration	00:20	Prerequisite jobs	P1E-005
Default machine	M1A-003	Alternative machines	-
Job ID	P1E-007	Job start time	13:35
Job Duration	00:20	Prerequisite jobs	P1E-006
Default machine	M1A-002	Alternative machines	-
Job ID	P1E-008	Job start time	14:05
Job Duration	00:20	Prerequisite jobs	P1E-007
Default machine	M1A-008	Alternative machines	-
Job ID	P1E-009	Job start time	14:25
Job Duration	00:05	Prerequisite jobs	P1E-008
Default machine	M1A-009	Alternative machines	-
Job ID	P1E-010	Job start time	14:30
Job Duration	00:30	Prerequisite jobs	P1E-009
Default machine	M1A-006	Alternative machines	-

Appendix F: Schedule Gantt Charts

Test schedule #3 (20 job schedule)

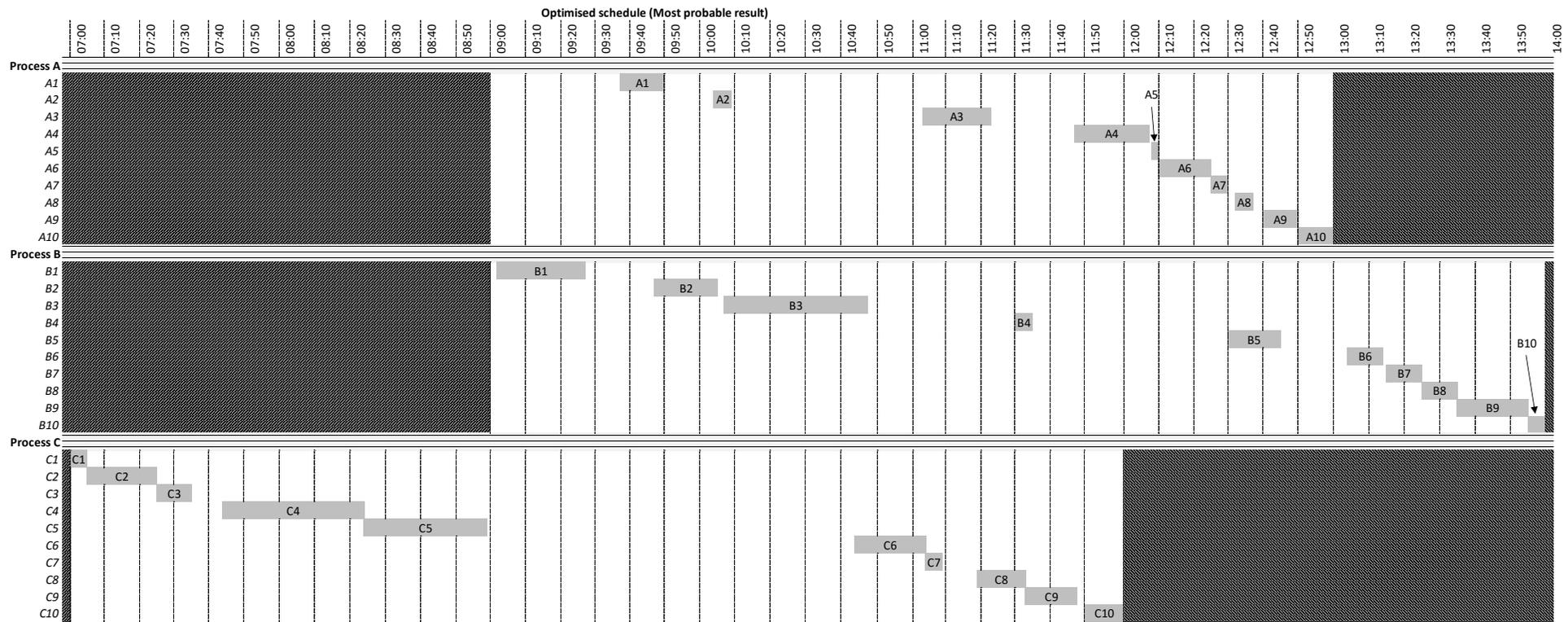


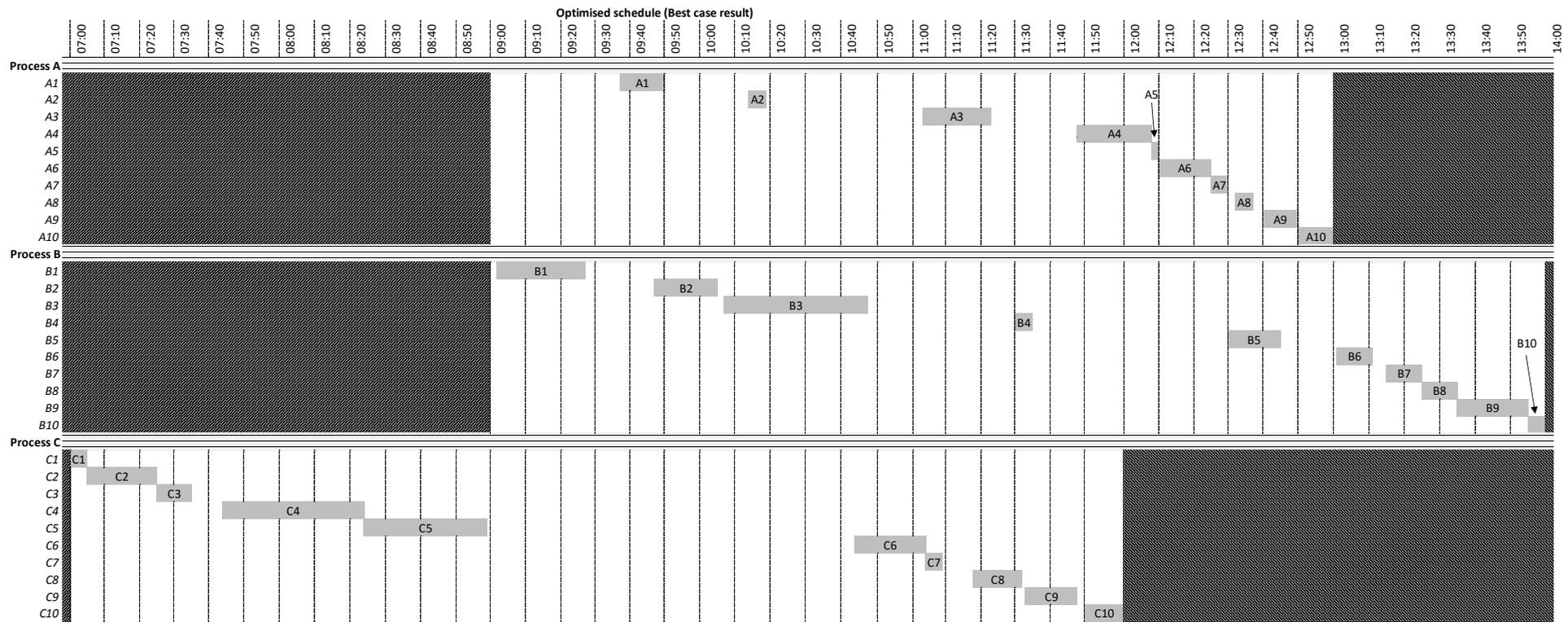




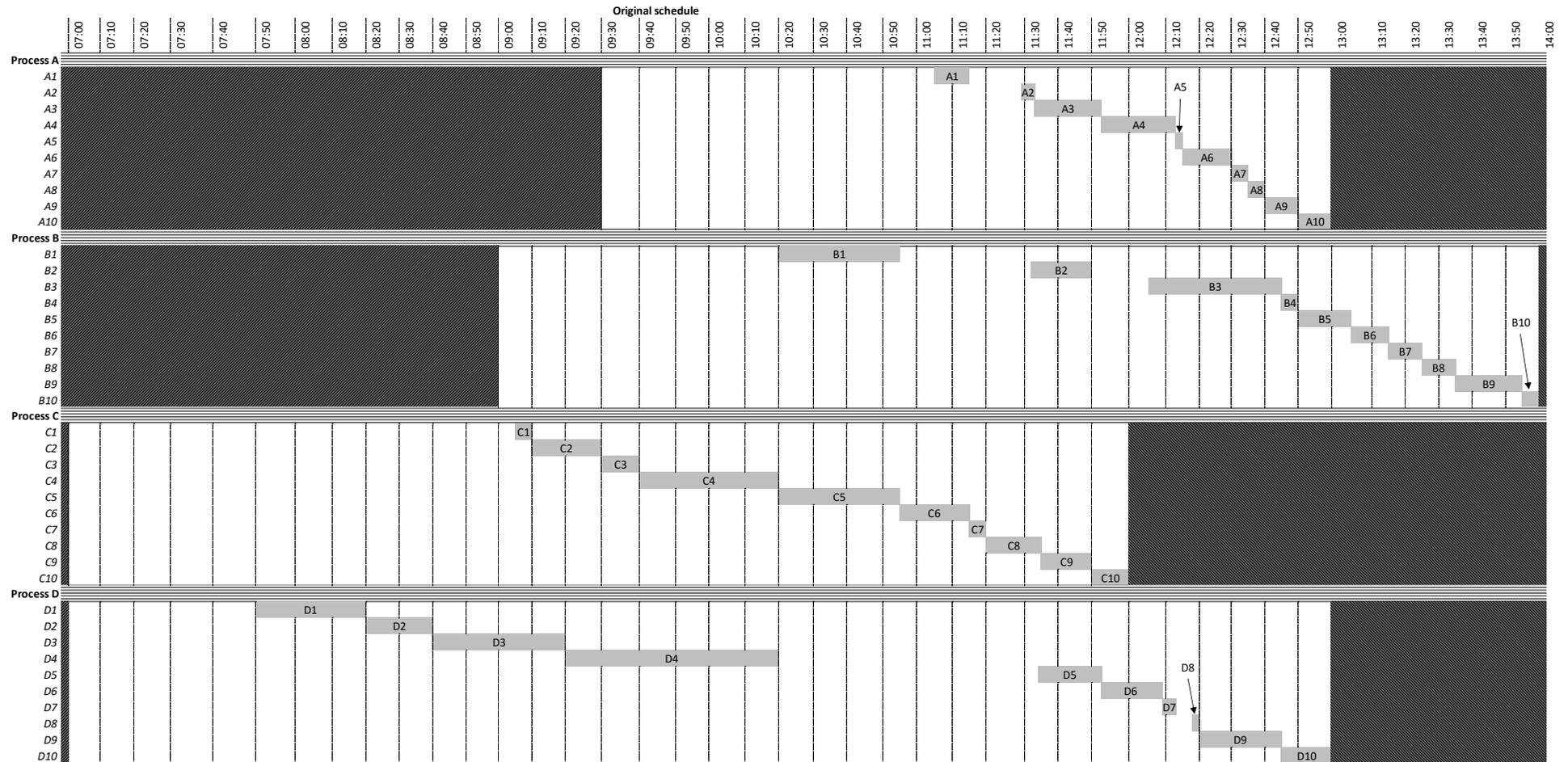
Test schedule #4 (30 job schedule)

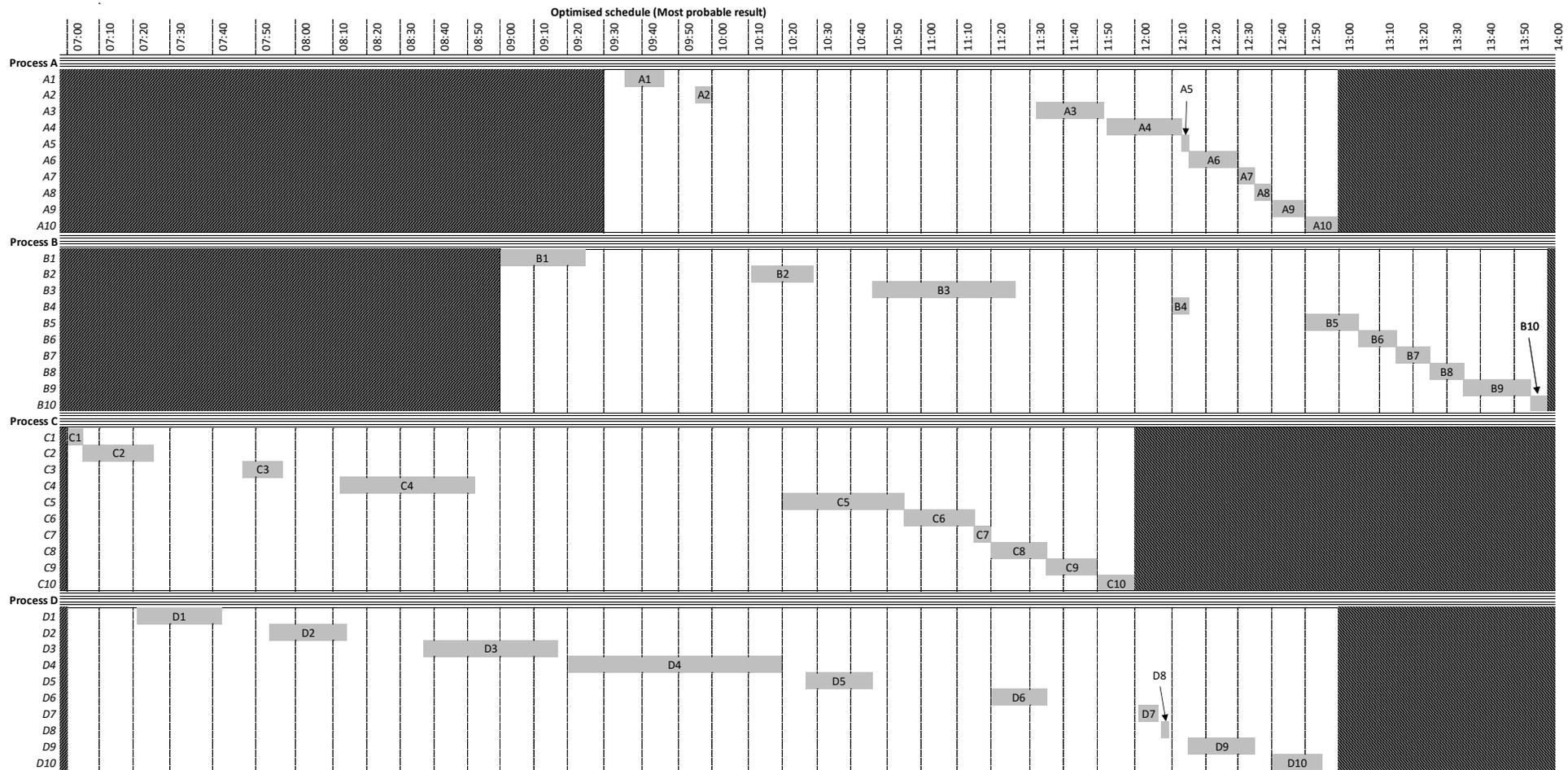


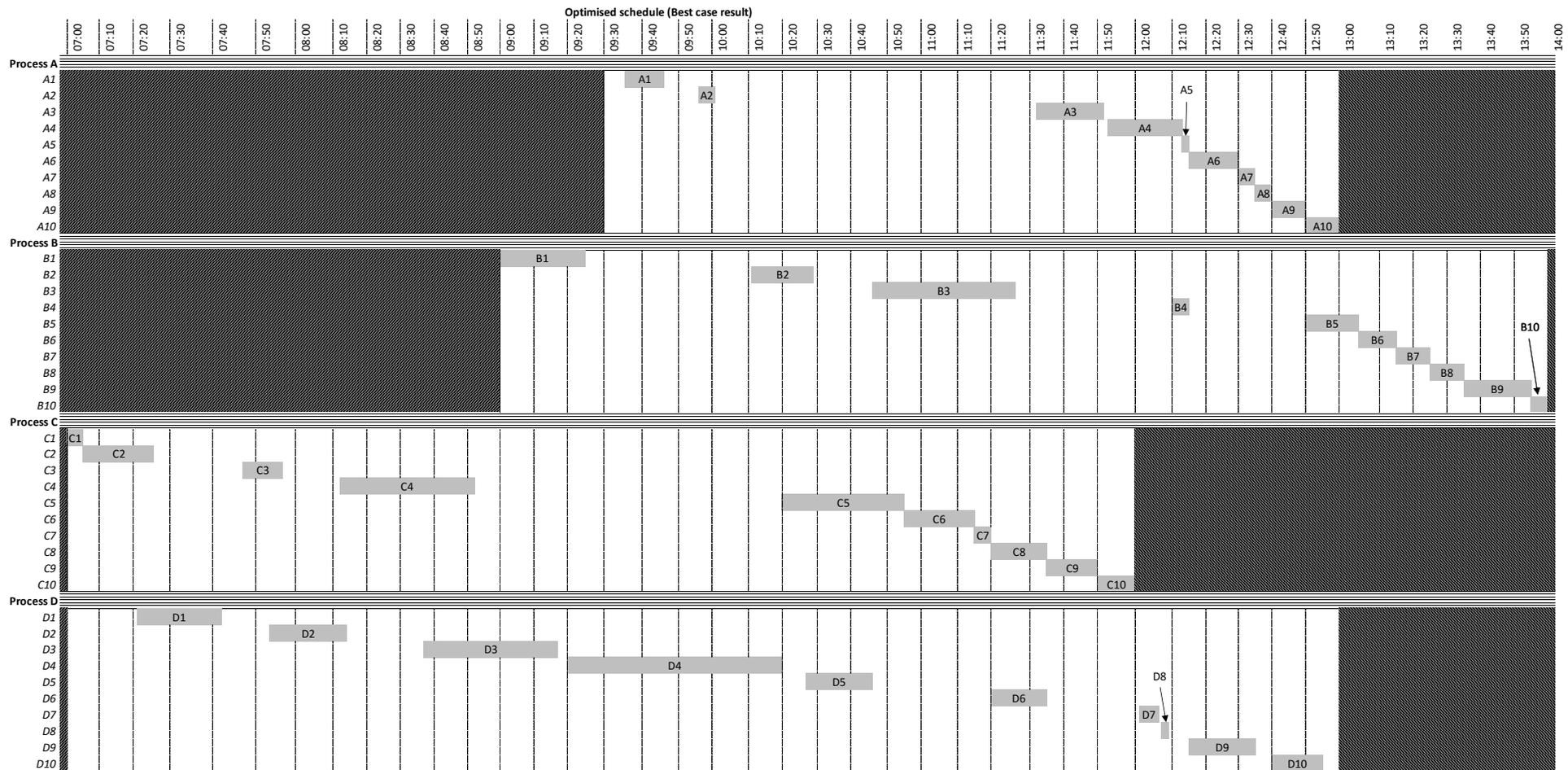




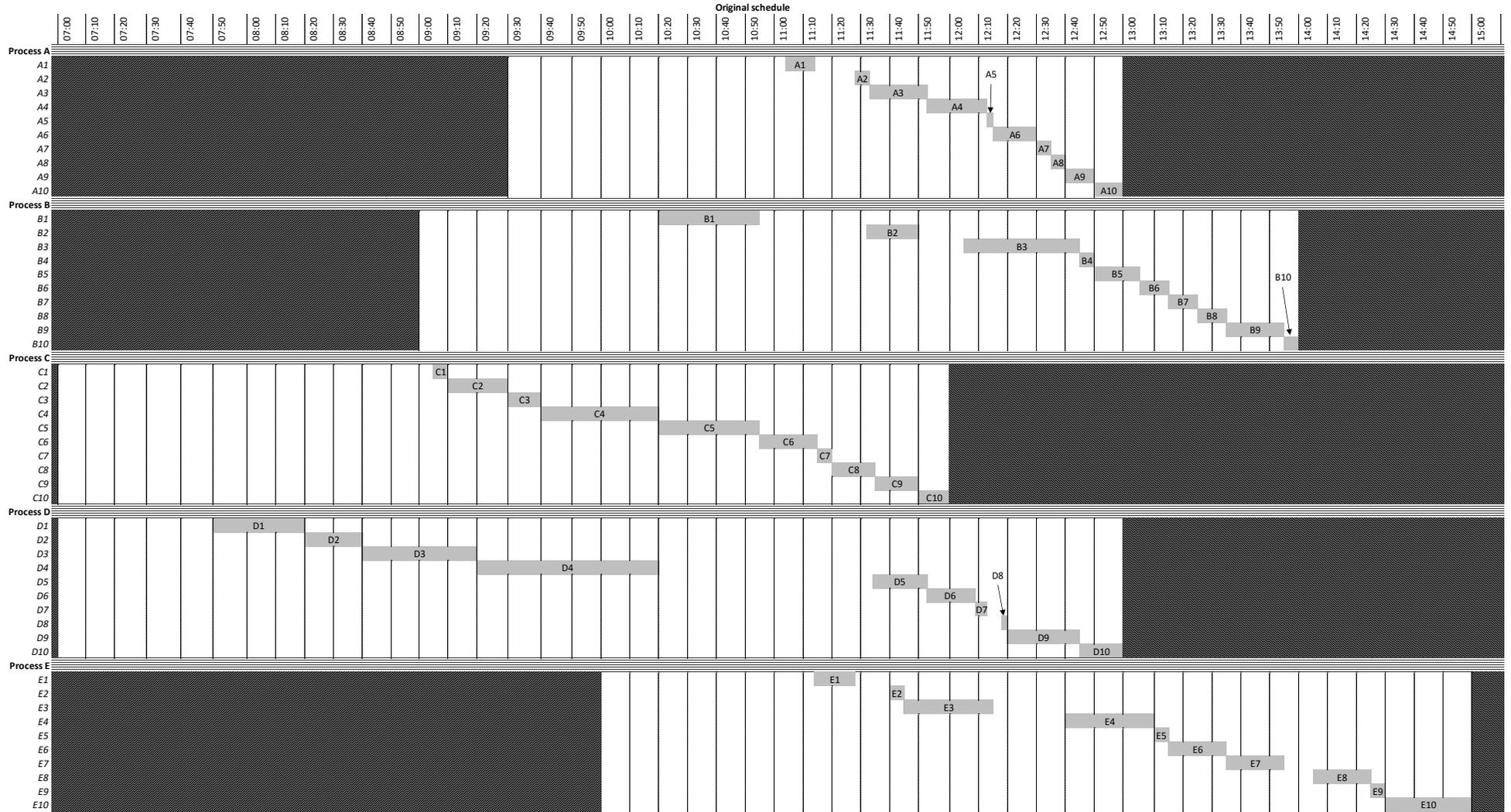
Test schedule #5 (40 job schedule)

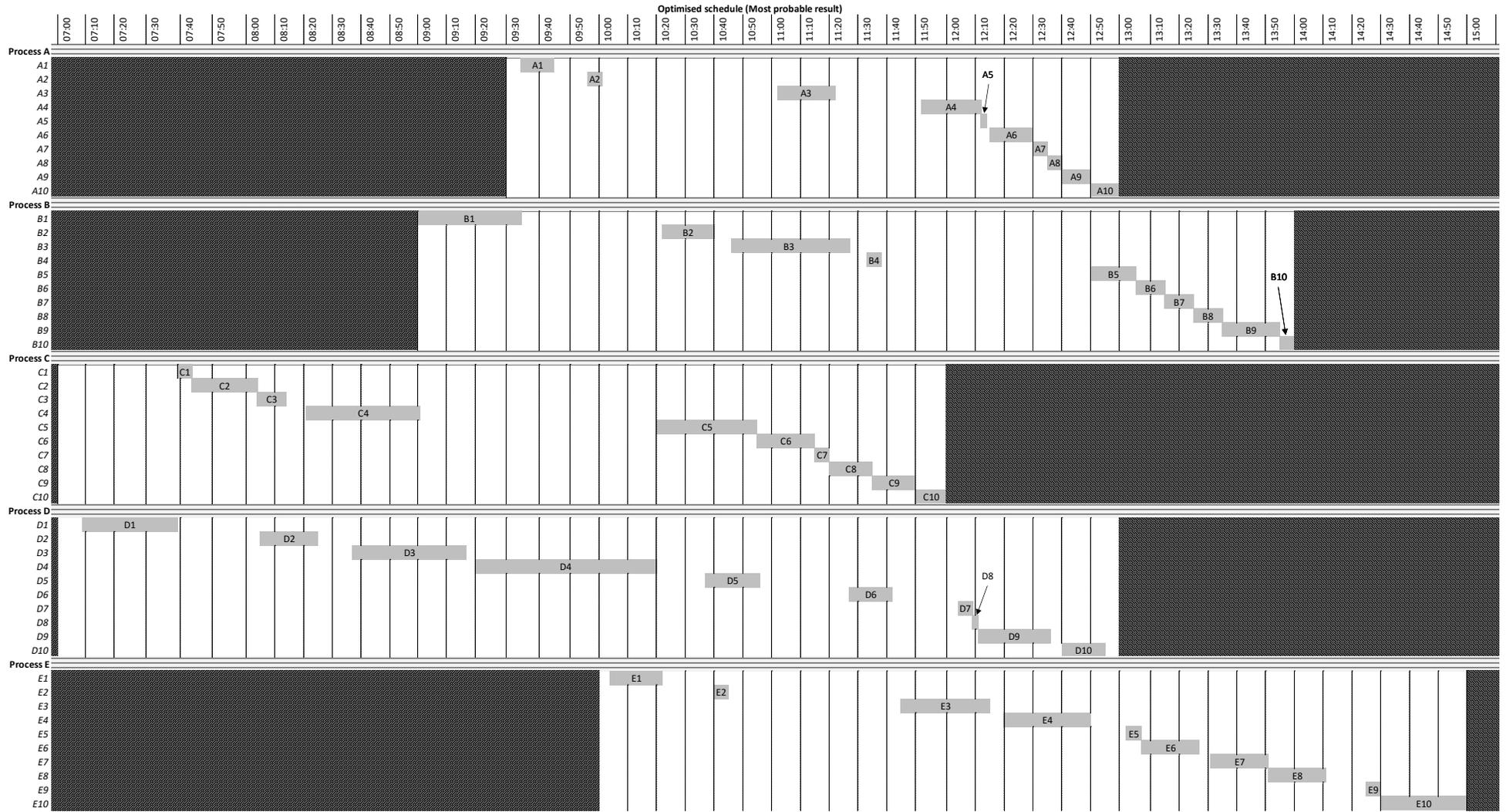


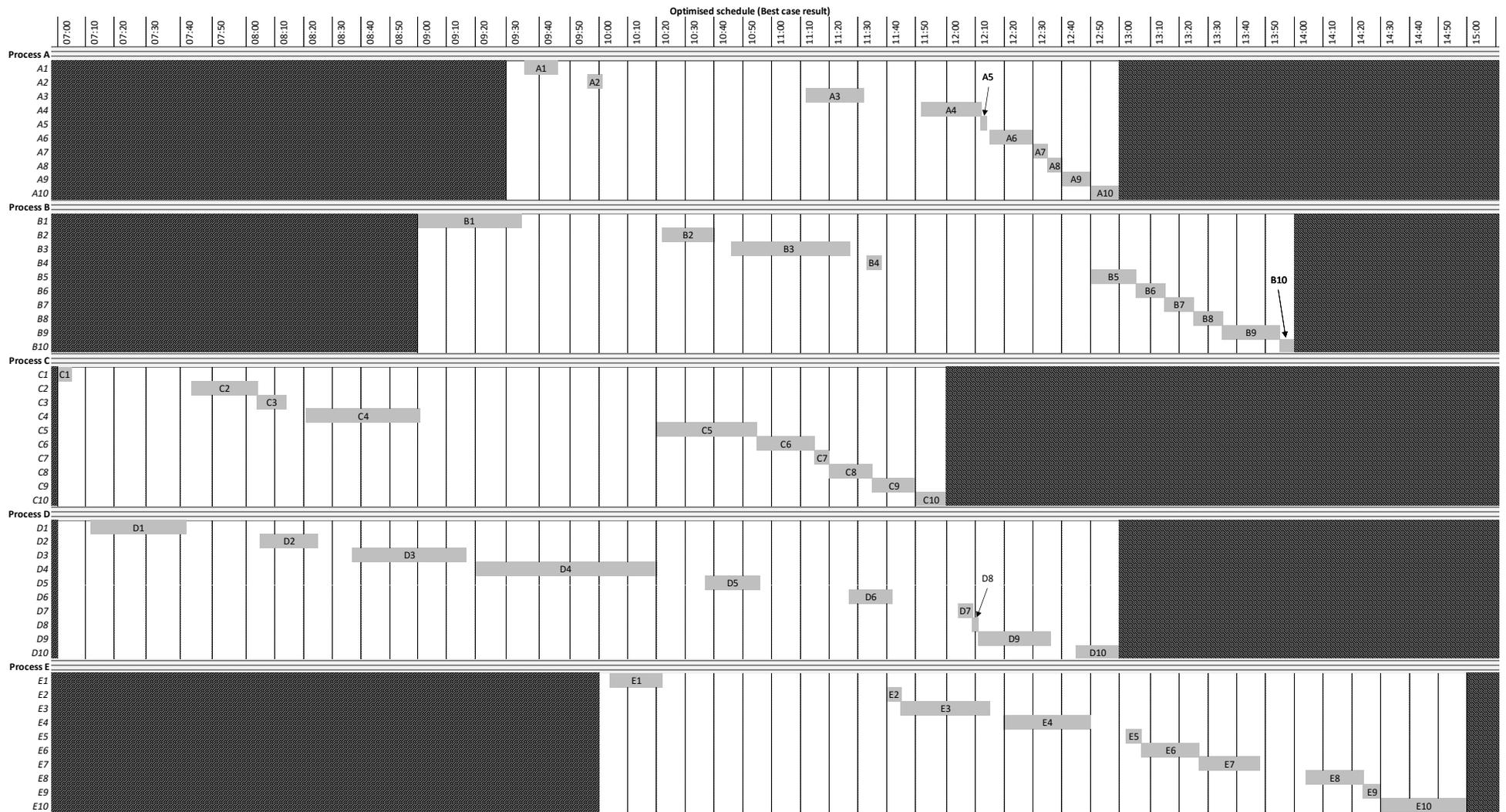




Test schedule #6 (50 job schedule)







Appendix G: Test Implementation Schedule

Process ID	A	Process release date	13:04
Process priority	1	Process Deadline	13:23
Job ID	P1A-001	Job start time	13:12
Job Duration	00:03	Prerequisite jobs	-
Default machine	Fan heater	Alternative machines	-
Job ID	P1A-002	Job start time	13:15
Job Duration	00:05	Prerequisite jobs	P1A-001
Default machine	Compressor	Alternative machines	-
Job ID	P1A-003	Job start time	13:20
Job Duration	00:03	Prerequisite jobs	P1A-002
Default machine	Vacuum	Alternative machines	-
Process ID	B	Process release date	12:54
Process priority	1	Process Deadline	13:43
Job ID	P1B-001	Job start time	13:15
Job Duration	00:10	Prerequisite jobs	-
Default machine	AC motor	Alternative machines	-
Job ID	P1B-002	Job start time	13:25
Job Duration	00:10	Prerequisite jobs	P1B-001
Default machine	AC motor	Alternative machines	-
Job ID	P1B-003	Job start time	13:35
Job Duration	00:50	Prerequisite jobs	P1B-002
Default machine	Compressor	Alternative machines	-
Job ID	P1B-004	Job start time	13:40
Job Duration	00:03	Prerequisite jobs	P1B-003
Default machine	Fan heater	Alternative machines	-
Process ID	C	Process release date	11:00
Process priority	1	Process Deadline	13:00
Job ID	P1C-001	Job start time	13:02
Job Duration	00:03	Prerequisite jobs	-
Default machine	Fan heater	Alternative machines	-
Job ID	P1C-002	Job start time	13:05
Job Duration	00:10	Prerequisite jobs	P1C-001
Default machine	AC motor	Alternative machines	-
Job ID	P1C-003	Job start time	13:28
Job Duration	00:05	Prerequisite jobs	P1C-002
Default machine	Compressor	Alternative machines	-

