Barriers to the implementation of Flexible Demand services within the GB electricity generation and supply system

by

Graeme John Hodgson

A Doctoral Thesis

Submitted in partial fulfillment of the requirements for the award of
Doctor of Engineering of Loughborough University

13 September 2013

Dedication

To Vicki. Thank you.

Abstract

The implementation of a low carbon electricity system within the GB requires a significant change to the generation mix with an increasing role for renewable generation. Much of this generation will be intermittent. To date system balancing has largely relied on predicting demand and ensuring provision. With substantial intermittency, continuation of this paradigm necessitates significant investment in peaking plant and/or storage. However, some of this investment can be avoided by harnessing the flexibility inherent in many electrical loads. Despite the attractiveness of such services, we do not see their large-scale implementation. The aim of this thesis is to consider why. A historical analysis reveals that both nationalisation and subsequent privatisation provide precedents for significant structural change as the integration of large-scale flexible demand might require. The need for political will is identified as a crucial enabling factor. Without an ideological driver, however, a perception of economic and/or technological risk can preclude the implementation of supportive policy. This perception is addressed through demonstration. An effective demonstration must show the ability to aggregate many small loads in a coordinated manner. A genetic algorithm that provides this core dispatch and optimisation capability is presented. This algorithm is shown to be effective in aggregating many small loads to provide a net effect that can be used as a balancing service and to do so in an optimal way considering both cost and reliability. Having demonstrated feasibility appropriate incentives must be created. An initial outline for a framework based on SysML is presented that can be used to identify where structural barriers to implementation are present to aid the design of appropriate policy incentives.

Keywords: demand response; flexible demand; low carbon electricity; scheduling; combinatorial optimisation; genetic algorithms; enterprise relationship modelling; SysML.

Acknowledgements

A number of individuals have been a tremendous source of knowledge, inspiration and support during this research. At E.ON New Build and Technology Limited I would like to thank Dr Chris Horne, Alice Delahunty, Sarah Curtis, Dr Conor Clifford and Rachel Stanley. At Loughborough University I would like to thank my supervisor Dr Murray Thomson who has provided excellent support and guidance throughout this work. There are many others that I have met, or worked with, along the way that have been of great help and to whom I owe my thanks.

This research was conducted through the Systems Engineering Doctorate Centre at Loughborough University and was co-funded by E.ON New Build and Technology Limited and the Engineering and Physical Sciences Research Council, UK.

Table of Contents

Chapter 1 Introduction 12
1.1 Description of the Problem 12
1.2 Approach to the Problem 19
Chapter 2 Literature Review 21
2.1 Introduction 21
2.2 Current UK Energy Consumption 22
2.3 Future UK energy policy 25
2.4 GB Electricity System Architecture 28
2.5 Electricity Generation 31
2.6 System Operation 36
2.7 Balancing Services 40
2.8 Characteristics of FD Services 43
2.8.1 Demand Side Management and Flexible Demand 43
2.8.2 Types of FD Service 47
2.8.3 Market Opportunities for FD 51
2.9 Summary and Conclusions 54
Chapter 3 A Historical Analysis of the GB Electricity Industry 55
3.1 Introduction 55
3.2 Evolution of the GB Electricity Industry in the Post-war Era 55
3.3 A Political Barrier to FD Services 66
3.4 Summary and Conclusions 70
Chapter 4 A System for the Demonstration of FD Services 72
4.1 Introduction 72
4.2 FD Demonstration System Architecture 73
4.3 Design of the Demonstration System 74
4.4 Scheduling and Dispatch 80

4.5 Scheduling and Dispatch Requirements 81
4.6 Summary and Conclusions 85
Chapter 5 An Algorithm for the Dispatch of Loads in a FD System 87
5.1 A Review of Literature on Optimised Scheduling 87
5.1.1 Scheduling Approaches within Demand Response 87
5.1.2 Types of Optimisation Algorithm 90
5.1.3 Parameterisation of Nodes 92
5.2 Algorithm Selection 94
5.3 Problem Complexity 95
5.4 Metaheuristic Algorithms 98
5.5 The Fitness Function 102
5.5.1 Power 104
5.5.2 Economic Cost 104
5.5.3 Reliability 105
5.5.4 The Complete Fitness Function 106
5.6 Description of the Genetic Algorithm 107
5.7 Summary and Conclusions 122
Chapter 6 Analysis of the Results of Testing of the Dispatch Algorithm 123
6.1 Implementation 123
6.2 Testing, Tuning and Performance Analysis 125
6.2.1 Creation of the Test Data 125
6.2.2 Test Descriptions 130
6.2.3 Analysis of the Test Results 133
6.3 Summary and Conclusions 153
Chapter 7 A Methodology for Modelling Structural Barriers Within Industrial Systems using SysML 155
7.1 Introduction 155

7.2 The Requirements of an Analytical Framework ... 156
7.3 A Review of Literature on Modelling Stakeholder Relationships ... 157
7.4 Barriers as an Emergent Property of Relationships ... 163
7.5 Using SysML as an Enterprise Analysis Framework to Model Relationships within the DD-FD Project ... 166
7.5.1 DD-FD Structure ... 170
7.5.2 DD-FD Behaviour ... 172
7.5.3 Modelling Relationships ... 174
7.6 Summary and Conclusions ... 181
Chapter 8 Conclusions ... 182
8.1 Research Objective ... 182
8.2 Research Conclusions ... 182
8.3 Claims of Originality ... 187
8.4 Suggestions for Further Work ... 188
8.4.1 Optimised Dispatch ... 188
8.4.2 Structural Modelling using SysML ... 189
Bibliography ... 191
Appendix A: Published Papers ... 204
Appendix B: Example Source Code ... 205

List of Figures

Figure 1: UK final energy consumption and electricity generation by fuel type for 2012. 23
Figure 2: GB electricity network architecture. 29
Figure 3: example sources of energy used in electrical power schemes. 32
Figure 4: installed generation capacity by generation category (DECC, 2013). 34
Figure 5: average power station size by generation category (DECC, 2013). 34
Figure 6: example frequency trace over a 60-minute period for the GB system (National Grid, 2011a). 37
Figure 7: the "predict and provide" operating paradigm. 38
Figure 8: example variation in demand over a typical winter day (source: National Grid). 39
Figure 9: the service types comprising Demand Side Management. 45
Figure 10: energy and flexibility value chains. 53
Figure 11: timeline showing key legislative developments within the UK electricity system. 56
Figure 12: the political-industrial cycle of inaction. 69
Figure 13: research outcomes. 71
Figure 14: FD demonstration system architecture. 74
Figure 15: viewpoint analysis of DD-FD system requirements. 75
Figure 16: alternatives for FD control strategies. 82
Figure 17: simple rectangular consumption curve. 85
Figure 18: types of optimisation algorithm. 91
Figure 19: crossover operation. 115
Figure 20: mutation operation. 116
Figure 21: flowchart for GA (1 of 4). 118
Figure 22: flowchart for GA (2 of 4). 119

Figure 23: flowchart for GA (3 of 4). 120
Figure 24: flowchart for GA (4 of 4). 121
Figure 25: distribution of test node power ratings for TD2. 128
Figure 26: the best fitness by generation for each of the 10 runs in Test 1. 136
Figure 27: the rate of change in the median fitness value for each data set used in Test 3. 137
Figure 28: fitness improvement by generation for Run 1 of Test 1. 139
Figure 29: count of the number of times each of the known top 20 solutions was found by the GA in Test 2. 140
Figure 30: frequency of use of individual offers within the best 10 solutions in Test 1. 141
Figure 31: count of the number of offers used in each of the best solutions in Test 1. 142
Figure 32: median fitness by generation for each data set in Test 3. 144
Figure 33: relationship between the required number of generations to achieve an acceptable fitness and the offer pool size derived from the results of Test 3. 145
Figure 34: number of solutions considered in each run in Test 1 (median shown as a solid black line). 146
Figure 35: results of performance testing with the TD2, 250-offer data set. 149
Figure 36: results of performance testing with the TD2, 500-offer data set. 151
Figure 37: execution times for the performance tests. 152
Figure 38: stakeholder map for the DD-FD project. 168
Figure 39: SysML diagram types (Source: Friedenthal et al., 2008). 170
Figure 40: SysML block definition diagram for the DD-FD system. 171
Figure 41: main use cases within the DD-FD system. 173
Figure 42: internal block diagram for the demo site block. 175
Figure 43: SysML block definition diagram for the value chain. 177

Figure 44: internal block diagram for a commercial relationship. 178
Figure 45: adding constraints as constraint notes to a SysML diagram. 180
Figure 46: example complex consumption curve. 188

List of Tables

Table 1: key 2012 demand statistics for the GB electricity system (DECC, 2013).........24

Table 2: emissions reduction targets as set out in the Climate Change Act 2008 and a 2009 Statutory Instrument (UK Government, 2009a; UK Government, 2009b). 26

Table 3: key UK generation statistics for 2012 (DECC, 2013). 35

Table 4: balancing services procured by National Grid. 41

Table 5: types of DR programme [after Albadi and El-Saadany (2008)]. 49

Table 6: system level non-functional requirements for the FD demonstrator. 76

Table 7: system level functional requirements for the FD demonstrator. 77

Table 8: summary of design choices for the FD system. 79

Table 9: scheduler requirements. 84

Table 10: numbers of possible combinations with different numbers of controllable nodes. 97

Table 11: structure of a chromosome. 108

Table 12: test data characteristics. 126

Table 13: number of solutions for each valid number of selected offers for the TD1 data set. 129

Table 14: pool characteristics for the test data sets. 130

Table 15: fittest 20 solutions for the TD1 data set. 134

Table 16: solutions from with modified scalars. 135

Table 17: execution times for Test 3. 147

Table 18: performance testing configurations. 148

Table 19: performance testing configurations with the TD2, 500-offer data set........150

Table 20: comparison chart for alternative modelling frameworks. 158

Table 21: interfaces used in the `makeOfferToSO` use case. 175

Chapter 1

Introduction

1.1 Description of the Problem

Concerns over anthropogenic global warming, leading to potentially irreversible and devastating climate change, are stimulating efforts to reduce global emissions of greenhouse gases (GHGs) into the Earth's atmosphere. One of the principal emitters is the global energy industry through its consumption and combustion of fossil fuels (IPCC, 2007). In response many governments are actively pursuing policies aimed at reducing their respective countries' reliance on fossil fuels. A key mechanism for achieving these policy goals is the increased use of renewable energy both directly (e.g. biomass-fired heating schemes) and indirectly via intermediate conversion into electricity. This expanded use of renewable energy sources for electricity production, many of which are intermittent, has implications for the operation of the electricity system used within Great Britain (GB). These implications are the starting point for the research described within this thesis.

The nature of electrical power is such that supply and demand must be constantly balanced. In extreme cases a mismatch in supply and demand can lead to system outages. Given the dependence of modern societies on electricity as a source of energy such outages can have widespread and catastrophic effects, such as was the case in India in 2012 (BBC News, 2012). India places particular stress on its electricity system, as it is a country undergoing rapid economic development. In contrast most developed countries, such as Britain, are experiencing more modest growth and such outages are thankfully extremely rare. In Britain society has become accustomed to, and expects, extremely high reliability from its electricity system. This high reliability is principally due to the flexibility of conventional fossil-fuelled generation. Generators vary their output autonomously in real-time in sympathy with fluctuations in supply and demand whilst fluctuations over longer timescales (minutes to days, weeks and even years) are handled through forecasting the expected demand and scheduling the availability of generation. This is possible because conventional generation has an inherent energy store in the form of the fuel fed into the power station (e.g. coal, gas, uranium) or in the water held behind a dam. This operational model is known as "predict and provide".

The expanded use of renewable energy presents a challenge to this model. Wind is seen as the principal means by which additional renewable energy generators will be added to the GB system (National Grid, 2009b). However, wind is an intermittent renewable energy source: the output of a wind turbine is limited by the instantaneous wind speed. Wind, therefore, has considerably reduced flexibility compared to conventional generation. This has little overall effect on system balancing when the proportion of intermittent renewable generation within the overall generation mix is low, but in the future where such generators constitute a significant proportion of generation this will no longer be the case. The availability of generation to meet all demand fluctuations can no longer be relied upon. In this case the "predict and provide" model is no longer viable.

There are three potential solutions to this problem that can be implemented either alone or in combination:

(i) Increase the generation reserve either by retaining conventional plant that would otherwise be retired, or through additional plant.

(ii) Add energy storage to the system, i.e. plant that converts electrical energy at times of surplus into a form that can be stored over the required timescale, and then releases it back into the system by converting it back into electricity as needed.

(iii) Exploit demand flexibility by managing demand to match the available supply, i.e. balance the system by actively modifying the power demand.

Definition

In this thesis energy storage is defined as that provided in the supply side of the market. Storage at or close to the point of use (i.e. distributed storage) is provided in the demand side of the market and is therefore considered to be a type of flexible demand. This is in accordance with the International Energy Agency (IEA) definition of demand side management, of which the demand flexibility considered here is a sub-type:

Demand Side Management refers to all changes that originates [sic] from the demand side of the market in order to achieve large-scale energy efficiency improvements by deployment of improved technologies. (IEA, 2012)

For all these options it is important to consider both the technical and economic implications. Technical implications are considered first. Option (i) constitutes business as usual and presents no new challenges. Many different energy storage schemes suitable for implementing option (ii) have been proposed but the majority remain largely unproven, relying on significant technical advances to be viable (Frost & Sullivan, 2012a). To date only pumped hydro schemes have achieved any level of viability comprising some 99% of existing schemes worldwide (The Economist, 2012b), but suitable locations for new schemes are scarce. Energy storage schemes do, however, have the advantage of fitting into the existing "predict and provide" model.

Option (iii) breaks the "predict and provide" model and requires the design and implementation of a new approach to operating the system. It requires the ability to modify the consumption of connected loads as needed, in a manner similar to the way in which generator output can be modified, such that it can play an active part in system balancing. This is often referred to as Demand Response (DR). The ability to control consumption is an inherent feature of all energy-consuming devices since ultimately they can all be turned on or off, connected or disconnected, turned up or down by the end user. The technical challenge is in coordinating that action to achieve a desired aggregate effect and then incorporating it into system operation while continuing to provide a reliable service to consumers.

Considering economic implications, option (i) will be expensive. The annual load factor of existing conventional plant will fall. Additionally, any new conventional plant will have a lower annual load factor than it would have had if it had been added to a system with little intermittent generation, like that of today. Idle capacity is expensive and so a rise in end-user energy costs is inevitable with this option. As an example the recent 2 GW Pembroke CCGT plant, a typical flexible gas power station, has a reported build cost of around £1 billion (Gas to Power Journal, 2012c).

Definition

The *annual load factor* is the ratio of the actual energy output of a generating unit to the maximum possible energy output of that generating unit over a full year (i.e. assuming the generator produces the nameplate power for the entire year without any outages). It is often expressed in percentage terms.

Option (ii) will also be expensive. Even if suitable locations can be found the capital costs for pumped hydro schemes are high. For example the 1.7 GW Dinorwig scheme was completed in 1982 and took 10 years to build at a cost of £425 million (First Hydro Company, 2012). The equivalent cost today would be more than

£1.2 billion. Other storage schemes have unpredictable costs due to their technical immaturity but even when they reach maturity there is no reason to assume that the build cost of a storage plant will be less than that of a similarly sized conventional power station. Storage will therefore also significantly increase energy costs and the lack of suitable sites for pumped hydro means any investor in storage must be subject to significant risk associated with unproven technology.

Option (iii) also involves cost. It requires the construction of a communications infrastructure and IT systems to manage the dispatch of load. The cost of the loads themselves is borne by the consumer that purchased them to provide their main function and is therefore not a factor. However, there is a need for additional functionality to support DR and that must form part of the implementation cost. One of the main costs is the communications infrastructure. This cost is however mitigated by the migration to a connected society with communications infrastructure being constructed already for other purposes; e.g. mobile data networks, broadband and smart metering. Any (or a combination) of these networks could be used to support DR. Smart metering does not of itself provide DR but it does provide a communications infrastructure. The UK Government has already committed to the rollout of smart meters to all homes and small businesses in the UK beginning in 2014 and expected to be complete by 2019. The estimated cost is £8.6 billion (DECC, 2011). The conclusion is that starting in 2014 a communications network capable of supporting DR will gradually be implemented at minimal cost to those DR services as the majority of the operational and capital costs will be borne elsewhere. The other major cost component is the IT systems required to support operation of the DR services. There is no reason to assume that these IT systems will cost any more than IT systems of similar size and complexity. This is likely to be the order of millions rather than billions of pounds. Overall, then, the cost of building the infrastructure required to utilise demand flexibility is very likely to be considerably less than even the cost of a single conventional power plant.

All of these options have advantages and disadvantages: there is no perfect answer to the problem of balancing a system with significant intermittent renewables. In reality it is likely that what is required is to find the best mix of all the available options. Although to date a great deal of political focus has been on making an attractive investment case for new generation (both centralised and distributed) and on supporting nascent storage technologies, there are now a number of enabling factors that make a powerful case for substantial investment in DR:

- The need to reduce GHG emissions means that there will be a large amount of intermittent renewable generators in the future and a way to maintain system balance must be found.
- Building new conventional generation plant or storage plant is very expensive.
- Operating conventional generation plant at low annual load factors is very expensive.
- In contrast to energy storage, DR requires no major technological breakthroughs.
- The committed construction of a dedicated smart metering communications infrastructure, together with the maturity of other telecommunications networks, means that there are a number of alternatives to support the communication requirements of DR; none of which require DR to bear the entire capital and operating costs.

However, if DR appears to offer such a strong case for its adoption, it is curious that we do not see a commitment to a large-scale deployment alongside that of smart metering. The hypothesis is that there must be some significant barrier to its adoption that is not immediately apparent, but which is acting to reduce the incentive to invest. The starting point for this research is therefore the following question:

> What is preventing a policy commitment to, and investment in, the large scale adoption of demand flexibility as a core component of system balancing within the future GB electrical energy system?

As discussed above, DR does not entail investment or technical development exceeding that required by the alternatives. It is unlikely then that the barriers to adoption are due to either cost or missing technology. It is more likely that the reason for a lack of commitment and investment is due to a perception that for whatever reason the use of demand flexibility for core system balancing incurs too much risk.

The perceived risk is likely due to a combination of factors such as:

- Will consumers deem the service impact acceptable relative to any reward they might receive?
- Can the service be trusted to provide balancing services that enable system reliability to be maintained at a level comparable to those achievable using conventional alternatives?
- What is the business model? How can a profit be made?
- How does it impact existing investments and business models?

These are valid questions and taken together the adoption of demand flexibility as a core component of system operation might seem too risky a proposition for consumers, industry and politicians alike. However, the alternatives are potentially so expensive in comparison that it is logical to make an attempt to break down this perception of risk. The purpose of this programme of research, and the subject of this thesis, is to begin that process. In recognition of the first bullet above, the focus is on a type of DR that has a core requirement of utilising demand flexibility in a way that is acceptable to consumers. In this work this is referred to as Flexible Demand (FD).

1.2 Approach to the Problem

The research described in this thesis was undertaken with the objective of satisfying the requirements of a research component for an Engineering Doctorate in Systems Engineering. The problem described in the preceding section is multi-faceted and multi-disciplinary encompassing technical, economic, legal and social factors. The interplay between these factors means that the problem must be considered in a holistic manner. It is therefore ideally suited to a Systems Engineering style of analysis.

The first task is to consider the context of the problem. Chapter 2 provides this context through a literature review that establishes how the current GB system operates – with a particular emphasis on balancing the system – and by characterising the different types of DR services, including FD.

When radical change is proposed it also makes sense to look back in history and see what lessons can be learnt. Chapter 3 looks at the evolution of the GB electricity industry in the post-war period and considers the drivers for change, both technical and political. From this analysis the need to both demonstrate the technology and provide a means to understand its impact is deduced.

A £1 million collaborative research project entitled "Demonstration of Distributed Flexible Demand (DD-FD)" and led by E.ON Engineering Limited (now E.ON New Build and Technology Limited) provided the vehicle in which to do this and the remainder of this research was conducted within the scope of that project.

The next task is to demonstrate that FD is able to supply a balancing service. A major part of that is to show that a large number of small loads can be dispatched in such a way that their concerted action produces an aggregate effect that is useful as a balancing service. This requires development of an algorithm that can do this in an optimised way. The architecture of the FD demonstrator is is described in Chapter 4. Chapters 5 and 6 describe the design and testing of an appropriate algorithm.

The final task is to develop a framework that can be used to investigate the impact on existing organisations and structures of adding FD. This assumes that the implementation of FD is by adding it to the existing system rather than by redesigning the system to include FD. This approach is evolution rather than revolution. It is considered a more realistic and practical approach to the adoption of FD, given the need to maintain service to consumers with minimal cost implications whilst any changes are implemented. With such a framework it is possible to see how relationships need to change and this will facilitate the development of appropriate incentives to support business models that exploit the technology. The development of such a framework based on the Systems Modelling Language (SysML) is described in Chapter 7.

Chapter 2

Literature Review

2.1 Introduction

In this chapter a review of UK energy policy together with a description of the operation of the British electricity system and how this relates to DR services is provided. This review is essential to understand the context into which Flexible Demand is required to integrate. Note that literature reviews pertaining to optimised scheduling and enterprise modelling are provided within the relevant chapters: Chapter 5 for optimised scheduling and Chapter 7 for enterprise modelling.

Modern human societies are highly reliant on the use of energy to power the machines used to overcome the limitations placed on them by their own physical bodies. Technological progress depends largely on the availability of ever-greater energy resources. Energy is consumed by industrial processes, domestic and commercial consumers, and transport. Prior to the industrial revolution, societies used sources of energy drawn directly from the environment around them. The discovery of ways to liberate the concentrated chemical energy stored within fossil fuels led directly to a step change in the pace of technological progress.

The discovery of electromagnetic induction spurred a further period of technological development and the widespread adoption of electricity as a multi-purpose energy carrier. For many decades the principal energy source for electricity generation has been fossil fuels. However, the recognition of the need to reduce emissions of carbon dioxide is forcing a return to the environmental energy sources that dominated prior to the use of fossil fuels. This change has a significant impact on the way society sources and manages energy. This chapter provides an overview of current and future UK energy needs.

2.2 Current UK Energy Consumption

Today direct use of fossil fuel energy through combustion at the point of use is the preferred mechanism for the production of low and high-grade heat. The same is true for transport mainly via the internal combustion and jet engines. Electricity is principally used to power appliances, both domestic and commercial, and industrial processes. Comparatively little electricity powers either heating or transport. Some electricity is generated directly from renewable sources such as wind and water, but most electricity consumed is generated from a primary fuel such as coal, gas or uranium.

Figure 1 illustrates the situation for the UK for the year 2012 (DECC, 2013). The chart on the left shows in what form consumers consumed energy. The majority of energy (77.5%) is delivered to consumers as hydrocarbon fuels in the form of natural gas or petroleum. Only 18.5% of energy is currently delivered in electrical form. Of the electrical energy consumed, 67% is generated from the burning of hydrocarbon fuels as illustrated in the chart on the right. It is clear that today the UK is heavily dependent on fossil fuels for both primary and secondary (principally electricity) energy.

Energy is supplied to consumers using three main distribution networks reflecting both the physical form of the energy carrier and the principal use of it. Natural gas is almost exclusively used for heating and power generation and is principally supplied using a national network of underground gas pipelines owned and operated by National Grid. The network comprises over 278,000 km of pipes and connects the majority of consumers (National Grid, 2011b). Petroleum products (mainly diesel, petrol and kerosene) are principally used for transport (~70% in 2012 (DECC, 2013)) and are supplied in bulk form through a nationwide network of distributors operating fleets of vehicles and distribution outlets. No one company dominates this sector of energy distribution in the way National Grid does for natural gas.

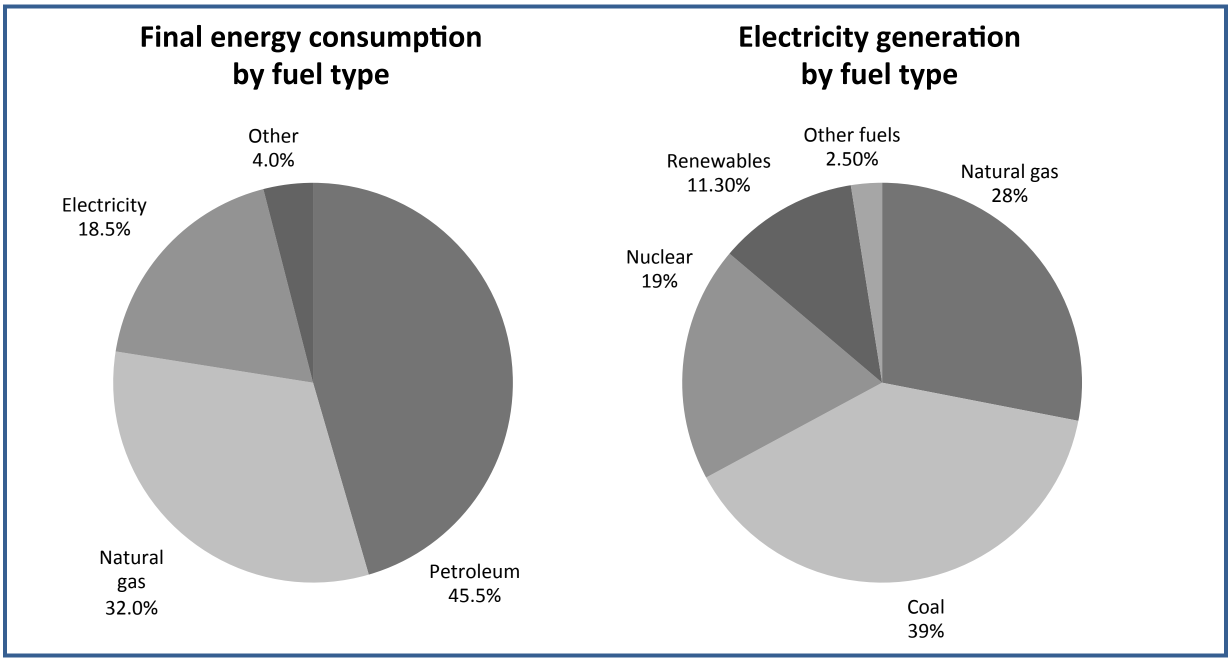

Figure 1: UK final energy consumption and electricity generation by fuel type for 2012.

Although smaller in volume than either the natural gas or petroleum energy sectors, the electrical energy sector is of immense importance. This is due to the utility of electrical energy. The energy stored within natural gas and petroleum products is converted into either thermal or kinetic energy that is concentrated and ideally suited for heating and transport. It is not a good fit where a large number of much lower power tasks, perhaps requiring fine control over the energy input, are required. It is for these tasks that electrical energy is an ideal fit. It can be converted into a variety of end uses including amongst others mechanical work, heating, lighting and information processing. Such conversions can be highly efficient and finely controlled. The tremendous utility of electrical energy means that it holds great value to consumers.

An analysis of the nature of the electrical demand is useful to understand what kinds of loads it is comprised of. This is important, as this is the basis of the flexibility that is used by FD services. In 2012 the overall electricity demand in the GB system was 319.4 TWh. This demand was divided between different consumer types as shown in Table 1 (DECC, 2013). It is immediately apparent from these figures that a substantial proportion of the demand is domestic and other consumers, as opposed to industrial. This reflects changes in the UK economy over the last few decades which has moved it away from a dependence on heavy industry.

Table 1: key 2012 demand statistics for the GB electricity system (DECC, 2013).

Consumer type	Demand
Iron and steel	3.4 TWh
Other industry	94.5 TWh
Transport	4.1 TWh
Domestic	114.7 TWh
Other consumers	101.0 TWh
Exports	1.7 TWh
TOTAL	319.4 TWh

Heavy industry such as iron and steel making predominantly use electricity to power industrial processes whereas light industry and domestic consumers predominantly use electricity to power appliances such as refrigeration, heating, ventilation and information processing. This change in the demand base means therefore that there is a corresponding shift in the load type away from industrial processes to appliances. The most intensive energy consuming appliances are those that perform some kind of thermal process (e.g. refrigeration, heating, ventilation, air conditioning, washing). These types of appliance usually have an inherent energy store that means they are well suited to FD and so this shift may well work in favour of the implementation of FD.

2.3 Future UK energy policy

There has been considerable public debate in recent times over climate change and, in particular, global warming. Whilst the debate in the media has been, and to some extent remains, particularly heated (Boykoff and Boykoff, 2007), the debate within the scientific community is largely over. In its Fourth Assessment Report published in 2007, the Intergovernmental Panel on Climate Change (IPCC) states (IPCC, 2007):

> "Warming of the climate system is unequivocal, as is now evident from observations of increases in global average air and ocean temperatures, widespread melting of snow and ice and rising global average sea level."

Policy makers are of course heavily influenced by public opinion but are also receptive to the prevailing scientific opinion. Within Western governments there is now general acceptance amongst policy makers that there is a need to act on the emissions of GHGs in order to counter climate change. In the UK concern over the implications of climate change led to the 2006 Stern Review which established the economic case for addressing the issue (Stern, 2007).

In 2007 the European Council of the European Union (EU) set two key targets for member states (Commission of the European Communities, 2008):

- A reduction of at least 20% in GHGs by 2020 – rising to 30% if there is an international agreement committing other developed countries to "comparable emission reductions and economically more advanced developing countries to contributing adequately according to their responsibilities and respective capabilities".
- A 20% share of renewable energies in EU energy consumption by 2020.

This led to the UK Climate Change Act 2008 that has established legally binding targets for overall UK emissions reductions for the years 2020 and 2050, and budgets for each of the three five year periods from 2008 to 2022. These targets and budgets are shown in Table 2 (UK Government, 2008a).

Table 2: emissions reduction targets as set out in the Climate Change Act 2008 and a 2009 Statutory Instrument (UK Government, 2009a; UK Government, 2009b).

Year	Target (below the baseline of 1990 emissions)
2008-2012	22%
2013-2017	28%
2018-2022	34%
2020	34%†
2050	80%
† Originally stated as 26% in the Climate Change Act 2008 but amended in a Statutory Instrument in 2009.	

Whilst the Climate Change Act 2008 does not place targets on individual industrial sectors, the Department for Energy and Climate Change (DECC) has sought to establish how individual industrial sectors, including the electrical power sector, might contribute to these objectives. DECC's 2050 Pathways Analysis report (DECC, 2010) presents its view of how the targets might be reached. This report describes six potential pathways (labelled Alpha, Beta, Gamma, Delta, Epsilon and Zeta) plus a reference case.

Pathway Alpha assumes a balanced approach including behaviour change on the part of consumers and businesses; expansion of renewables; introduction of Carbon Capture and Storage (CCS) technology and replacement of existing nuclear assets with new nuclear plant. The other pathways are variations on Alpha with each case considering the exclusion of one of the key Alpha requirements. The reference case represents a business-as-usual pathway. Pathway Alpha reflects current policy and represents the current strategy and is considered here as the most likely scenario.

Pathway Alpha assumes a relatively flat overall energy demand between today and 2050 but a doubling of electricity generation from just less than 400 TWh per year today to more than 800 TWh per year in 2050. This reflects the anticipated electrification of transport and heating, currently fuelled predominantly by fossil fuel derived hydrocarbons. The electricity demand is met by a declining base of unabated thermal generation – entirely eliminated by 2050 – to be replaced by renewables, nuclear and CCS plant in roughly equal quantities. If Pathway Alpha is to be believed then this implies that by 2050 one third of all electricity generation in the UK will be sourced from generators having an intermittent supply characteristic that is to a larger or lesser degree unpredictable, depending on the primary energy source.

In this scenario "predict and provide" is no longer viable. Whilst renewables are unlikely to be relied upon for energy security they will nevertheless be used whenever they are available. This will be driven by economic and environmental objectives. Once the capital cost of construction has been incurred, operation of nonthermal renewable generation, e.g. wind, is low cost due to the effectively zero cost of the primary energy source. Cost is only incurred in servicing the plant construction costs and in covering maintenance and distribution costs. Over the life of the asset the cost per MWh is therefore directly related to the total energy produced over its life. Maximum return is achieved if every unit of energy available to the plant is captured and sold. Clearly plant availability, distribution capacity and market price are crucial to such generators. Generation also incurs very little carbon penalty once the plant has been constructed. For these reasons in a minimum carbon system renewable generation should always be operated at full output, restricted only by its own availability and the energy available within the primary energy source.

Operation of renewables at full output implies that the variability in the primary energy source will be directly superimposed on the power available within the system. Given that it can take several hours for alternative generation (thermal CCS or nuclear) to be ready to supply extra power it is clear that balancing the system will become more difficult than it is today. In the future there will be greater uncertainty in the power available a few hours into the future, i.e. when it is too late to request additional thermal generation to come on-line or, conversely, to stand down. Therefore, without a significant expansion of storage there must either be larger reserve margins or there must be considerably more participation of demand in system balancing than there is today.

2.4 GB Electricity System Architecture

In this section the architecture of the electrical energy generation and distribution system within Great Britain is described. The system of Northern Ireland is specifically excluded as it is integrated with that of the Republic of Ireland and is operated as part of the Irish system. The Irish and British systems are interconnected via an HVDC interconnector but are not synchronised with each other.

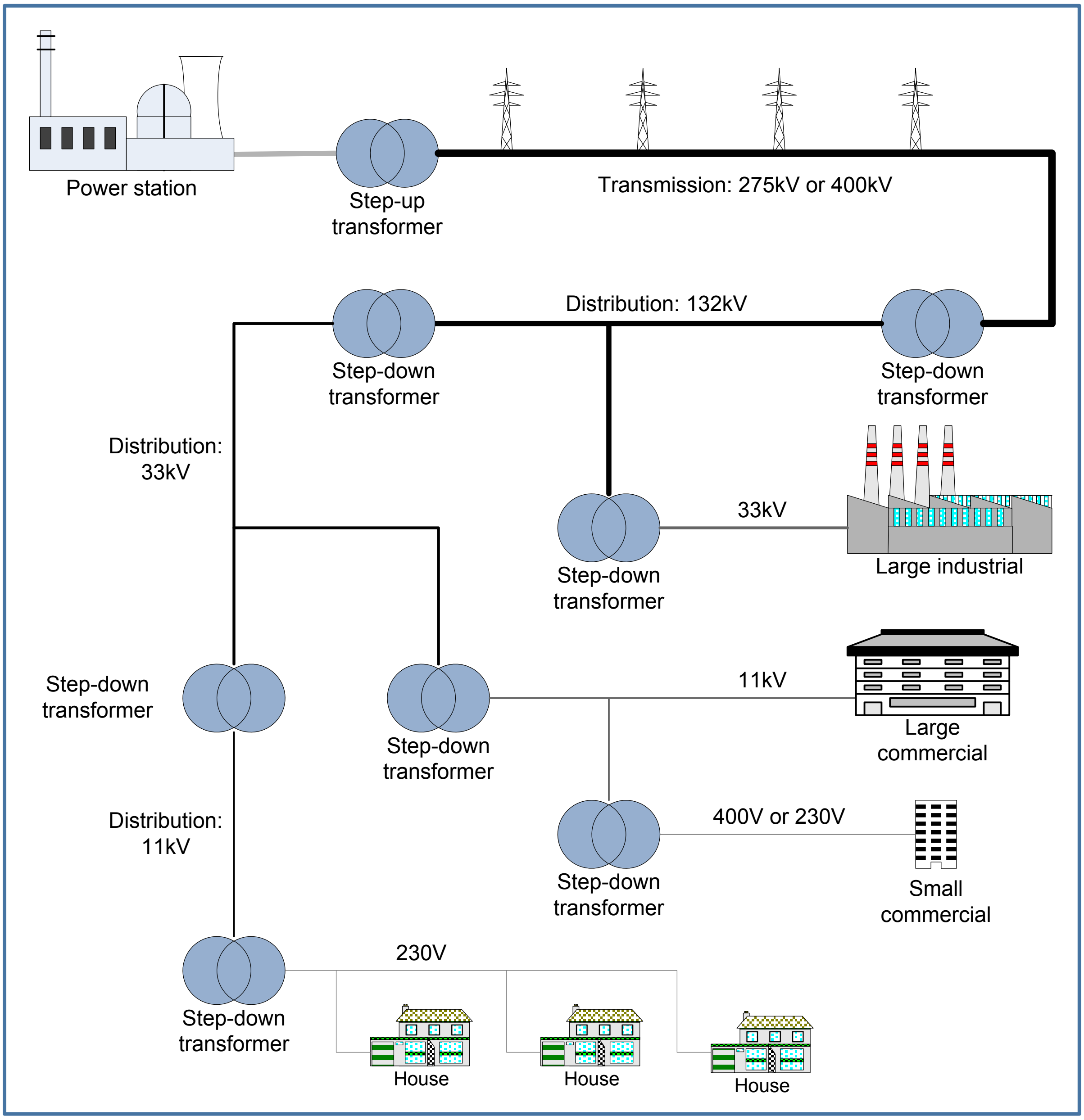

Figure 2: GB electricity network architecture.

The British system has evolved into its present state over a considerable period of time. Electrical generation and supply systems were first built independent and isolated from each other providing power to a local town or metropolitan area. As networks and demand expanded some level of interconnection took place although this was often hampered by incompatible technical standards. Beginning in the 1930s construction of the national grid started with the intention of interconnecting the large number of disparate systems. This facilitated greater scale and a drive to greater efficiency. This in turn led to a structure characterised by large, centralised generation, a high voltage transmission network and medium-to-low voltage distribution networks. The network architecture of the British system today is shown in Figure 2.

The architecture is hierarchical and top-down. Energy flows from the generators to the loads. Very little information flows back from the loads up the system hierarchy. Currently the only information that can be said to flow back up the hierarchy is metering data. It is often sampled at most once a month. Power is generated in a small number of relatively large power stations and flows through a network of cables and transformers to a very large number of consumers whose individual demands are very much lower than the power station capacity. This hierarchy of power flows is reflected in the hierarchy of voltages that is chosen to match the required power flow over a particular cable. Higher voltages are used where higher powers are required, to minimise current flows and the accompanying heating and thermal losses.

Although there is a SCADA network in the high voltage grid and in some of the medium and low voltage distribution networks this functionality does not extend all the way to all consumers. There is, therefore, currently very limited visibility of the electrical characteristics at the consumer termination of

Definition

SCADA is an abbreviation for Supervisory, Control And Data Acquisition: a term describing information networks intended to provide telemetry and command capability.

the network. This is perceived as a major deficiency and there is currently extensive interest in the deployment of Smart Grid and Smart Meter technology to address this (Farhangi, 2010).

2.5 Electricity Generation

A variety of different energy sources are used but all power stations either produce electrical power by liberating energy stored within a fuel or by converting energy available in the environment into electrical energy. Of those that use stored energy this can either be in the form of a fuel or as potential energy (e.g. hydroelectric dam schemes). Some example energy sources are shown in Figure 3.

All generation schemes that use a fuel source inherently have the capability to be dispatched. That is, they can be called upon to generate at any time subject to fuel being available. Controlling the rate of fuel consumption controls their output. Of those schemes that harvest energy from the environment any scheme that has a storage capability also inherently has the capability to be dispatched. Thus hydroelectric dams can be called upon to generate subject to the level of water behind the dam. Controlling the water flow controls their output. Kinetic and solar energy schemes convert environmental energy into electrical energy in real-time and do not have any inherent storage; they can only produce a useful output when the environmental energy source is available, e.g. the wind is blowing or the sun is shining. Their output is controllable but only within the limits of the energy available. Unlike schemes with storage, if the output of these schemes is reduced below that of the energy source then the balance is lost into the environment. In a wind scheme this might be through angling the rotor blades to spill some of the wind and in a solar array this might be by moving operation away from the maximum power point.

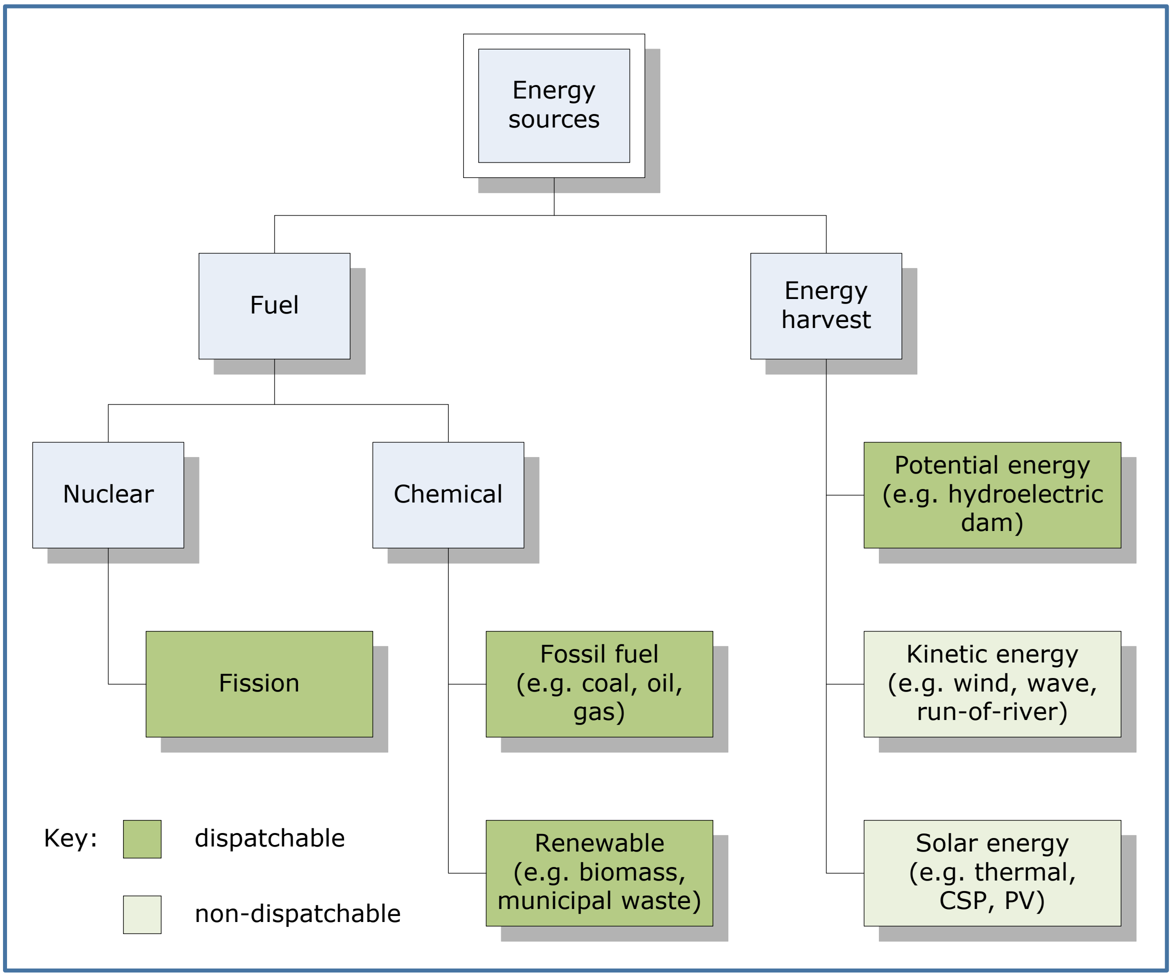

Figure 3: example sources of energy used in electrical power schemes.

Fuel based generation schemes are principally thermal power stations that produce heat through chemical or nuclear reactions. Gas turbines use the combustion gases to directly drive a turbine that powers a generator. This is known as an Open Cycle Gas Turbine (OCGT). Where the hot exhaust gases are further used to raise steam to drive a steam turbine to generate additional electricity it is known as a Combined Cycle Gas Turbine (CCGT). The CCGT is more efficient overall but less responsive than an OCGT. Non-gas schemes capture the heat of reaction in a coolant that is either directly or indirectly used to drive steam turbines that drive electrical generators. Energy harvest schemes based on wind and water use the kinetic energy of their working fluid to drive a generator whilst solar schemes either generate electricity directly (solar photovoltaic) or concentrate the solar radiation to generate heat in a working fluid in a similar manner to thermal power stations.

The exact mix of generation types within the system varies from year-to-year, as new generators are commissioned and old generators retired. A summary of the key generation statistics for 2012 is provided in Table 3 whilst Figure 4 shows the relative sizes of the major generating asset categories for the same year. Note that this chart uses a reduced number of categories to that shown in Table 3 by combining some sub-categories. This has been done for the sake of clarity. The mix is dominated by fossil fuels, predominantly coal and gas. Together these assets comprise ~ 67% of the generating mix. Oil, another fossil fuel, makes up another 2.6% of the mix leaving only ~ 30% of the mix supplied by non-fossil fuel sources (including nuclear). Figure 5 shows the average power station size by generation type. Coal, CCGT and nuclear stations tend to be much larger on average than their renewable alternatives. It is clear that a shift to a low carbon electricity system will lead to a fundamental restructuring of the GB generating base. This is considered further in Chapter 3.

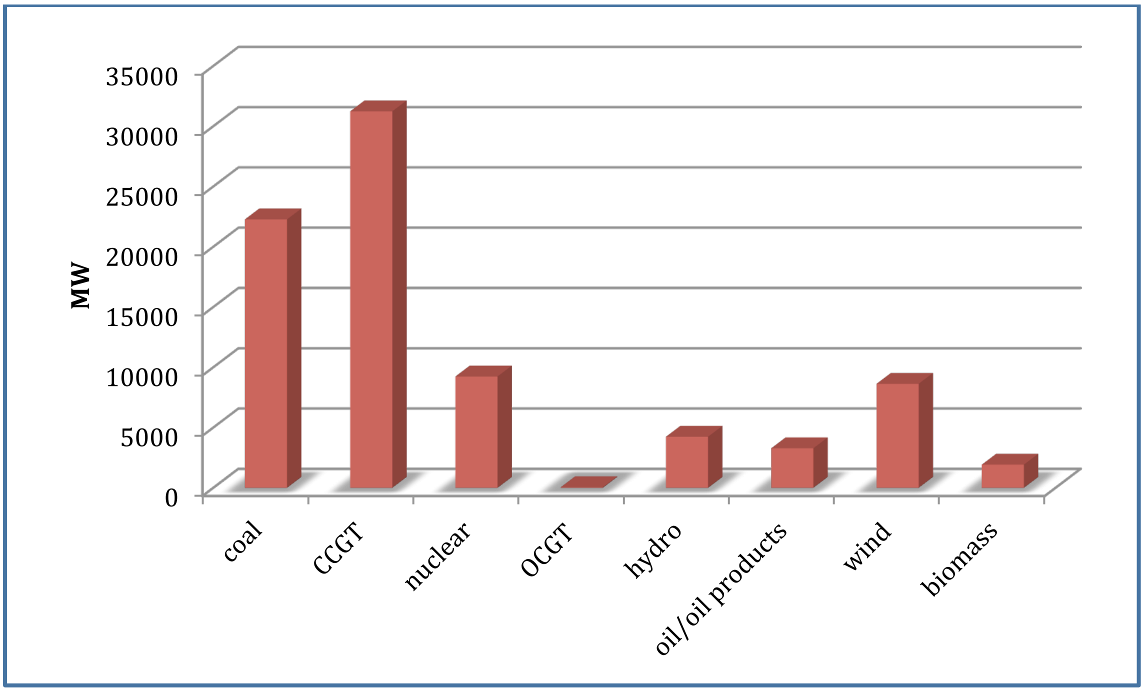

Figure 4: installed generation capacity by generation category (DECC, 2013).

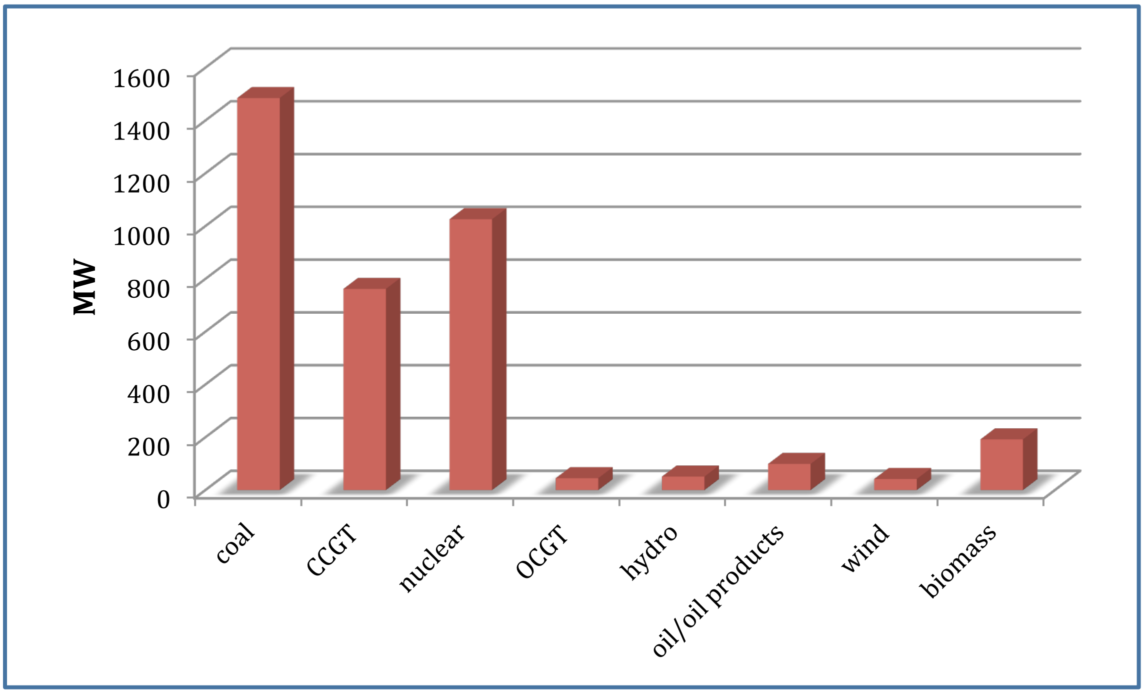

Figure 5: average power station size by generation category (DECC, 2013).

Table 3: key UK generation statistics for 2012 (DECC, 2013).

		Generation capacity (MW)	**Average load factor**
Major Power Producers	Coal	23,072	48.6% (of which coal-fired stations 57.1%)
	Oil	2,338	
	Mixed or dual fired	3,113	
	Gas turbines and oil engines	1,651	
	Other renewables	1,096	
	Nuclear	9,946	70.8%
	CCGT	33,113	30.4%
	Hydro	1,392	35.8%
	Pumped storage	2,744	12.3%
	Wind	3,277	25.6% (onshore) 33.7% (offshore)
Other generators	Conventional steam	2,446	Load factors were not available for this category of generation.
	CCGT	2,207	
	Hydro	157	
	Wind	492	
	Other renewables	2,196	
Total generation capacity		89,241 MW	
Maximum load		57,490 MW (64% of capacity)	

2.6 System Operation

To maintain system stability the amount of generation must match the demand in real-time. In an AC system, as is the case for every major power system worldwide, the system frequency provides a measure of the system balance. Excursions from the nominal frequency, 50 Hz in the UK and Europe, indicate an imbalance: an increasing frequency indicates generation exceeds demand and a decreasing frequency indicates demand exceeds generation. The frequency is the same across the system and this signal is therefore of immense utility for system control. However, it only provides an indication of *aggregate* behaviour and cannot be resolved to determine load behaviour in particular parts of the system.

The GB System Operator (SO) has the statutory duty to maintain system stability and is required by legislation to maintain the frequency within ± 0.5 Hz of the nominal frequency. In the GB system the SO is National Grid[1]. In practice National Grid applies internal operational parameters to maintain it within ± 0.2 Hz. Real-time control is achieved through enforcing, via the Grid Code[2], certain active generators on the system to automatically change their active power output in response to real-time fluctuations in frequency. The frequency is constantly changing as the demand changes in real time and the generation changes to match it but there are few excursions outside the operational parameters under normal circumstances. An example trace over a 60 minute period using data taken directly from the National Grid website is shown in Figure 6. System operation relies on accurately predicting future demand and ensuring that sufficient generation is available to meet it: "predict and provide".

[1] See http://www.nationalgrid.com/uk/Electricity/

[2] See http://www.nationalgrid.com/uk/Electricity/Codes/gridcode/gridcodedocs/

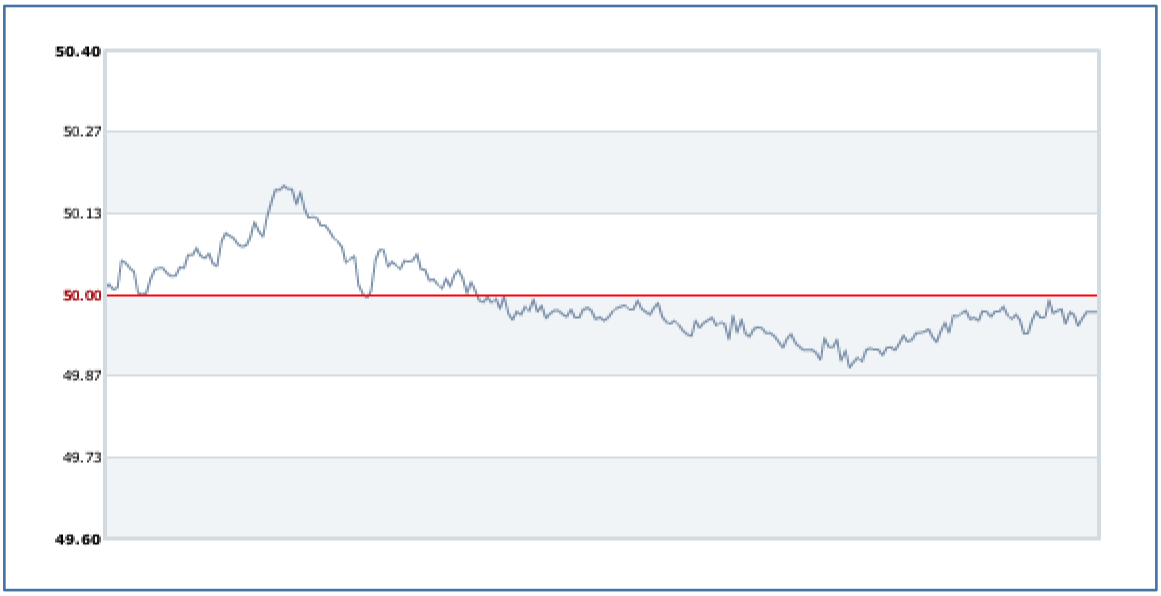

Figure 6: example frequency trace over a 60-minute period for the GB system (National Grid, 2011a).

This mode of operation is summarised in Figure 7. In this mode the SO is required to predict what the demand will be in a variety of different timeframes ranging from the next settlement period through to years. The SO uses a wide variety of information to inform this forecast including historical data, weather forecasts and knowledge of extraordinary energy consuming events or system maintenance. Generators are notified as required in response to demand fluctuation or unforeseen events and the required energy is delivered to maintain the system frequency at its nominal value.

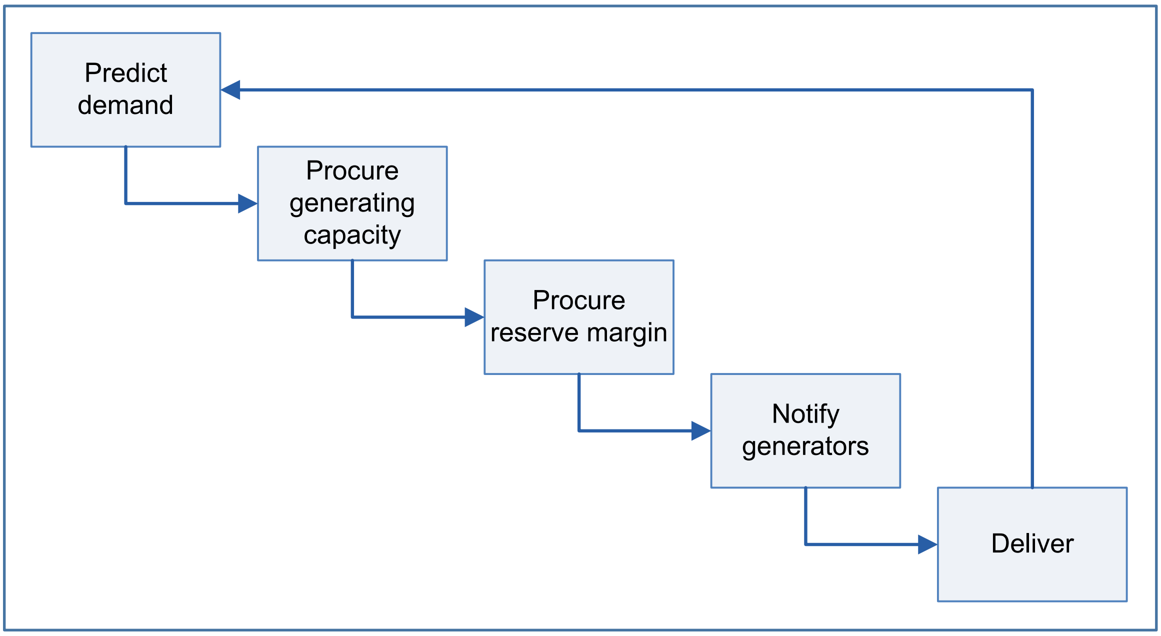

Figure 7: the "predict and provide" operating paradigm.

This process is repeated over and over each day to ensure continuity of supply at the minimum cost to the consumer. The stability of the system over many decades is testimony to the effectiveness of this operating model. However, it is dependent on two principal assumptions:

- That it is possible to control generation, i.e. that the generating capacity will be there when it is needed.
- That it is acceptable to incur the costs associated with very low utilisation of some assets such as peaking plant.

Power demand varies significantly over a single day, for example Figure 8 shows a demand over a typical winter day. This large variation in the daily demand profile could be met by having all stations on-line and varying their output in real-time, but to do so would be highly inefficient. Variations over the day in excess of 10 GW are not uncommon. As can be deduced from Figure 5, this equates to the entire output from a number of stations. To maintain all stations on-line would mean that for much of the day a significant number of generating units would be on-line and synchronised but producing very much less than their rated output whilst continuing to incur the costs of operating the units. These additional costs would be borne ultimately by consumers. A more efficient solution is to only have on-line sufficient generation to meet the predicted demand plus a margin to cover unforeseen events. As it takes time for a power station to reach operational readiness from off-line the SO must schedule power plant in advance. Forecast accuracy is therefore of critical importance to system operation.

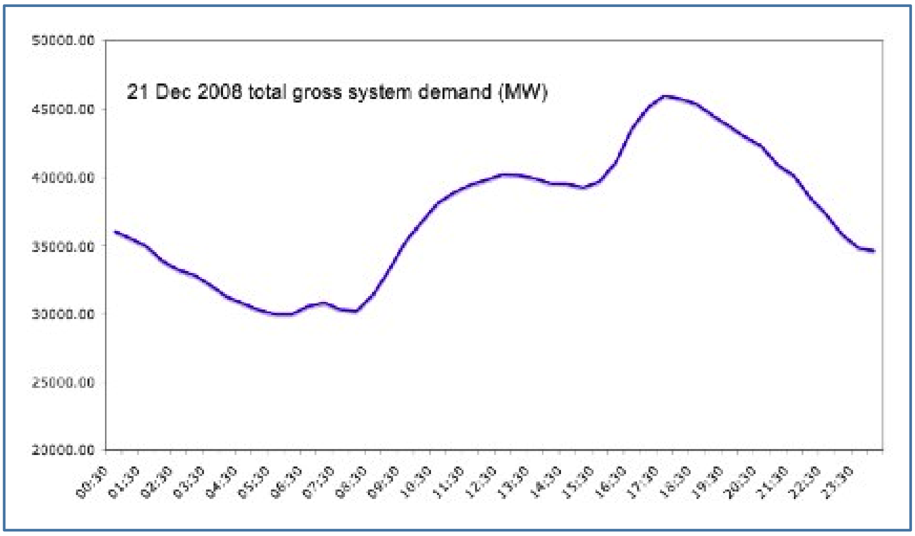

Figure 8: example variation in demand over a typical winter day (source: National Grid).

In addition to normal operation the SO must also ensure that there is sufficient reserve capacity available on the system to guard against the single biggest outage. In the GB system this equates to the loss of Heysham 2 nuclear station at full output, i.e. 1220 MW (DECC, 2013). Such losses can occur without warning and in such cases the SO must respond rapidly to maintain system stability. It is also possible for coordinated consumer demand to result in significant increases in demand over very short periods. Such coordinated action typically results from consumer response to television programmes with widespread interest such as major sporting occasions. In the majority of cases such demands can be anticipated and planned for. Despite these difficulties the GB SO is able to maintain an overall reliability of supply > 99.999% calculated as:

$$\left[1-\left(\frac{\text{Estimated unsupplied energy}}{\begin{array}{c}\text{Total energy that would have been supplied by the}\\ \text{transmission system had there been no loss of supply incidents}\end{array}}\right)\right]\times 100\%$$

Source (National Grid, 2010)

2.7 Balancing Services

To maintain system balance under the operating requirements described above, the SO procures balancing services from Generators to provide it with the reserve margin it needs to ensure that there is always sufficient generation available to meet demand. This requirement extends from real time through to years in advance. FD services do not generate electricity; instead they provide a means to adjust demand to meet supply and it is expected therefore that the principal market for FD services will be the balancing market. It is worthwhile taking a deeper look at these services.

Definition

In this thesis the term *Generator* refers specifically to an entity that generates electrical power and injects it into the National Grid.

National Grid produces a Balancing Principles Statement (National Grid, 2009a) that describes the principles underlying its procurement of balancing services. The services themselves are described in the Procurement Guidelines (National Grid, 2006) and are summarised below. National Grid currently procures three different types of balancing service:

- Ancillary services
- Buying or selling electricity via offers and bids made in the Balancing Mechanism (BM)
- Other energy related products

In its procurement guidelines, National Grid specifically states that it is interested in procuring services from DR providers both via the BM and via bilateral agreements outside of the BM. Table 4 summarises these balancing services.

Table 4: balancing services procured by National Grid.

Service	Balancing Mechanism	Bilateral Agreement
Reactive power	✓	
Fast reserve	✓	✓
Short term operating reserve (STOR)	✓	
Frequency response		✓
Demand inter-trip		✓

To understand whether these services present possible market opportunities for FD it is important to take a closer look at them.

- **Reactive power:** the injection of reactive power is required at various points across the network to facilitate the control of system voltages. This is unlikely to be supplied by FD as it is unlikely to be a source of reactive power.
- **Fast reserve:** this service provides rapid and reliable delivery of active power or demand reduction. Delivery must be within two minutes of the instruction

from National Grid, to be sustainable for a minimum of 15 minutes and to ramp at a rate in excess of 25 MW/minute. This service provides rapid response to unforeseen frequency changes on the system and could potentially be supplied by FD.

- **Short Term Operating Reserve (STOR):** this is a contracted service with a minimum capability of 3 MW available within 240 minutes of an instruction from National Grid. The delivery must be sustainable for a minimum of two hours. Provision is either as a committed service or as a flexible service. FD could provide this.

- **Frequency response:** under this service class, there are three sub-services:

  - **Mandatory Frequency Response (MFR):** MFR capability is required of all Generators caught by the grid code and involves the provision of a range of variability on the generating units to match dynamic demand changes in real-time. This is not applicable to FD.

  - **Firm Frequency Response (FFR):** FFR is similar to MFR but National Grid contracts for this service rather than imposing it via the grid code. It is an automatic change in output or demand in response to changes in frequency. It can be dynamic, i.e. continuously varying output or consumption in response to frequency, or non-dynamic, i.e. an abrupt change triggered by a defined frequency deviation. FD could provide this.

  - **Frequency Control by Demand Management (FCDM):** FCDM is similar to non-dynamic FFR. A frequency excursion passing a pre-defined threshold triggers the action, usually via frequency sensitive relays. However, FCDM involves the rapid (within two seconds) and abrupt disconnection of demand as opposed to a defined change in consumption.

A provider of FCDM is required to make the service available 24 hours a day and to deliver the service for up to 30 minutes. There is a minimum requirement of 3 MW. FD could provide this.

- **Demand inter-trip:** under fault conditions, an inter-trip reduces or disconnects demand. Load disconnection occurs at specific points in the network. This is not applicable to FD.

In summary, the candidate services for FD are Fast Reserve, STOR, FFR and FCDM. Of these FCDM is specifically a demand management service. The faster services tend to be more valuable so Fast Reserve is more valuable than STOR however the requirements are considerably more stringent. FFR is an interesting option but is not a dispatched service; rather the generator or load is expected to alter its output in sympathy with the system frequency. FCDM is primarily aimed at the abrupt disconnection of large loads. In the next section the characteristics of FD services are examined to establish a best fit to the type of balancing service procured by National Grid.

2.8 Characteristics of FD Services

2.8.1 Demand Side Management and Flexible Demand

The services that might be provided by the demand side are all modifications to the pattern of consumption. Collectively these services form a group called Demand Side Management (DSM), of which FD is a sub-category. Demand Side Management is a collective term covering three different techniques for the management of demand side energy consumption. These techniques are energy efficiency, energy conservation and demand response (Boshell and Veloza, 2008). All three of these techniques have the aim of either reducing overall energy consumption by consumers, or modifying their patterns of energy consumption. In the case of modified consumption the aim is to smooth the aggregate demand curve to reduce the need for peaking plant. The three techniques are described in more detail below:

Definition

Flexible Demand is a type of Demand Response in which the power consumption of a flexible load is modified in such a way that the performance of the service provided by the load remains within the bounds of acceptability to the service user.

- **Energy efficiency:** measures taken to reduce the total amount of energy required for the provision of any given service. This is achieved largely through more efficient appliances and through the use of materials to prevent energy waste, e.g. thermal insulation.

- **Energy conservation:** modifying behaviours such that existing services are either eliminated or replaced by lower energy services. This is usually accompanied by a reduction in the utility of the services or some other compromise on the part of the consumer. For example, using a lower temperature cycle on a washing machine conserves energy but may also result in poorer washing performance. The consumer is therefore required to make a judgement as to what degree of reduction in service quality they will accept against the benefits of reduced energy consumption (such as reduced energy costs).

- **Demand Response (DR):** certain types of energy consuming services can have their pattern of energy consumption modified in response to events occurring on their connected energy supply networks. This applies equally to both gas and electricity networks but DR is more commonly associated with electricity networks. The effect on the energy consumer will vary according to the degree of intervention. The intervention could be an immediate cessation of supply without any consideration of the effect on the consumer, or it could be a graceful degradation of service.

The relationship between the different DSM categories is shown in Figure 9.

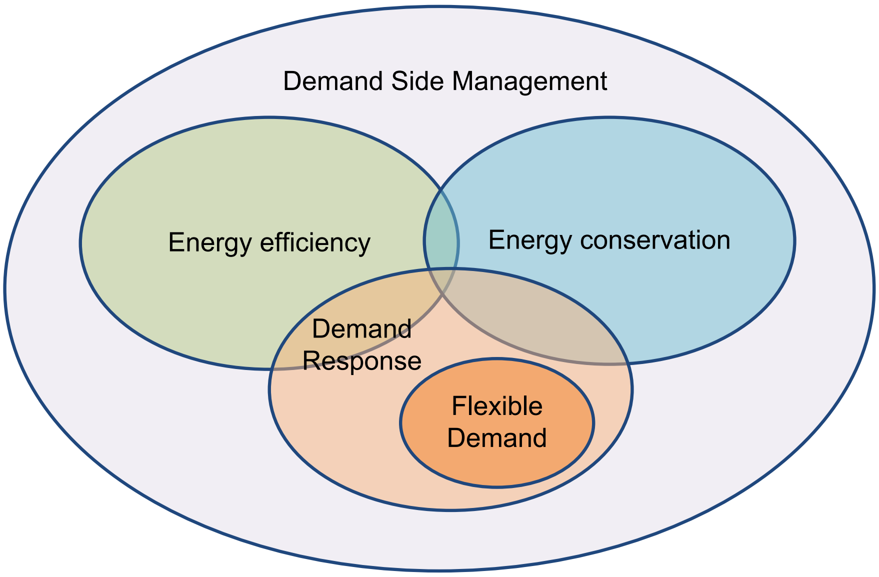

Figure 9: the service types comprising Demand Side Management.

FD services are a type of DR service but the objective is to provide DR in support of network operation in a manner in which the effect on the end user is minimised. Specifically, the end user retains control in that it can specify the constraints on the service performance and any consumption change will be such that the service remains within these performance constraints. For example, an electrically powered heating system might reduce its power consumption on command, but continue to consume enough power to ensure that the temperature remains within comfort limits as dictated by the end user. In the ideal case the end user will not notice the effect on their quality of service although this might not always be possible. These services exploit the inherent flexibility of certain energy consuming systems that either contain an element of energy storage, or have operation that is not time critical.

With systems that have an inherent store of energy the flexibility arises because operation is frequently to a much tighter tolerance than is necessary to provide the consumer with a satisfactory service. For example, many HVAC installations aim to keep the air temperature within a degree or so of the set point whereas most building occupants could tolerate larger variations in temperature without suffering undue discomfort[3]. The large thermal mass of the building provides a substantial energy store and considerable thermal inertia that can be used to advantage in providing flexibility.

With systems that are not time critical, the energy consuming operation can be delayed or advanced as required. For example, delaying the start of the wash cycle of a washing machine. In this case the demand response takes advantage of the consumer's flexibility in terms of when they need their clothes to be washed. However, it is important that the appliance is given responsibility for deciding when, and to what degree, the service can be interrupted since only the appliance can calculate the effect of so doing. An unanticipated interruption could result in annoyance, wastage or – in extreme cases – damage to the appliance or something upon which the appliance is operating.

Using FD, consumption can be delayed or advanced, either in part or whole. However, to utilise this latent flexibility any FD control system must be provided with some mechanism of knowing how much flexibility is available from the appliance at any one time, subject to any constraints imposed by the users of the appliance. This requires a more complex and intelligent mode of operation than DR services that seek to curtail energy consumption without consideration of the effect this will have on users of the energy consuming service.

[3] See ISO standard 7730 for a detailed discussion of this topic (BSI 2005).

A final point about FD relates to the perception of the service by those consumers who will participate by offering their flexibility. This point is non-technical but is an important aspect of gaining consumer acceptance and trust. *Demand Response* implies that consumers must respond to a command to alter their demand as a part of a centralised command and control model, with little or no opportunity to influence how their consumption might be modified: it evokes a perception of a loss of control. In contrast *Flexible Demand* implies the consumer possesses something of value – flexibility – and that the consumer is an active participant in the electrical supply industry, exercising choice in how their demand is used to aid overall system operation and effectively trading something of value that they own. In contrast it evokes a perception of retaining control and only ceding control on terms acceptable to them.

2.8.2 Types of FD Service

FD provides for the trade of a commodity: flexibility. Flexibility is a good that consumers of electrical energy possess and that forms the basis of trade in DR systems. FD trades flexibility in terms of modification to power consumption. By varying the timing and power consumption of individual loads in a coordinated manner it is possible to produce an aggregate effect that is similar to the way supply is varied in conventional generation. The entity that undertakes this aggregation is referred to as an *Aggregator* and acts as the intermediary between the consumers who each offer a relatively small amount of power and the balancing markets that typically require relatively large offered power as described above. Note that FD can also provide energy flexibility since energy is simply power × time.

In addition to the variables of time and power each individual managed load also has a fixed parameter in terms of its location on the power network. Whilst this cannot be varied for individual loads it is possible by choosing the loads that are to be used to create an aggregate effect at a specific point in the network. Again this is similar to generation where the SO might choose a particular generator based on the current pattern of power flow on the transmission network. In some ways the very large number of loads potentially creates the opportunity to more finely tune this effect than is possible with large, centralised generation. Potentially, FD can be localised to individual feeders within the distribution network offering the possibility of localised balancing within the distribution network.

Ultimately it is the consumer that agrees to provide the flexibility that forms the core of FD services. One way of encouraging consumers to participate is through financial reward. DR programmes can be classified as either incentive based or price based in which the decision to curtail – or increase – load is driven by electricity pricing (Albadi and El-Saadany, 2008). An incentive based programme pays the consumer for taking part in a programme, usually via some kind of credit. A price based programme rewards consumers for their participation. The desired behaviour is incentivised via appropriate energy pricing. Within each of these classes of programme are a number of alternative approaches as summarised in Table 5.

Table 5: types of DR programme

[after Albadi and El-Saadany (2008)].

Incentive based		**Price based**
Classical	**Market**	
Direct Load Control	Demand Bidding	Time Of Use (TOU)
Interruptible / Curtailable	Emergency	Critical Peak Pricing (CPP)
	Capacity	Extreme Day CPP
	Ancillary Services	Extreme Day Pricing
		Real Time Pricing (RTP)

Under the incentive based programmes are the classical and market based approaches. Classical programmes include those where direct control is exercised over the load under management, and those where energy consumption can be interrupted or curtailed. These programmes do not necessarily have any bidirectional communication and price signals do not form part of the control mechanism. The incentive is a financial reward for agreeing to the particular control method employed. Examples include the Economy 7[4] scheme and the FCDM balancing service described previously. Market based schemes are those in which a provider of flexibility bids into a financial market. The provider decides how much they are willing to sell their flexibility for and offers it for sale at that price. Such pricing might contain a capacity element, a usage element or both. The purchaser decides whether to accept the bid or not. In many market based approaches all winning bids are paid the price at which the market clears which can be highly

[4] See http://www.npower.com/Home/Electricity-and-gas/Types-of-meter/Economy7/index.htm for information on a typical Economy 7 tariff

beneficial to efficient providers. Such markets are not currently open to consumers as the amounts they are willing to sell are too small to meet minimum tradable quantity requirements, however Aggregators can and do participate. Within the UK there are a number of existing Aggregators that are already actively trading flexibility within the balancing market[5].

Price based programmes are aimed at consumers and aim to influence behaviour using price signals. Thus pricing might be varied such that it is more expensive at times of system stress. Consumers respond by curtailing non-essential use. The different types of price-based scheme vary mainly in their approach to timing. TOU[6] has a different price according to time of day whereas CPP[7] prices energy higher during the daily peak(s). Extreme day pricing applies TOU or CPP to specific days. RTP[8] reflects wholesale market spot pricing in the consumer pricing and moves the consumer price in sympathy with the wholesale markets.

Whichever type of programme is implemented, the central concept is that of the shifting of consumption. Consumption can only be shifted through the dispatch of controlled loads, either via direct control or via an intermediary such as a consumer as in the case of price-based schemes. Effective dispatch is critical to maximising the utility of DR. Where loads are managed by a control system then manual dispatch becomes infeasible for large-scale implementations that might be simultaneously

[5] See http://www.nationalgrid.com/uk/Electricity/Balancing/demandside/aggregators/

[6] See http://www.ausgrid.com.au/Common/Our-network/Metering/Time-of-use-pricing.aspx for an example scheme.

[7] See http://www.sdge.com/save-money/demand-response/critical-peak-pricing for an example scheme.

[8] See http://www.powersmartpricing.org/how-it-works/ for an example scheme.

controlling many thousands or even millions of individual loads. Such implementations require automated dispatch. This area is likely to be a key differentiator between different aggregators and is an area that is examined in more detail later in this thesis.

2.8.3 Market Opportunities for FD

The substantial changes to the generation mix necessitated by climate change mitigation policy mean that over the next few decades FD is likely to have a very significant opportunity. This is recognised by both DECC in its Pathways Analysis (DECC, 2010) and the GB SO (National Grid, 2009b). In its recent study into the options for delivering secure and sustainable energy supplies in the UK, the industry regulator – Ofgem – has also recognised the need for greater participation from the demand side (Ofgem, 2010).

There is indeed a compelling case for FD. In 2008, Capgemini estimated up to 202 TWh annual energy saving, 100 million tons CO_2 annual emissions reduction, €50 bn avoided investment and €35 bn annual savings for customers across the EU-15 group[9] are possible by 2020 with the implementation of FD (Chardon et al., 2008). Further, construction of 1 MW of FD capability can be of the order of 40% less capital cost than 1 MW of gas-fired peaking plant (Osborne and Warrier, 2007). Further incentive for FD is provided by the avoided costs it facilitates due to the existing low utilisation of the system (<50%) (Strbac, 2008). It has been argued that both the customer and the supplier are incentivised to participate in FD even without adding

[9] EU-15: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal, Spain, Sweden and the United Kingdom

any external costs such as emissions; the volatility in wholesale pricing provides sufficient incentive (Braithwait, 2003).

> **Definition**
> In this thesis the term *Supplier* refers specifically to an entity that supplies electricity to a consumer and is paid by the consumer.

Generators produce electricity that is carried to the consumer over a transmission and distribution network. A Supplier purchases electricity in the wholesale market and sells it to consumers in the retail market. The Supplier either reimburses Generators directly (via bilateral agreements) or indirectly (via the BM). Thus, energy flows from the Generator to the consumer and money flows in the opposite direction.

With pricing-based DR schemes as outlined above there is no other value chain: the effect of DR is simply to modify the quantities that flow over the existing value chain. FD, however, is different: it introduces a second value chain in which a different commodity, flexibility, is traded. The consumer sells flexibility to an Aggregator (who could be a Supplier) and flexibility thus flows from the consumer to the Supplier, Generator or SO via the Aggregator. Financial compensation flows to the consumer via the Aggregator. These value chains are shown in Figure 10.

It is for the Aggregator to decide how to aggregate many small consumption changes into an aggregated package that can be traded with, for example, the SO. Similarly, the Aggregator determines how to disaggregate the payment in order to compensate consumers for their flexibility.

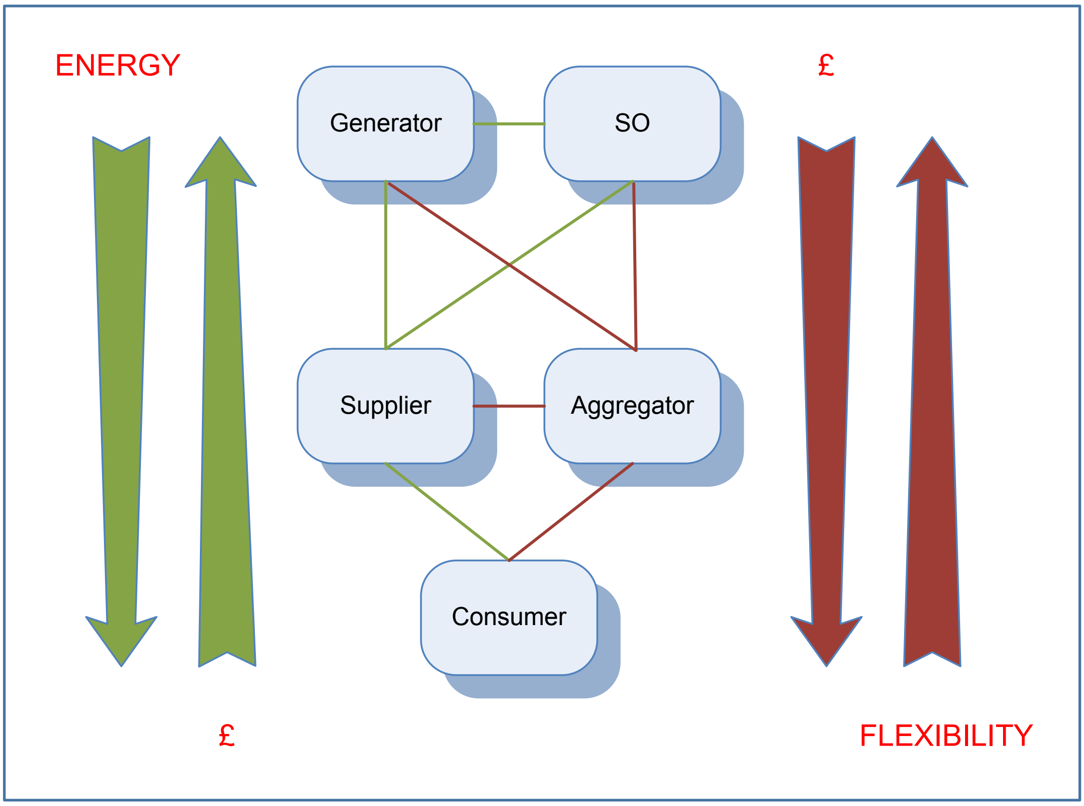

Figure 10: energy and flexibility value chains.

FD services used for balancing can be applied in a number of time horizons from real-time to hours, or even days, in advance. Although the initial target for FD services is the supply of balancing services to the SO, this is only because that is the only market that currently exists for these kinds of services. In a future scenario in which there are significant amounts of distributed generation connected to the distribution network, or where the sudden increase in electric vehicle and heating loads leads to capacity constraints within it, one can easily imagine a case where a Distribution Network Operator (DNO) is required to actively balance supply and demand on the distribution network in a manner similar to that performed by the SO on the transmission network. In this scenario the DNO effectively becomes a Distribution System Operator and a purchaser of balancing services.

2.9 Summary and Conclusions

This chapter has described the context in which FD services are envisioned. Hydrocarbon fuels dominate current UK energy consumption. Electricity provides 18% of energy needs at the point of consumption and a substantial proportion of that is derived from fossil fuels. The majority of UK consumption occurs in the domestic and commercial sectors. The 2006 Stern Review together with wider concerns about climate change led policy makers to enshrine targets for overall reduction in the level of GHG emissions. These targets were contained in the Climate Change Act 2008 and call for an 80% reduction by 2050. The question, then, is how to achieve it. DECC has proposed a number of pathways and it is clear that the substantial electrification of heat and transport, coupled with simultaneous de-carbonisation of electricity generation, is expected to play a key part in achieving these goals.

However, because electricity supply and demand must be continuously balanced the substantial use of intermittent renewable generation such as wind power presents a significant challenge to system operation. Unless significant investment takes place in peaking plant or energy storage there is the real risk of outage on the electricity system. Both options are expensive. There is, however, an alternative: by using the demand side as an integral component of system balancing it is possible to reduce the overall balancing costs whilst maintaining system reliability.

Demand side measures include energy efficiency, energy reduction and demand response. The focus of this work is the latter of these and, in particular, a sub-category in which active load management is done in such a way as to reduce (or increase) consumption whilst maintaining an acceptable quality of service to the consumer. This is termed *Flexible Demand*. FD is expected to be more acceptable to the consumer and therefore more commercially viable than alternative approaches.

Chapter 3

A Historical Analysis of the GB Electricity Industry

3.1 Introduction

When radical change is proposed it makes sense to look back in history to see if lessons can be learnt. With that in mind in this chapter the evolution of the GB electricity industry in the post-war period is examined. The objective is to determine the drivers for change and to consider not just technical drivers but also political drivers. The rationale is that in an industry that is so tightly linked to overall macroeconomic policy it is highly likely that policy is a key influence over investment decisions. To understand what barriers there might be to the implementation of FD an understanding of the interaction between policy and technology, and the policy evolution over the last several decades is an essential prerequisite. This is explored by considering how changes in policy over the years have led to structural changes within the industry.

3.2 Evolution of the GB Electricity Industry in the Post-war Era

The architecture and operation of the British electricity system is not dictated by technology alone, nor does technology dictate the commercial and regulatory structure of the industry. To a very large extent policy decisions have influenced the structure of both the system – technology, architecture and operation – and that of the industry including the network of commercial entities, relationships, markets, regulatory bodies, etc.

Before 1947 the GB electricity system comprised more than 550 separate electricity generation and supply undertakings. Many of these were municipal systems supplying local needs. The path to the modern framework began with the Electricity Act 1947 (UK Government, 1947) and between 1947 and today there has been considerable legislation aimed at implementing Government energy policy (see Figure 11). A number of substantial structural changes have taken place in response to policy decisions by the UK Government.

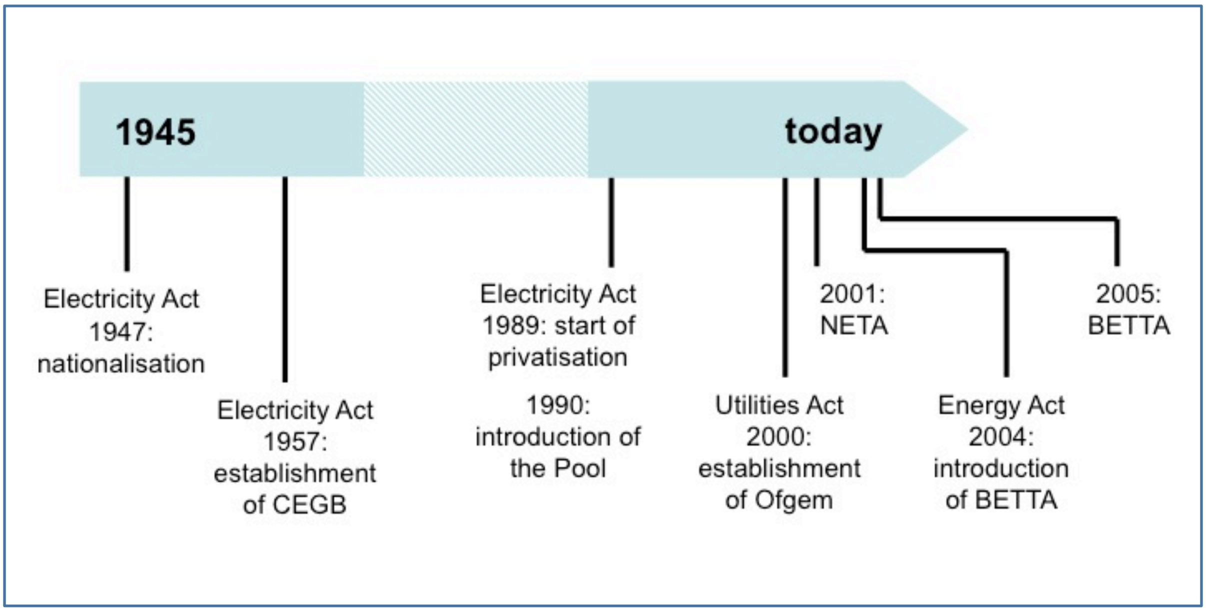

Figure 11: timeline showing key legislative developments within the UK electricity system.

The Electricity Act 1947 nationalised the electricity supply industry in the UK and created a single Central Authority, 12 Area Boards covering England and Wales, and two Area Boards in Scotland. In Scotland, the vertically integrated Area Boards were responsible for generation, transmission and distribution. In England and Wales, the

Central Authority was responsible for generation and transmission whilst the Area Boards were responsible for distribution.

The Electricity Act 1947 Act sprung from the pursuit by the post-war government of Clement Attlee of a 'mixed' economy in which several key industries, including electricity, were taken into public ownership. The principal aim of policy at the time was to recover the economy of the UK from the damage done to it by the Second World War and to provide more protection for the working classes through employment and the establishment of a welfare state (Childs, 2006). There was also techno-economic logic in consolidating the industry as such consolidation allows for the harmonisation of standards and the realisation of scale economies. Nevertheless, the structural change imposed on the electricity industry arose from a political philosophy that the UK would be better off if certain key industries were held in public ownership. The reform was therefore primarily driven by socio-economic rather than techno-economic concerns.

This structure lasted for ten years before modification in 1957 by further legislation. The Electricity Act 1957 (UK Government, 1957) provided for the dissolution of the Central Authority and replaced it with two bodies splitting the responsibilities of the former Central Authority between them. The role of the Central Electricity Generating Board (CEGB) was to provide generation and transmission. The role of the Electricity Council was to oversee the industry and provide a coordinating function in areas common to the CEGB and the Area Boards. This allowed the CEGB to concentrate on the business of generation and transmission without being concerned with non-core activities such as industry-level strategy and pay bargaining. Overall control of the industry remained with the government with the various heads of the industry bodies ultimately reporting to the relevant Secretary of State.

The industrial structure created by the Electricity Act 1957 remained largely unchanged for the next 32 years with the industry residing in continuous public ownership throughout the various changes of government. Throughout this period the overriding goal was reliability, enshrined in the CEGB's mission "to keep the lights on"[10]. During this period there was considerable technical evolution. The 1950s saw a rapid increase in the size of generating sets first to 200 MW and then 500 MW. These larger units brought significant economies of scale and efficiency increases. In 1948 the average power station implied efficiency was only 21% but by the mid 1950s there had been a step change in efficiency to > 30% (DECC, 2009).

At the same time extensive investment in transmission at 132 kV and 275 kV increased transmission efficiency. By the mid 1960s sections of the Grid were further upgraded to 400 kV. In addition new fuels were being considered to supplement coal (the primary fuel source). Following the success of the Calder Hall nuclear prototype, commissioned in 1956, the CEGB opened its first commercial nuclear power stations at Berkeley and Bradwell in 1963. Power station building meant that overall UK plant margin increased from zero in 1948 to > 50% in 1973 before falling back to 20-25% in the 1980s (DECC, 2009).

However, in 1989 the election of the Conservative Party under Margaret Thatcher led to a fundamental change in political philosophy. The incoming Conservative government fervently believed in the power of the market to foster innovation, increase efficiency and drive down prices. It embarked on radical reform of British industry involving the privatisation of a number of state industries in key sectors such as telecommunications (British Telecom in 1984) and manufacturing (e.g. British Aerospace in 1981). The energy industry also underwent privatisations such as BP (in stages between 1979 and 1987) and British Gas (in 1986). The electricity industry was

[10] See http://www.eurofound.europa.eu/eiro/2000/12/feature/uk0012105f.htm

also selected for privatisation and the enactment by the UK Parliament of the Electricity Act 1989 (UK Government, 1989) repealed in their entirety the Electricity Act 1947 and the Electricity Act 1957, and laid the legislative framework for the transition from public to private ownership. As in 1947 there was no substantive techno-economic justification for this change and the drivers for fundamental structural change to the industry were therefore once more socio-economic (Littlechild, 2010).

The Electricity Act 1989 retained the 12 Area Boards but allowed for the transfer of all their assets to new companies. The assets and functions of the CEGB were transferred into three generating companies and one transmission company. The generating companies were:

- PowerGen
- National Power
- National Power (Nuclear) [subsequently renamed Nuclear Electric]

The transmission company was National Grid Company. The purpose of the Electricity Act 1989 was to introduce competition into generation in England and Wales and to prepare the industry for eventual privatisation. The Electricity Act 1989 fundamentally changed the structure of the industry from a centrally managed, publicly owned monopoly to a market populated by competing private entities. The scale of this change was enormous and presented a significant challenge to the industry.

The new industry was characterised by a system of licensing overseen by a new Director General of Electricity Supply reporting to the relevant Secretary of State. The Director General had statutory duties to promote competition and to protect consumers. The mechanism for discharging these duties was no longer via direct control but now via regulation. From its initial conception the purpose of the Electricity Act 1989 was to begin the process of migration toward a market-based approach to electricity supply. Competition was to be the major driving force. The system of licensing would eventually allow new entrants access to the electricity markets. Initially the Crown owned in their entirety the bodies created by the Electricity Act 1989. Subsequently the Government privatised them through the gradual placing of their stock on the London Stock Exchange.

At first, only competition in generation in England and Wales was possible. The establishment of the "Pool" from April 1990 facilitated this. The generating companies bid into the Pool and the SO, i.e. National Grid Company, selected who would generate based on the bid prices. There was no demand side bidding into the Pool and so buyers were forced to pay the prices set by the supply side. The generating companies thus largely dictated wholesale prices. In addition there was no liberalisation of electricity supply and consumers had to take supply from the Regional Electricity Company (REC) that replaced their Area Board. At the time of privatisation there was to be no retail competition. The problem was that the generating companies were dependent on long term contracts with the RECs to ensure an adequate return and the RECs were unwilling to agree without the guarantee of a monopoly of supply in their area. As the Government's primary concern was its privatisation agenda it agreed to delay retail competition (Littlechild, 2010). By mid 1999, however, most consumers were able to purchase their electricity from any electricity supplier – not just the vertically integrated REC (Tovey, 2005).

The SO operated the Pool and all generating companies above a defined threshold were required to bid into the Pool. Those generating companies chosen to generate were then paid the System Marginal Price (SMP), i.e. the highest successful bid into the Pool or, in economics terms, the price at which the market cleared. This system gave rise to allegations of market manipulation due to it being effectively a duopoly between PowerGen and National Power. These companies were incentivised to withhold some generating capacity in order to maintain a high marginal price: the increased profit from smaller volume but higher pricing outweighing higher volume at a lower price (Woo et al., 2003). These concerns were addressed through the reform contained in the Utilities Act 2000 (UK Government, 2000). Through this Act the structure of the wholesale market was redesigned and the New Electricity Trading Arrangements (NETA) were introduced.

NETA became active in March 2001. It abolished the Pool and allowed for the first time bilateral agreements between generating companies and electricity supply companies, or between generating companies and large consumers, without the involvement of the SO. All parties now needed to notify the SO of their contracts – excluding the financial details – for each settlement period (48 half-hour periods within each day). The SO operated a Balancing Mechanism (BM) the purpose of which was to trade energy to ensure the SO could maintain the balance of supply and demand given that no party could have perfect foresight of what their respective energy supply availability or demand requirements would be. The introduction of NETA meant that the proportion of electricity traded outside of the BM rose to >95% (Tovey, 2004).

The Utilities Act 2000 also made changes to the regulatory structure of the industry. Before 2000 the Office of Electricity Regulation (OFFER) regulated the electricity industry. OFFER had dual responsibility for market regulation and consumer protection. However, the Utilities Act 2000 split these responsibilities and amalgamated electricity market regulation with that of gas. The Office of Gas and Electricity Markets (Ofgem) carries out these regulatory duties. One further change was the separation of electricity supply and distribution; the effect of this change was to remove the requirement that the entity supplying electricity to a consumer and the entity distributing it be the same.

With the success of NETA a decision was taken to expand the mechanism to include the Scottish network. As a result the Energy Act 2004 (UK Government, 2004) contained the British Electricity Trading and Transmission Arrangements (BETTA) that were introduced on 1 April 2005. There was little change in terms of trading when migrating from NETA to BETTA, other than the inclusion of Scotland within the BM and the licensing of a single SO for the entire GB system. Today the single GB SO is National Grid Electricity Transmission (NGET) and the three transmission licence holders are: NGET, Scottish Power and Scottish Hydro. NGET does not own the transmission networks in Scotland.

BETTA, and NETA before it, employs two trading systems: bilateral agreements and the BM. The BM is the mechanism by which the supply and demand of electricity is balanced on a day-by-day, hour-by-hour basis. NGET is responsible for maintaining system stability and, therefore, for balancing the system in real-time. All parties trading electricity on the GB system are required to adhere to the Balancing and Settlement Code (BSC) that defines how the BM and imbalance settlements will work. The BSC is implemented on behalf of the industry by ELEXON (see www.elexon.co.uk) (Khan, 2009). ELEXON is a company established by the industry under the BSC as the sole operator of the systems through which all energy trading, balancing and settlement takes place. All entities engaging in electrical energy trading via the GB transmission and distribution system are required to sign the BSC and operate according to its rules.

The BSC is a complex mechanism. The following provides a summary of the operation of the mechanism drawn from information available from ELEXON (ELEXON, 2012; Khan, 2009). For the purpose of managing the system and settlement each day is split into 48 half-hour slots termed *settlement periods*. A settlement period is the basic accounting block of the BM. All trades are arranged around the settlement period and settlement – i.e. how much each party contracted for, used and therefore any difference between the two – is calculated and processed by ELEXON.

Trades are generally arranged in advance for each settlement period via bilateral agreements that are negotiated between parties. These can be renegotiated any number of times but must be fixed one hour before the settlement period starts. This point is termed *gate closure*. The parties are required to notify the SO of the volume of energy traded and the relevant settlement periods, but not the financial value, prior to gate closure.

During the settlement period the system should balance according to the trades that were fixed prior to gate closure, however in reality this rarely happens, as forecasts are never perfect and system outages do occur. Fine balancing of the system in real-time is the role of the SO. Within each settlement period the BM is used to balance the system by ensuring that supply and demand are continually matched. Condition C16 of the transmission licence granted to National Grid Company contains the balancing obligations (Ofgem, 2009). To satisfy these obligations the SO needs to ensure that there is sufficient headroom on generators which are on-line to cope with second-by-second changes and also that there is sufficient capacity held in reserve to meet both forecast and un-forecast demand.

Balancing services are procured in advance by the SO and are then used within each settlement period as required. In addition to the real-time second-by-second variation provided by generating units, which aim to keep the system stable at precisely 50 Hz, some Generators offer to either generate extra electricity or reduce generation upon instruction by the SO. Some Suppliers also offer to either increase or reduce demand as required by the SO. Generators and Suppliers willing to be flexible in this way make bids and offers into the BM prior to gate closure. An offer is a proposal to increase generation or reduce demand, whilst a bid is a proposal to reduce generation or increase demand. These are additional to any trades fixed outside of the BM. The value of bids and offers are set by the Generators and Suppliers and the SO can choose from these bids and offers according to its requirements with the overall aim of minimising the cost of balancing.

At the end of each settlement period total volumes supplied and consumed are accounted for and settlement accounts updated accordingly. Those Generators who contracted to supply a certain volume but were unable to do so have to pay penalty charges for the energy they were unable to supply. This energy was procured from spare capacity available from other Generators. These other Generators are compensated for this energy. Similarly Suppliers who consumed more energy than they contracted for will pay a penalty charge associated with this extra energy. In this way both Generators and Suppliers are incentivised to provide accurate forecasts.

A Generator that generated more energy than it was contracted to provide will be paid for the extra energy and a retailer that offered not to use all the energy it was contracted for will be paid for that which it did not use. Energy bought is purchased at the System Buy Price (SBP) and energy sold is sold at the System Sell Price (SSP). Depending upon whether overall there was more energy consumed than had been expected (i.e. the market was 'short') or whether overall there was less energy consumed than had been expected (i.e. the market was 'long') the SBP or the SSP, respectively, dominates. The SBP and SSP are dynamic and are functions of the total energy traded and the cost of it, which will vary according to supply and demand, and also the costs of generation and transmission.

The SBP and SSP are always different. The SBP is at a premium to the system spot price, i.e. the average market price, and the SSP is at a discount to the system spot price. Thus those Generators or Suppliers who are frequently in a position where they are in deficit will pay the largest penalties. Those in excess will be paid for the additional energy produced or not consumed, but the unit price (the SSP) will be lower than for those who are penalised, who are paying the SBP. The use of differential pricing often leads to a financial surplus over the entire day. This surplus is divided amongst all participants in proportion to their total traded volumes.

NETA was regarded as having had success in driving down the cost of energy to the consumer (Tovey, 2004). This success led directly to the decision to incorporate the Scottish system through the establishment of BETTA. However, like the Pool before it there have been criticisms. One major criticism was that the nature of the penalty mechanism is such that it favours traditional generating companies and electricity suppliers. For example, the mechanism actively works against renewable generators, as they will tend to generate in excess if, say, the wind speed is higher than expected. For this extra energy they will receive the SSP. In contrast if the wind drops and they are unable to meet their obligations they are required to purchase electricity at the SBP. Thus, they receive a proportionally lower reward for overproduction compared to the penalty for underproduction. The uncertainty in energy production, coupled with the differential pricing, means they are subject to proportionally higher financial risk and are incentivised to underestimate production to mitigate against having to buy energy even though the result is that overproduction will be in effect under rewarded. Further, because their volumes are small they get a disproportionately lower share of any daily surplus even though the nature of their operation means their contribution is likely to be disproportionately large. The system works in favour of larger players who are able to tightly control their generation or supply. Clearly large conventional generation plant is ideally suited to this task. This situation is explored in detail in two position papers provided to Ofgem (Cornwall, 2007; Littlechild, 2007). This aspect of the operation of the system demonstrates how it is structured around the conventional predict and provide model and assumes that generation has an expenditure model like that of conventional thermal plant.

3.3 A Political Barrier to FD Services

The preceding analysis has identified the tight coupling between policy and the structure of the GB electricity industry. This raises the question as to whether there may be a political bias against FD. Some authors have already pondered the barriers to the implementation of FD type services. Strbac (2008) considers a lack of Information and Communications Technology (ICT) infrastructure, a lack of understanding of the benefits (not just by consumers) and increased complexity of system operation to be crucial. Issues around ICT and system complexity could be regarded as technical and not necessarily structural. However, these technical issues do have a material impact on policy as they influence the evidence provided by industry to policy makers. The European research project ADDRESS (see www.addressfp7.org) has also sought to identify barriers to implementation. The ADDRESS project explicitly considers what barriers there might be to implementation. It cites the following as major sources of risk to implementation and, hence, potential barriers (Belhomme et al., 2009):

- Consumer acceptance
- The regulatory framework (particularly in terms of incentives)
- Commercial / contractual issues
- Conflicts of interest
- Pricing model
- Billing and information management

As such it does consider policy in the form of regulation. However, neither Strbac nor ADDRESS offers any explanation as to why there seem to be few incentives for implementation. Incentives flow from government in response to decisions taken to pursue particular policies. For example, the UK Government provided in the Energy Act 2008 (UK Government, 2008b) for a mechanism of Feed-in Tariffs to incentivise the implementation of small-scale (up to 5 MW) low-carbon generation. The Energy Act 2008 also includes obligations in relation to the deployment of smart meters but does not explicitly provide for incentives in relation to FD type services.

Earlier it was established that the industrial structure of the GB electricity industry closely follows government policy. The network of relationships within the industry, and the operating paradigm, evolve to reflect that policy. The inability to monitor and control the millions of loads on the system was a technical limitation that forced a "predict and provide" mode of operation and this is embedded within the policy and regulation, underpinning the structure and operation of energy markets. Even through the considerable upheaval resulting from privatisation and the move away from central planning to markets the system has been successful in maintaining very high levels of reliability in the supply of electricity to consumers.

The implementation of renewables presents the first real challenge to the "predict and provide" model. FD can be part of the solution but it means changing the way we generate and control electricity consumption. It involves a change to the way the system is operated and brings renewed fears over system reliability as well as the unknown social impact that the management of demand might bring. Beyond the desire to increase the proportion of low carbon generation there is not a clear political will to force through change in the operating model to support the participation of the demand side. At the same time the uncertain impact on established industry business models is likely to raise concern within the collectively dominant existing industry players. With no clear evidence that FD will deliver what it promises it is easy to see why a conservative view might dominate and FD be relegated to an interesting but not too serious option: it might be perceived to be better to follow a strategy of building large, centralised plant – be it generation or storage – under the control of the industry and continue to "predict and provide". Such a strategy lies within the industrial comfort zone having readily understandable and quantifiable risk. However, it also entails substantial cost to the consumer.

There is, therefore, the potential for a "cycle of inaction" as shown in Figure 12. The lack of demonstrators able to show the technical and commercial viability of FD services leads to a perception of risk within political and industrial stakeholders. Without a credible demonstration the services are judged too high risk to warrant serious investment and consideration as a major component of the BM. This risk leads to a lack of political will to back the services via appropriate incentives. Since the structure of the industry is so tightly coupled to legislation and regulation, the lack of incentives means that legislation does not include appropriate support for the removal of regulatory – and structural – barriers that would support the required transformation of the operating paradigm. The persistence of a structure that embeds the "predict and provide" model acts to prevent the effective implementation of commercial scale FD services and, hence, commercial scale demonstration projects.

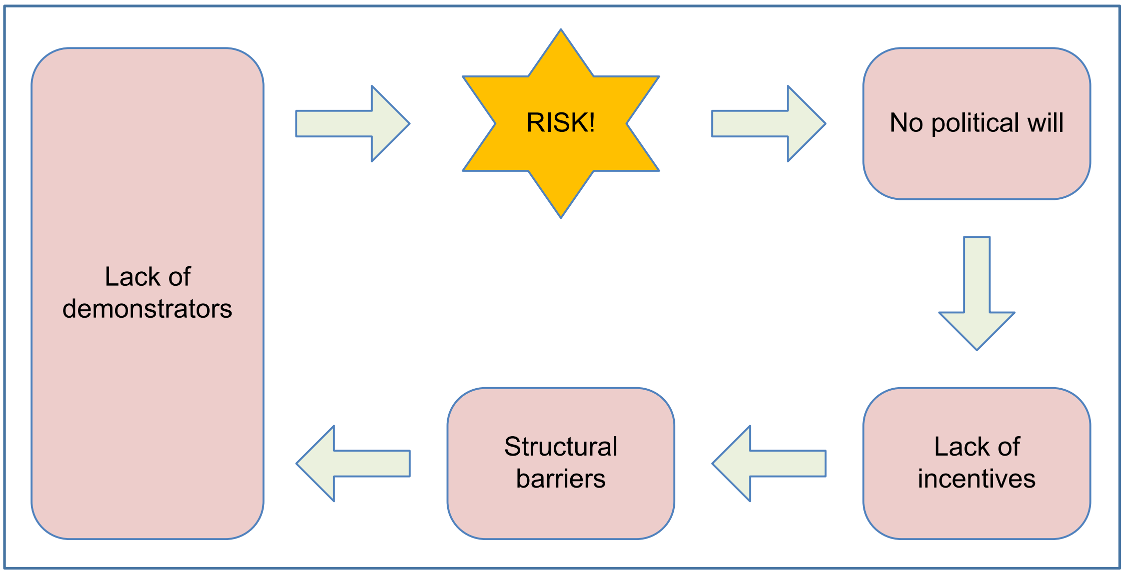

Figure 12: the political-industrial cycle of inaction.

To break this cycle it is necessary to reduce the risk associated with this technology, which can only be done through demonstrating its potential. Thus there is a need to demonstrate commercial-grade technology but within the existing operating paradigm. This is precisely the position wind generation was in until climate change provided the needed political will to produce the incentives needed to convert it into a mainstream technology[11].

There is also value in gaining a greater understanding of structural barriers. Once the technology has been demonstrated in such a way as to demonstrably reduce the associated risk then knowledge of structural barriers can be used to design appropriate policies. Such policies should be aimed at eliminating or reducing barriers to implementation. To address these two aspects – technology demonstration and barrier identification – the remainder of this thesis describes

[11] Incentives for wind generation were introduced via the Non-Fossil Fuel Obligation in 1990 (Electricity Act 1989) and the Renewables Obligation in 2002 (Utilities Act 2000).

research conducted into a system for demonstrating the feasibility of commercial-scale FD services for system balancing and a mechanism for understanding structural barriers within industrial systems.

3.4 Summary and Conclusions

Through a historical analysis this chapter has illustrated the tight coupling between energy policy and the structure of the electrical power generation and distribution industry within Great Britain. A key conclusion of the analysis of the evolution of the industry described in this chapter is that the industrial structure we see today is largely a result of technical constraints and ideologically driven economic policy enacted via primary legislation. Both UK and European policy makers and academicians have informed this policy. The aims of macro-economic policy have been to provide electrical energy to consumers reliably and affordably, whilst engaging private industry to promote innovation. The introduction of NETA and then of BETTA have been highly effective in achieving these aims.

As discussed in the previous chapter, however, climate change mitigation through a reduction in GHG emissions requires a fundamental change to the GB power system. Prioritising cost reduction is at odds with climate change mitigation. Policy makers are therefore presented with a dilemma. On the one hand there is a desire to de-carbonise energy production whilst on the other is the desire to protect consumers – who have become accustomed to relatively cheap energy – from extreme price rises. De-carbonisation under the conventional operating paradigm can only be accomplished with substantial capital investment, which inevitably means price rises. Consumers are also voters and few politicians are keen to force steep energy price rises onto consumers.

In conjunction with other low-carbon technologies FD offers the possibility of meeting de-carbonisation goals whilst minimising cost but because it challenges the existing operating paradigm it is perceived as entailing risk. The tight coupling between policy and the structure of the industry means that change is extremely difficult without political will and that will is either absent or, at most, ambivalent whilst the perception of risk remains. We are locked into a "cycle of inaction".

In Chapter 2 a broad industrial view was presented. This chapter has built on that by providing a historical perspective. Together they have allowed the identification of a perception of risk as the principal cause of inaction in respect of FD services. To address this issue of risk a demonstration of the viability of FD as a core balancing service, and identification of structural barriers to the implementation of FD is needed. In the remaining chapters these two issues will be examined as summarised in Figure 13 below.

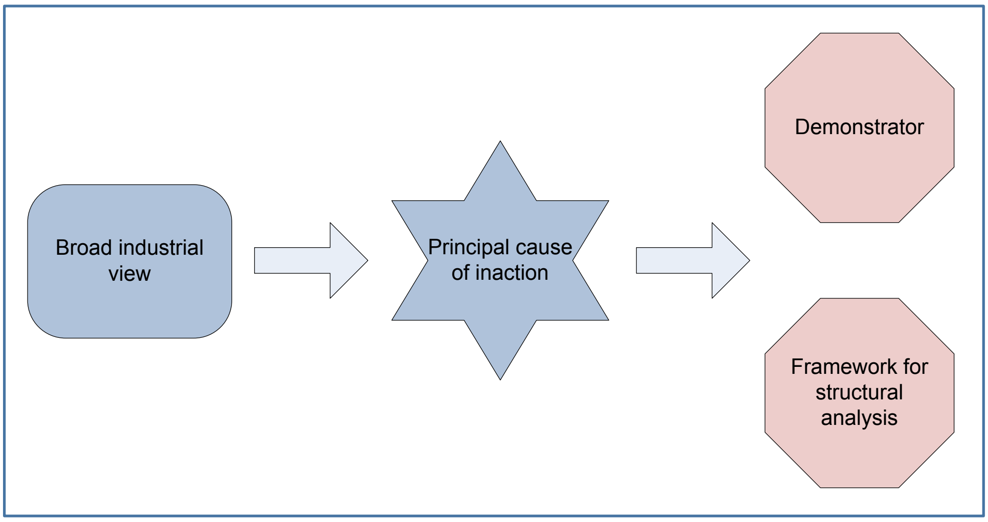

Figure 13: research outcomes.

Chapter 4

A System for the Demonstration of FD Services

4.1 Introduction

In the previous chapter the need to enlist the support of policy makers when introducing new energy technologies such as FD was identified. It was further identified that politicians are reluctant to provide such support whilst there might be a perception of risk arising from the impact the introduction of a technology such as FD might have. One mechanism for overcoming this reluctance is to demonstrate the viability of the technology. In this chapter a system for such a demonstration of FD as a balancing service provider is described. This is based on research undertaken within the scope of the DD-FD project as described in Chapter 1. First a high-level overview of the system is provided before proceeding to describe in detail the author’s principal contribution around the research and development of the core scheduling and dispatch functionality.

For clarity a number of terms are used within the context of the FD system that have specific meaning. These are defined on the next page.

Definitions

End user: an entity (e.g. a consumer) that uses electrical power in some appliance or process.

Load: the energy consuming appliance or process whose consumption is modified to provide the FD service.

Node: a group of one or more loads managed as a single provider of flexibility.

Offer: an offer of flexibility made by a node.

Control Centre: a collection of functions to support the reception and processing of flexibility offers.

Aggregator: an entity using the facilities of the Control Centre to offer balancing services based on aggregated offers from nodes.

4.2 FD Demonstration System Architecture

The FD system concept is based on the concept that an Aggregator receives offers of flexibility from participating nodes, collates these offers into tradable packages and then offers balancing services based on these packages. It follows that the Aggregator must possess some reception, collation and dispatch functionality. In this architecture this functionality is centralised and embodied in a Control Centre under the management of the Aggregator. Centralising the functionality aids development and management.

A communications network over which a management protocol is operated facilitates communications with the participating nodes. This protocol was designed by members of the DD-FD team, including the author, as part of the project and provides functions to support the transmission of offers and dispatch commands together with a number of other supervisory functions. This protocol was not, however, a deliverable of the research described in this thesis and is therefore not described further. The physical architecture of the system is shown in Figure 14.

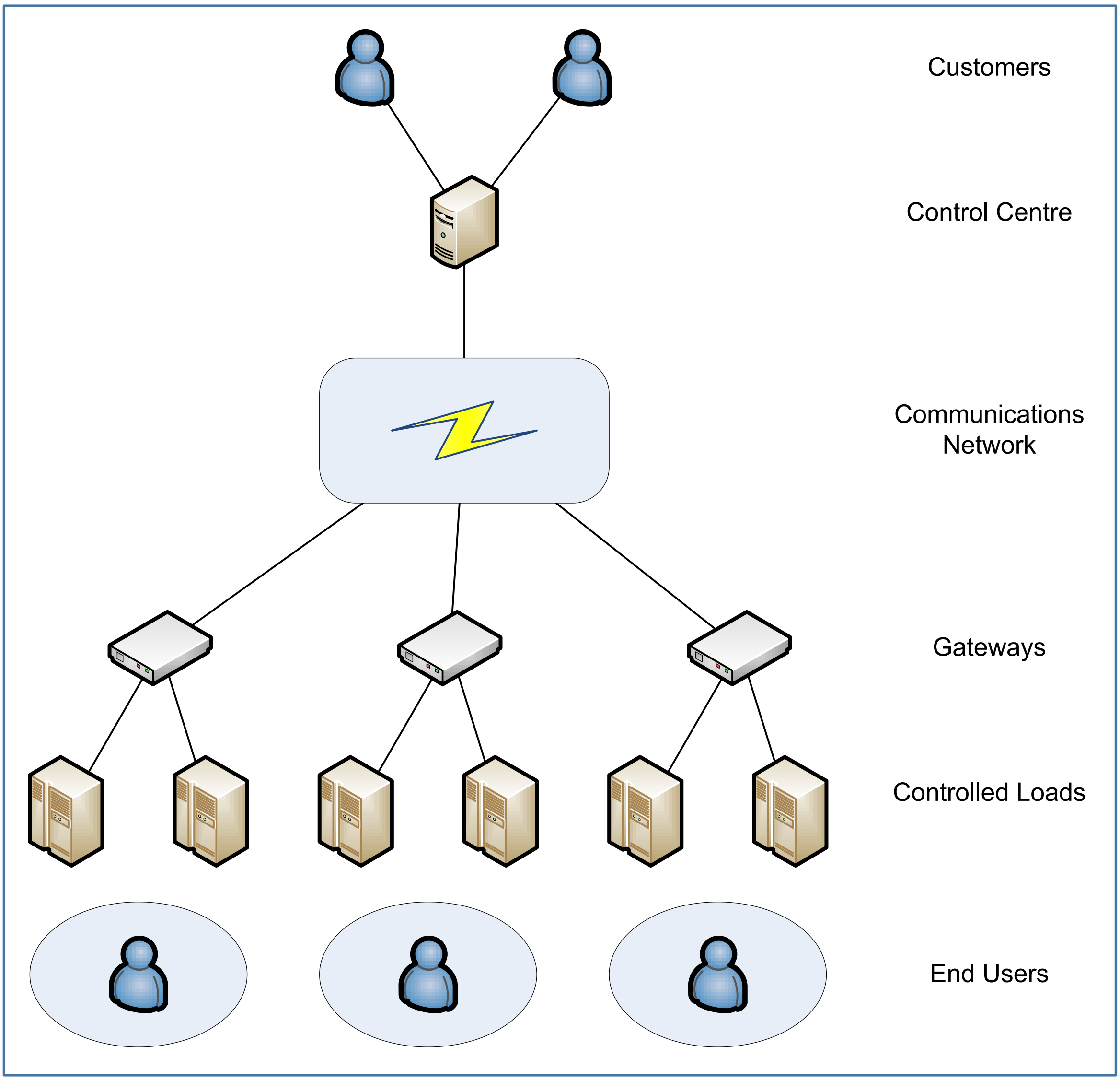

Figure 14: FD demonstration system architecture.

4.3 Design of the Demonstration System

The starting point for the design of the system is a number of high-level system requirements that are aimed at creating an FD service capable of demonstrating its technical feasibility as a technology for providing balancing services. These requirements were developed in collaboration with other members of the DD-FD team at a requirements gathering workshop run by the author. In broad terms the system envisaged here should ensure that the End User continues to receive a level of service from the controlled Load within the bounds of what is acceptable to them, whilst still reliably providing the capability to manage power consumption.

To further refine the requirements the author analysed the requirements using Viewpoint Analysis (Burge, 2008). The result of this analysis is shown in Figure 15.

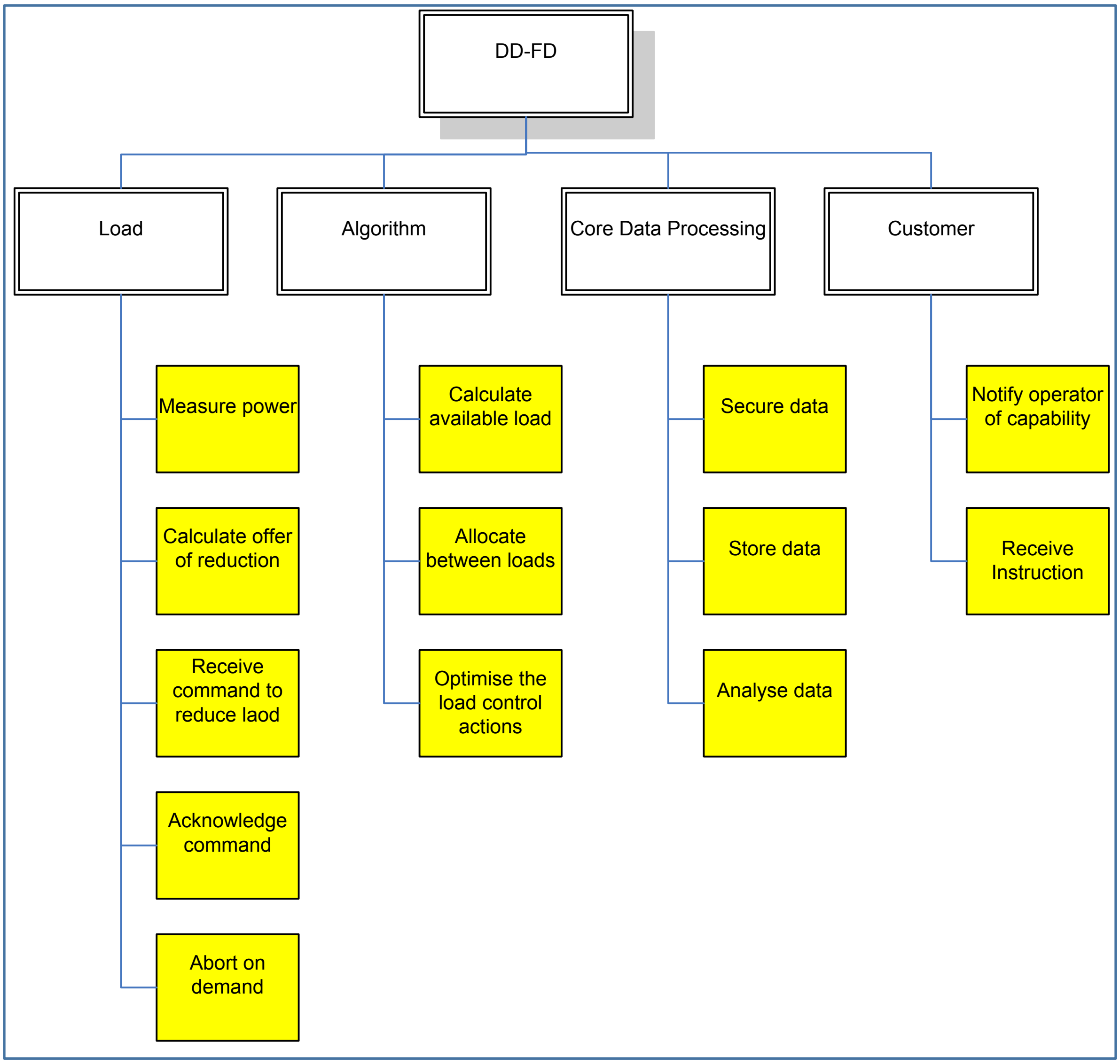

Figure 15: viewpoint analysis of DD-FD system requirements.

From the Viewpoint Analysis a number of non-functional and functional system level requirements were identified. These are shown in Table 6 and Table 7 respectively together with the reasoning behind their inclusion.

Table 6: system level non-functional requirements for the FD demonstrator.

Non-Functional Requirement	Reasoning
Reliable	To be useful as a balancing service any purchaser of flexibility must be able to rely on the service providing what has been contracted. If it is not reliable then it must be backed up with some other balancing service, which will undermine its commercial viability.
Use open standards	To ensure that third parties can design controllable loads with minimal commercial barriers thereby increasing the available portfolio of suitable loads.
Independent of load type	Should be able to control any load (or mix of loads) capable of having their power consumption modified irrespective of what it is.
Real-time availability	It must be available to be used when required without restrictions on when the service is available (linked to reliability).
Secure	Participants and customers must be able to trust that the system will not expose their own systems to hostile third parties.
Scalable	The overhead of adding new loads (irrespective of what they are) must be minimal to maintain viability.

Table 7: system level functional requirements for the FD demonstrator.

Functional Requirement	Reasoning
Calculate available load	The demonstration system needs to regularly update its view of the available load to present an accurate picture.
Notify operator/customer of capability	The current capability is the basis of offers to flexibility customers.
Receive commands to initiate an event	To support the receipt of customer commands to initiate a load-shifting event.
Dispatch loads to meet aggregate requirement	The Control Centre must send many dispatch orders in a coordinated fashion to ensure the aggregate behaviour is as contracted.
End user retains facility to abort or opt out	Customers must not feel they are being managed, rather that they retain a level of control.

The non-functional requirements for scalability and independence of load type have a particular bearing on system design. The former implies that the level of per node processing required in the Control Centre must be minimised. If the Control Centre is required to model the behaviour of each load in order to calculate or predict offers then it is unlikely to be scalable beyond a small number of nodes. The latter implies that the Control Centre must view each load as a "black box" with very little information about its internal workings. This has the advantage that if the composition of the node changes (e.g. new loads are added) it does not mean that the representation of the node within the Control Centre needs to be changed, only the values of the parameters that describe it.

These two requirements together mean that the Control Centre has very little information about the internal functions of the nodes and has to work without the ability to accurately predict the behaviour of the nodes under management. Essentially the only information available to the Control Centre relates to externally visible aspects including where it is and how much power it is, or has been, consuming. It may also be possible to gather certain data relating to the environment in which the node is operating, such as ambient temperature. Since the operating paradigm requires nodes to make offers (see Figure 15) it is possible to infer a list of parameters that might be available from each node:

- Location of the node
- Current power consumption
- Offer of consumption adjustment
- Offer validity period
- Period for which the adjustment can be maintained

The variables are essentially power, time and geography. It would also be possible for the node data to include a price, if a market-based decision mechanism were to be employed. However, that is not the approach taken here.

In this system each of the nodes under management periodically issue an offer to adjust their power consumption at a set time and for a defined period but the consumer ultimately retains control (see the abort requirement in Table 7). There is no guarantee that when asked to do execute they will comply. As described above, to maintain scalability the Control Centre knows little about the nature of the nodes under management. This implies that the nodes themselves must have sufficient "intelligence" to calculate offers and must therefore embed knowledge about the behaviour of the loads that constitute the node. This is logical since the node, not the Control Centre, is in the best position to relate environmental and other data to a predicted consumption pattern and to calculate permissible adjustments based on an understanding of the load operation and any locally defined constraints. These design choices are summarised in Table 8.

Table 8: summary of design choices for the FD system.

Characteristic	Requirement	Reason
Node knowledge	Generic communications protocol between the node and the Control Centre with no node-specific knowledge	Scalability
Data transfer	Power curve	Support a generic interface; only interested in power, not price
Operating paradigm	Accept or reject offer from the node	Only the node knows its capabilities and the degree to which service can be affected

As stated above, the Control Centre is responsible for receiving offers, aggregating offers into packages that can be used for balancing services and then controlling the dispatch of nodes once a request is made to execute an aggregated offer. Of these functions the scheduling and dispatch of loads is critical to meeting the system requirements.

The design of the system means that an Aggregator can never be sure that when a node is called upon to change its power consumption it will actually do it. Without a detailed understanding of the relationship between power consumption and environmental factors at the node the Aggregator must rely on previous experience to build a heuristic model of the node in order to predict its reliability. Given the uncertainty this entails it is important that the Aggregator has access to a sufficiently diverse load base to ensure that it will always be able to meet it's customers requirements. The dispatch algorithm therefore becomes crucial in this operating approach.

4.4 Scheduling and Dispatch

The principal task facing any commercial Aggregator is converting an order for demand modification into a series of dispatch orders that might be sent to the various loads under the Aggregator's control. This is essentially a scheduling problem and an objective of any commercial Aggregator must be to satisfy the SO's order whilst minimising the Aggregator's own costs. The Aggregator must do this in a reliable manner such that the SO views it as a trusted provider of essential balancing services. Without that trust it is unlikely that the SO will routinely use the Aggregator, instead opting for providers whom it knows will provide the required consumption change when required to do so.

The profitability of FD services relies on the difference between the revenue earned by selling balancing services and the Aggregator's own costs. These costs comprise the operational costs associated with the information systems and network, and payments to flexibility providers, i.e. those parties that consent to the modification of their consumption. There are no fuel costs. It is expected that out-payments to flexibility providers will form a significant component of the costs. The optimisation of dispatch must therefore seek to minimise costs. An effective dispatch algorithm will be a key differentiator against other Aggregators. The job of this algorithm is complicated by the limited information available to the Control Centre about each controlled node. This results from the requirement to hide the detailed operation of the loads. The only information that the Control Centre has access to is the offers sent by the node (both current and historic) and any other information about the node that it holds or has access to (e.g. location).

The dispatch algorithm is central to the effective discharge by the Control Centre of its functions in response to commands from the Aggregator. The algorithm must be able to select from the pool of available flexibility resource the optimal combination of nodes that when instructed to act in concert will produce the required aggregate effect. The importance of this aspect of the FD system means that it is the scheduling and dispatch algorithm that forms the focus for this research.

4.5 Scheduling and Dispatch Requirements

The preceding discussion has covered a number of system level requirements. In this section requirements specific to the scheduling and dispatch function of the FD demonstration system are discussed.

In this system the Control Centre does not seek to manage the consumption of nodes through manipulation of their operation. The negotiation takes place via the offer from the node rather than in response to a request to reduce consumption. Rules for avoiding performance degradation are embedded within FD nodes and are a factor in determining the offer made.

In systems that hide the node behaviour some employ an abstract parameter such as price as the basis of negotiation with the node, and others propagate a constraint to the node such as maximum power consumption. Those that model behaviour do so using empirical data or simulation to determine what the node can be requested to do, and still remain within whatever operational bounds have been set. These alternatives are summarised in Figure 16.

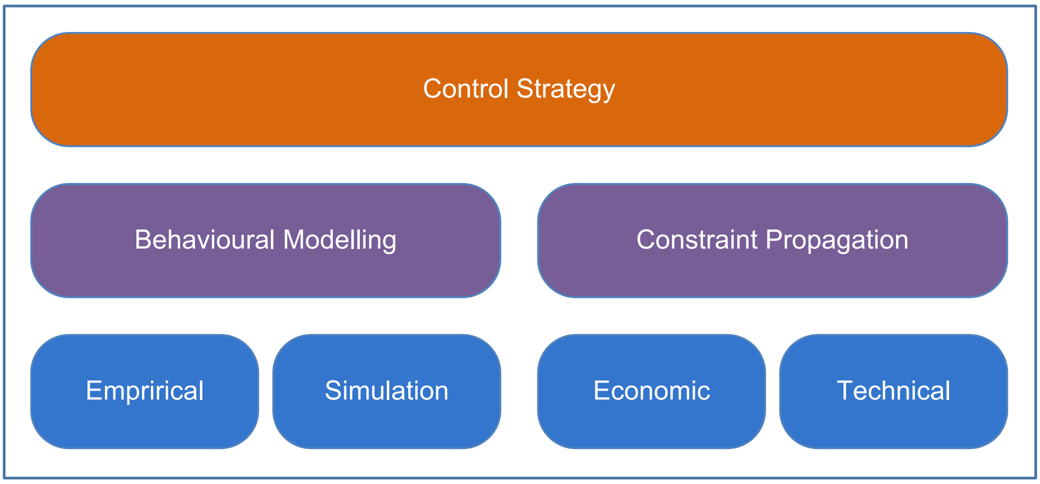

Figure 16: alternatives for FD control strategies.

None of these approaches meet the requirements of the FD architecture under research here. The use of a power constraint propagated to the node, or a pricing signal, is counter to the envisaged mode of operation of the FD system. In the FD system the offer is propagated from the node to the Control Centre, which can then choose whether to accept it or not. This ensures the requirement that ultimate control remains at the node. Some systems do make an offer to a load manager, but this is often an economic offer in terms of an offer to reduce consumption at a specified price. Such systems envisage a market for demand response in which each load may participate no matter how small its offer. Whilst this may one day be the case, current arrangements for the purchase of balancing services preclude the direct participation of small providers. STOR, for example, has a minimum requirement of 3 MW (see Section 2.7). To operate in such markets necessitates the role of an Aggregator. Finally, such approaches require the offer algorithm at the node to be more complex as it must determine not only power and time, but also the price at which it is willing to sell its flexibility.

The need to maintain a "black box" view of the nodes means that it is not appropriate to employ within the FD Control Centre behavioural modelling relying on detailed knowledge of the node under management. This does not exclude the

Control Centre from predicting behaviour based on past performance but such predictions must be restricted to using only data gathered in the normal course of operation, i.e. via the offer and reporting mechanism. The requirements of the scheduler implicitly define the requirements for the design of an appropriate offer structure and the required node parameters. The requirements are summarised in Table 9.

To improve delivery accuracy the scheduling and dispatch sub-system must learn from the past behaviour of nodes. Results from the DD-FD testing programme showed that the delivery by individual nodes rarely matches that offered. Accuracy can be improved at individual nodes by enhancing the offer calculation algorithms embedded within the gateways (see Figure 14) and/or load controllers. However, such algorithms may not be within the scope of control of the Aggregator. The Control Centre should not, therefore, assume that offers are accurate. Similarly, the configuration of appliances at a node may change without the embedded algorithm within the gateway or configuration of the gateway being changed to reflect this, resulting in potentially inaccurate offers. Although commercial arrangements between the Aggregator and the flexibility provider should incentivise accurate offers there is still a risk to the Aggregator attached to relying on offers as an accurate prediction of delivery. It is in the interest of providers of aggregate FD services to ensure that the Control Centre maintains an accurate reference of the performance of each node. Thus the recording of node performance is important. Also, the algorithm used to determine the dispatch sequence must include the capability to base its decision-making on a reliability parameter that reflects the relevant node's past performance.

Table 9: scheduler requirements.

<table>
<tr><th colspan="4">Requirements</th><th>Comments</th></tr>
<tr><td colspan="4">Context: The scheduler is the processing module within the Control Centre responsible for determining the optimum dispatch sequence in order to meet a specific customer request.</td><td></td></tr>
<tr><td colspan="4">Operational Requirement: Create the optimum dispatch sequence of nodes in order to meet a specific customer request.</td><td></td></tr>
<tr><td colspan="4">Non-functional System Requirements</td><td></td></tr>
<tr><td>RS1</td><td colspan="3">The system should be able to dispatch nodes with minimal latency following an instruction to initiate an event</td><td>Will determine system response time</td></tr>
<tr><td></td><td>Functional</td><td>Performance</td><td>Implementation</td><td></td></tr>
<tr><td>R1</td><td>Calculate optimal dispatch sequence</td><td></td><td>Combinatorial search</td><td></td></tr>
<tr><td>R2</td><td>Consider power, reliability, cost and (possibly) geography</td><td></td><td></td><td>Will need some weighting factors</td></tr>
<tr><td>R3</td><td>Receive node offer in terms of power and time</td><td></td><td>Multiple offers per node</td><td></td></tr>
<tr><td>R4</td><td>Learn from past behaviour</td><td></td><td></td><td></td></tr>
</table>

Requirement R3 has an implementation requirement specifying support for multiple offers from the same node. In the system described here the offers can be different powers for different periods but all offers are assumed to represent a rectangular response from the node as shown in Figure 17.

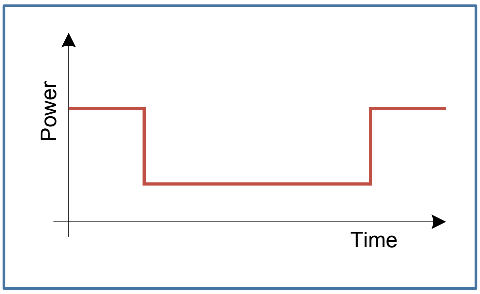

Figure 17: simple rectangular consumption curve.

4.6 Summary and Conclusions

This chapter has provided an overview of the requirements and architecture of a system for demonstrating the technical feasibility of using aggregated flexible demand to provide balancing services, such as STOR, to the SO. The architecture is designed considering a number of key requirements chosen to ensure the scalability and reliability of the system. In particular the architecture assumes that the Control Centre, which performs aggregation and dispatch, knows little about the nodes providing flexibility. This is a deliberate design choice to avoid the need to model the behaviour of the managed loads within the Control Centre. Such an approach supports scalability.

Each node within the system can be an arbitrary collection of heterogeneous loads. The interface with the Control Centre is via a gateway whose task is to calculate the available response based on local knowledge such as the load configuration and behaviour, and on local environmental conditions and constraints. Algorithms based at the node are best placed to calculate what flexibility can be provided. The operating paradigm is that the Control Centre does not have full control over the nodes; rather the nodes make offers of flexibility that the Control Centre can elect to accept or disregard.

This mode of operation means that the algorithm within the Control Centre responsible for choosing which nodes to use has a key role. This algorithm must optimise the choice of nodes based on the requirement from the SO, the offers available, the cost of using certain nodes and the historic reliability of nodes to deliver what they have offered. It is the design of this algorithm that forms the focus of the next chapter.

Chapter 5

An Algorithm for the Dispatch of Loads in a FD System

5.1 A Review of Literature on Optimised Scheduling

5.1.1 Scheduling Approaches within Demand Response

Optimal dispatch is a classic scheduling problem (Amato et al., 2007). The objective is to determine the best dispatch strategy given the aggregate requirement and the characteristics of the individual nodes. There are a number of ways of classifying demand response scheduling approaches. It is possible to categorise them by the scope of control, the basis on which decisions are made or the mechanism used to arrive at an optimal solution (measured according to some predefined criteria). Within the context of demand response, the scope of control refers to whether an end user-oriented viewpoint or a network-oriented viewpoint is assumed. In end user-oriented systems the aim is to control the demand to remain within some threshold such as power, energy or cost. The active loads are typically within the same building or site, as is the controller. Network-oriented systems aim to produce a defined response from each consumption node such as a building. In this case the controller / scheduler is located within the network.

From the end user side the scheduling problem is frequently one of controlling on-site loads in order to ensure that tariff constraints are not exceeded. Babu and Divya (2008) applied such an approach to the Nandi milk industry in India with the objective of reducing instances where the site load exceeds that permitted under the tariff. Calabrese et al. (2007) investigated a similar problem in a domestic setting in Italy where the power draw by a domestic property is limited and if it is exceeded the power is interrupted. In both papers artificial intelligence (AI) techniques based on an Artificial Neural Network (ANN) were applied to exert direct control over specific appliances. The objective being to avoid the need to simulate the behaviour of the loads and instead produce a probabilistic response by training the ANN to respond to different patterns of consumption.

Moura and Almeida (2010) considered the problem from the point of view of the SO within the context of a wider system balancing approach. In their paper they use multi-variable optimisation to determine the optimal generation mix given predicted climate conditions. Similarly, Aalami et al. (2008) use multi-criteria decision making (MCDM) and multi-attribute decision making (MADM) to determine the best mix of demand response programmes to achieve a given goal. Although each paper has a different focus, it is interesting to note the common theme of making an optimal decision based on a number of often conflicting variables and criteria. In both cases the algorithm is able to dynamically respond to changing parameters in contrast to a simple, static merit order approach. Within the context of the research here it is clear that is the task that the scheduling algorithm must be designed to do.

These papers also represent what might be described as opposite ends of the spectrum of scope for this problem. In the former case the objective is to ensure that the power drawn from the supply to the house or factory remains within certain tariff-related limits. What happens on the network side of the meter is not of concern. In the latter case the objective is to optimally balance the transmission system; what happens within the distribution networks is out of scope. The FD system considered in this research actually sits between them. The objective of the system described here is to provide balancing services and to do so it needs to be aware of how individual nodes are behaving but not the detail of how they achieve it. It is concerned about the behaviour of individual consumption nodes but not about the behaviour of individual appliances. It is also concerned about the aggregate behaviour of the nodes under its control but it is not concerned about the overall behaviour of the electricity system. This focus is deliberate since the intention is to sell services into the balancing market based on flexibility provided by end users.

Decisions about scheduling include both economic and technical considerations. Economic considerations include minimising cost (such as incurred penalties or cheaper energy) or maximising profit (such as selling FD services when wholesale prices are highest). Technical considerations include achieving a desired demand curve through the selective curtailment, or increase, in demand. The objective is to achieve a desired demand curve in the most economically optimal way.

Economic approaches frequently adopt a market model in which nodes react to price signals from the market (e.g. Cruickshank, 2008; Deindl et al., 2008; Hui et al., 2004; Junji et al., 2003; Praca et al., 2008; Ruiz et al., 2009; Tang et al., 2005; Yu and Yu, 2007; Zhang and Feliachi, 2003). Deindl et al. (2008) use fuzzy logic agents embedded within software to automatically bid into a balancing market on behalf of the end user. In this application the agent offers to make an adjustment to its consumption at an offered price. This offer will be accepted or rejected by buyers in the market.

The use of software agents acting on behalf of the end user is also examined by Amato et al. (2007) and Calabrese et al. (2007) but in these applications the agents are used only to manage power and do not participate in a balancing market. In this case agents on individual loads make offers to adjust their consumption. This offer reflects the relative value the end user places on the service provided by the load. Whilst this is not an economic value in the sense of Deindl et al. it does include an implicit representation of value. In both cases agents are autonomous and provide an efficient way of representing the wishes of the end user without the end user needing to be directly involved in each transaction, thereby reducing the burden on the end user when participating in demand response programmes. The end user is able to pre-define their constraints and the agent acts accordingly. This is important, as end user apathy may be a significant barrier to participation.

5.1.2 Types of Optimisation Algorithm

Agents represent one decision-making technique. Whilst agents are useful in representing the end user in competitive energy management strategies, they alone do not make decisions – they must employ an algorithm to do so. There are a great many alternative mathematical approaches to optimisation (Diwekar, 2008). A number of approaches that are reported in the literature can be largely divided into "traditional" approaches and the more recently developed "heuristic" approaches (see Figure 18).

Of the traditional approaches Belhomme et al. (2009) use Lagrange whilst Ramanathan and Vital (2008) employ dynamic programming. Both Ruiz et al. (2009) and Yu and Yu (2007) use linear programming. Linear programming and dynamic programming are very similar approaches although dynamic programming is best suited to multistage processes. In their paper Ramanathan and Vital optimise the dispatch of HVAC with the objective of minimising the disruption to users of the HVAC systems whilst meeting the load reduction objective. This is achieved through modelling the behaviour of the HVAC systems. Although similar to the problem here, as discussed earlier the FD architecture hides the behaviour of the loads themselves in order to preserve scalability and ensure that different types of loads can be incorporated, and so explicit modelling of the HVAC loads is not possible within this architecture. Ruiz et al. achieve a similar objective using linear programming. All of these approaches are classical optimisation in that they seek to optimise a goal variable through adjusting a number of linear control variables.

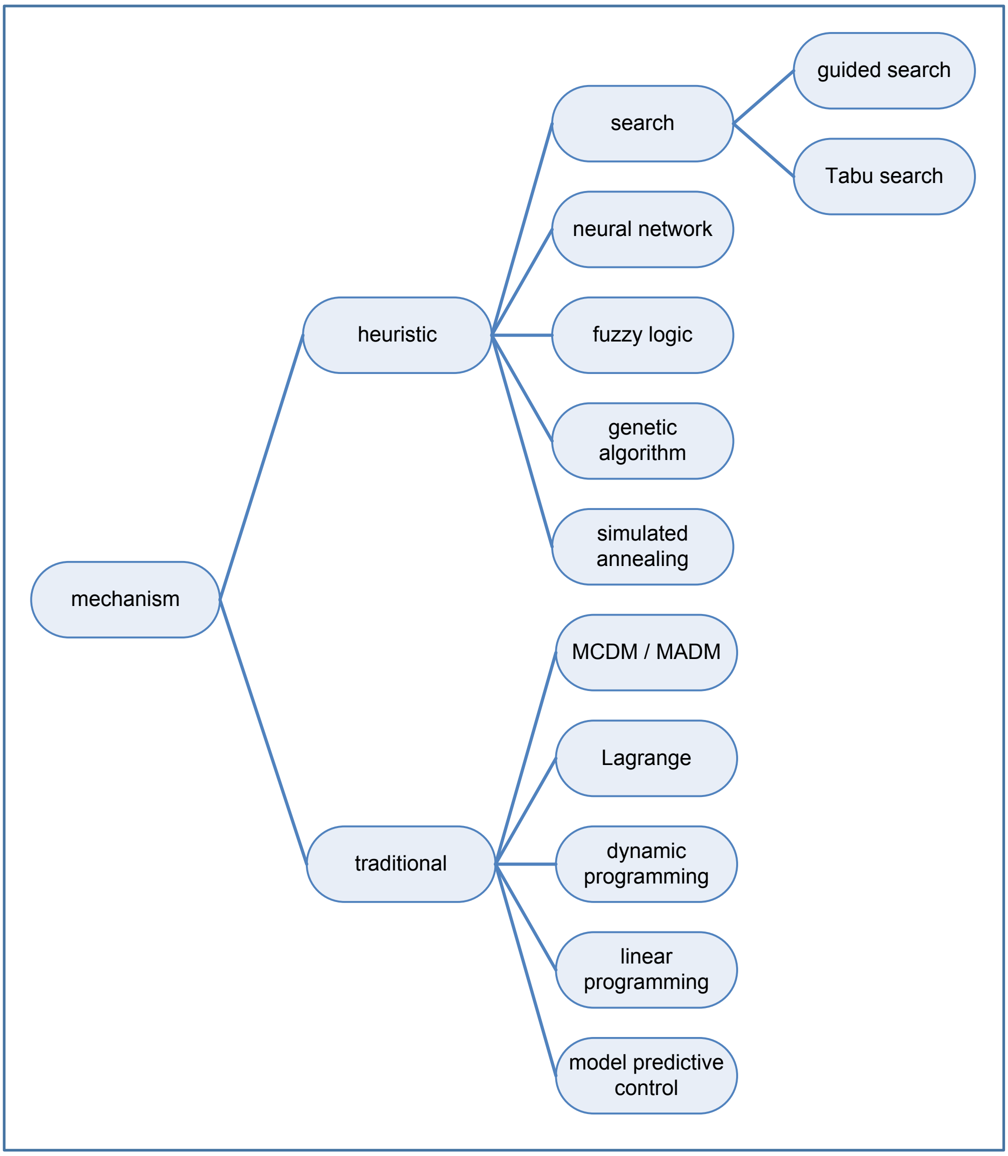

Figure 18: types of optimisation algorithm.

One final traditional approach is that of model predictive control as used by Stadler et al. for the control of domestic refrigerators (Stadler et al., 2009). In this approach a model of the device to be controlled is used to predict the aggregate behaviour of a large ensemble of devices under the control of an aggregator. Such an approach requires knowledge of the internal functionality of the loads and would not be readily transferrable to a heterogeneous mix of loads, as it would greatly impact scalability.

Of the heuristic approaches the application of Genetic Algorithms and Simulated Annealing to the load dispatch problem appears an under-researched area. They are, however, widely used to solve optimisation problems in other fields. Search techniques include guided search as used by Amato et al. (2007) and Tabu search (Vogel and Sonnenschein, 2007). Such approaches are useful for efficiently finding the optimal combination amongst a large population of possibilities.

5.1.3 Parameterisation of Nodes

The performance of an individual consumption node will be the summation of the performance of all the energy-consuming appliances within the node. In the simplest case the node may contain only one appliance but in some cases the node will contain a complex arrangement of heterogeneous appliances. The resulting aggregate performance must be reflected in a set of parameters that adequately describe how the node will respond should it be asked to participate in an FD event.

The treatment of heterogeneous mixes of appliances as a single energy-consuming node, whose consumption pattern is the combination of those of the individual appliances within the node, is an unusual problem within the context of demand response. To date most authors considering load performance prediction have concentrated on individual appliances in isolation from their environment. For example Calabrese et al. (2007) and Amato et al. (2007) both employ embedded load curves within the controlling agents to predict load behaviour. Stadler et al. (2009), however, do move some way to considering the behaviour of an "ensemble" of appliances. In this case the appliances are all of the same type – refrigerators – and the prediction is focused on the emergent behaviour of a number of appliances resulting from their simultaneous response to control signals, rather than their interaction with their environment. As different appliances will be at different points within their control cycle the emergent behaviour is not necessarily a simple summation.

Some authors have recognised that a practical load management system must cope with a heterogeneous load mix. Cascio et al. (2006) considers a mixed load scenario in the domestic setting and proposes that appliances include a behaviour description that a load manager might use to understand how a particular appliance might behave. Two classes of load are proposed: self-adaptive and pre-programmed. The self-adaptive appliances regulate their consumption within the limits specified by the load manager. This is done by negotiation in response to a request from the load manager to reduce power. This is similar to the offer concept adopted in this research. The pre-programmed appliances provide a detailed consumption profile that the load manager uses to micro-manage their consumption. This is somewhat similar to other authors' approach to understanding how loads consume energy, but in a much more generic way.

As with optimisation as described above, most authors either avoid understanding load behaviour, utilising an economic agent to parameterise performance (e.g. Kupzog, 2007), or assume a detailed knowledge of the underlying load behaviour in a manner similar to Cascio's pre-programmed appliances. The FD Control Centre will not know how a node uses the electrical energy it consumes. Information hiding is a key requirement to maintain scalability of the solution. Since the Control Centre does not know anything of the internal operation of a node the definition of its characteristics in terms of functional parameters is especially important. Some of these parameters must be provisioned within the Control Centre at commissioning and others will be exposed via the interface between the Control Centre and the nodes. Collectively these parameters must adequately describe the performance of the node such that it can be used in any scheduling calculation. The Control Centre treats each node as a "black box" and the chosen parameters and values must capture the complex behaviour of the node whilst under FD control. The chosen parameters must therefore be generic enough to be able to adequately describe any combination of loads that might constitute a single, controllable node.

5.2 Algorithm Selection

The literature review above considers in detail different optimisation algorithms. No one approach stands out as better, or worse, than the others on the basis of this literature review. Each has its merits and the choice of algorithm depends on the application in question. The traditional approaches of dynamic programming and linear programming offer a rigorous mathematical basis; however they do require the derivation of an objective function, which may not always be possible. In the case of the FD system this would require a function that produces some fitness measure as a function of the offers and node parameters. The mathematical relationship between the decision variables – in this case the offers – and the value of the objective function must be known. The decision variables are then varied to obtain the optimum solution. Some authors have employed these approaches, as they have been able to derive such a function by modelling the load behaviour and exercising direct control over the loads. However, a key requirement of the FD Control Centre is that it has no knowledge of how the nodes work and the interaction with the controlled nodes is via defined offers. As such the mathematical relationship between offers and objective function is unknown. It is therefore unlikely that a traditional approach is suitable for this application.

In the absence of a detailed mathematical knowledge of the node behaviour there are two approaches that can be adopted:

- An exhaustive search of the possible combinations to find the best fit.
- A guided search using a metaheuristic algorithm.

The exhaustive search is guaranteed to find the best solution but only at the expense of considering every possibility. A guided search will find an optimum but it cannot necessarily be guaranteed that the solution is a global optimum. An exhaustive search is clearly better but can only be considered if the problem is solvable using reasonable resources (i.e. processing power and time).

5.3 Problem Complexity

Requirement RS1 (see Table 9) requires the Control Centre to respond with minimal latency. The principal constraint on such a requirement is that of limited computational time. The scheduler will potentially need to compute the optimum solution within a matter of minutes to be able to provide the kind of fast response services identified in Chapter 2. The algorithm must also be scalable in that it must be able to scale to large numbers (possibly 100s or 1000s) of controllable nodes. This presents a classic optimisation problem: if the problem is a so-called "*NP*-hard" problem then the computational effort required by any algorithm in order to find a solution rises exponentially with the problem complexity (i.e. the number of combinations).

In complexity theory the concept of polynomial time is used to describe how the resources required for solving a problem scale with its complexity. Solvable in polynomial time means that the computational effort scales as some fixed power of the problem complexity (i.e. the number of symbols required to specify the input data) and can therefore be efficiently solved by adding extra processing power. The set of decision problems solvable in polynomial time is commonly referred to by the symbol P and this is a subset of all the possible problems including those not solvable in polynomial time, denoted as NP. The hardest NP problems to solve are referred to as NP-complete. It can be shown that if any polynomial time algorithm can be found to solve any NP-complete problem then $P = NP$ and, hence, all problems are solvable in polynomial time if only the right algorithm can be found. However, to date no such algorithm has been found and it is widely assumed that $P \neq NP$ (see Michiels et al., 2007 pp. 191-197). This has important implications. It means that if the problem considered here is NP-hard then no polynomial time algorithm exists and therefore the computational effort will scale exponentially with complexity.

The only way to prove that a problem is NP-hard is to prove that no algorithm capable of solving it in polynomial time exists. This is clearly not feasible. However, there are many similar problems that exist that have already been shown to be NP-complete. Perhaps the most famous of these is the travelling salesman problem in which the shortest route between all the cities within a specified group, visiting each city only once, is to be computed. For a small number of cities this is a tractable problem but the solution space grows exponentially with the number of cities: for 100 cities the number of possible solutions exceeds 10^{150}.

The solution space for the scheduler grows similarly. If there are M controllable nodes and each node makes F offers then the total number of offers, k, is

$$k = MF \tag{5.1}$$

The set of possible aggregation combinations comprises all possible combinations of offers from the k available offers, including combinations containing only one offer, the combination containing all offers, and any number of offers in between.

Note that it is combinations and not permutations that are relevant here. The number of members in the set of possible aggregation combinations, w, is therefore the sum of all possible combinations of any number of offers between 1 and k

$$w = \sum_{i=1}^{k} \frac{k!}{i!(k-i)!} \tag{5.2}$$

The DD-FD system was designed such that each node produces three offers so using equation (5.2) the size of the set of combinations for different numbers of nodes can be calculated. Some examples are shown in Table 10. It is clear that with any reasonable number of nodes the problem becomes rapidly unsolvable in polynomial time and is very likely a *NP*-hard problem. A guided search is therefore required.

Table 10: numbers of possible combinations
with different numbers of controllable nodes.

Nodes	Offers per Node	Size of Offer Pool	Number of Unique Combinations
1	3	3	7
3	3	9	511
5	3	15	32,767
10	3	30	~ 10^9
50	3	150	> 10^{45}

5.4 Metaheuristic Algorithms

The literature review earlier in this chapter identified the following candidate metaheuristic algorithms:

- Artificial Neural Network (ANN)
- Fuzzy logic
- Genetic Algorithm (GA)
- Simulated Annealing
- Tabu search

Each has advantages and disadvantages. ANNs, fuzzy logic, Simulated Annealing and GAs present the opportunity to incorporate re-calibration or learning directly into the algorithm. Approaches such as Tabu search are well suited to combinatorial optimisation problems in which an optimal combination is required from a large set of possible solutions, as is the case here.

ANNs require a definition of a training set that includes the majority of desired output patterns. For this application that would require defining all possible customer requests and pre-determining the combination of offers that might meet each request. This is problematic as it is difficult to determine in advance what requests might be made. Also, the offers made are dynamic and outside of the control of the FD Control Centre. It is similarly difficult to pre-determine what combination of offers might be available. Finally, the addition or removal of loads would require the ANN to be re-trained: a lengthy and laborious process. For these reasons an ANN is not a suitable approach here. Fuzzy logic provides a mechanism for reasoning with incomplete data. In this application the data are complete in that the task is to choose an optimal configuration from the available offers. Fuzzy logic is therefore not a suitable candidate.

The remaining three algorithms are all metaheuristic search algorithms. Such algorithms do not guarantee that the obtained solution is optimal; they strive only to give the best solution that can be achieved within the available computational limits, which are a function of processing power and available time. Metaheuristic search algorithms have the common problem of how to avoid becoming stuck in a local optimum, that may not represent the global optimum, and may actually be quite a poor solution given the available computational resource to find alternatives. The algorithms identified here all have techniques to overcome this problem (see Michiels et al., 2007 pp. 135-145).

A GA determines the best solution, and avoids local optima, by performing multiple walks within the search space. It does this by maintaining a population of candidate solutions and by selecting those that survive into the next generation using a fitness function that mimics an evolutionary selection process. Mutation, inheritance and crossover are used to generate new candidates and thereby escape local minima.

Both Simulated Annealing and Tabu search perform a single long walk within the search neighbourhood. To avoid getting stuck in a local optimum both algorithms permit deterioration in the fitness of the candidate solution. When a new iteration of the algorithm begins near-neighbours are compared with the current solution. If the threshold for determining whether a new candidate is selected is solely that it must be a better fit then this could confine the algorithm to a local optimum without exploring the wider search space. To overcome this the possibility of selecting a worse solution is allowed. The strategy for doing this differs between the two algorithms.

The Simulated Annealing algorithm mimics crystal formation in a metal as it is cooled in a controlled manner. This process is called annealing and results in a configuration of low energy and high strength. In Simulated Annealing the algorithm progresses by randomly selecting a near neighbour to the current solution. If it is better then it is selected to replace the current solution and the algorithm moves on. If it is not it can still be accepted but subject to a probability value calculated using the fitness differential and the current value of a global control parameter called the "temperature". The term "temperature" arises by analogy with the physical annealing process. The probability distribution is exponential and the global temperature value is decreased at the start of a new iteration (resulting in a tightening of the threshold for acceptance). If $f(s)$ is the fitness of the current solution, $f(s')$ the fitness of the next candidate and c_k the global temperature for the k^{th} iteration then the acceptance probability is

$$P = \begin{cases} 1 & f(s') \leq f(s) \\ \exp\left(\dfrac{f(s) - f(s')}{c_k}\right) & f(s') > f(s) \end{cases} \tag{5.3}$$

The probability of accepting a worse solution is therefore determined by the fitness such that higher fitness solutions have a higher probability of acceptance.

A Tabu search permits a move to the solution with the lowest decrease in fitness if no higher fitness candidate can be found. To prevent a return to solutions already considered, and therefore permit escape from a local optimum, a list of already considered solutions is maintained. All solutions on this list are "taboo" and cannot be reconsidered. The length of the list determines how far from the local optimum the algorithm will walk before it could potentially return. Short lists may therefore not prevent cycling into and out of a local optimum, whereas long lists may excessively restrict the neighbourhoods that can be searched. In practice the Tabu search can have excessive memory requirements (to store the lists) and excessive processing requirements (to iteratively consider all the nearest neighbours). GAs and Simulated Annealing are expected to offer superior performance and so Tabu search is not considered further.

Of the two remaining algorithms, GAs and Simulated Annealing, the choice is essentially between a multiple walk or single long walk approach. Some researchers have looked to compare the two approaches in other fields. Manikas and Cain (1996) found that GAs produced at least as good, and in some cases better, solutions when applied to a circuit placement problem in electronic engineering. Kundu et al. (2008) applied the two approaches to a personnel scheduling problem and established that Simulated Annealing was significantly faster than GAs. Although Simulated Annealing can be shown to asymptotically converge on the set of globally optimal solutions, it can only be guaranteed to do this given an infinite number of iterations (Michiels et al., 2007 pp. 149). No such proof exists for GAs. In practice multiple vs. single walks and convergence proof are the only real differences between the two algorithms; Diwekar provides a useful comparison summarising these points (Diwekar, 2008 pp. 99). On the basis of this evidence there is little to choose between the two algorithms.

The multiple walks of the GA imply a greater memory requirement to hold the solution population but with the added advantage that this can efficiently explore a large search space. Simulated Annealing, in contrast, is fast with low memory requirements and is good at exploring a limited and localised search space. This suggests that for a very large search space, as is expected in this application, the efficiency of the multiple walks used by the GA in exploring such a search space could be a key differentiator. The GA is therefore chosen as the best candidate in this application.

5.5 The Fitness Function

Irrespective of which algorithm is chosen, it is necessary to derive a fitness function that can be used to calculate the fitness of each solution found by the algorithm. The problem is how to calculate a value that indicates the relative fitness of all the candidate solutions. The fitness function must include all the parameters relevant to determining the cost to the Aggregator of using a particular offer from a particular node.

Each combination of node offers will result in a specific overall reduction (or increase) in consumed power for a specific period of time. The FD design assumes a step response for 15, 30 or 60 minutes and this is reflected in the offers made. The fitness function must include a measure of how far the aggregated offers deviate from the desired aggregate power level for the given response duration. Thus, if the aggregated order is for X MW for 15 minutes then each combination of 15-minute offers must be summed to determine the delta between the sum and X. This delta will form the power cost and must be minimised. Note that each combination can include any number of offers between 1 and n where n is the total number of offers in the offer pool. Thus, at first sight it would seem that the objective of the algorithm is simply to minimise the delta between the desired power and the actual power resulting from the summation of the relevant offers. Such an approach is, however, incomplete.

In practice the cost to the Aggregator is an economic cost since the Aggregator will maximise profit by minimising cost. There are two principal mechanisms by which the Aggregator can incur cost: compensation payments to node owners and penalty payments to customers. The exact form of the commercial arrangement between the Aggregator and node owners is unknown, but is not needed for the fitness function. It would likely contain a retention payment (possibly linked to capacity) and a usage payment. Payments might be tariffed or may be negotiated separately with each node owner. It is not intended for the scheduling algorithm to consider monetary values and instead the relative cost of using each controlled node is reflected in a cost parameter.

The cost of failing to meet the order from a customer would be reflected in a penalty mechanism between the Aggregator and the customer. However, it is not necessary to reflect this in the fitness function, as the aim of the scheduling algorithm should always be to get as close as possible to the customer's order. The underlying assumption is that much like the balancing mechanism described in Section 2.7 the penalty payments will always be larger than revenue payments for the same power. The SO needs balancing partners that are reliable and so it is important that the Aggregator always seeks to meet the order.

The Control Centre must also maintain knowledge of the reliability of individual nodes and use that knowledge in its calculation of whether to use a node or not. Reliability is incorporated into the fitness calculation by giving each node an associated reliability factor. This factor will be revised in light of experience and continually updated to ensure it reflects the node's current reliability. This is especially important given that an individual node will always retain the right to abort an event even once it is underway. Although such action would incur a penalty, it might still be rational for a node owner to take this action if the cost of loss of service, howsoever measured, would be greater. The Aggregator cannot, therefore, rely on a node delivering what it said it would and it must maintain a measure of a node's reliability.

So, the fitness function must include the summed power, the relative economic cost of the combination of offers and the relative reliability of the combination of nodes. These parameters must be combined in such a way as to provide the relative fitness of each combinatorial solution to the desired aggregate offer. The function is therefore a weighted sum of the three parameters

$$\Phi = f(\text{power, economic cost, reliability}) \tag{5.4}$$

The following provides a description of the calculation of each of the components of the fitness function before bringing them together into the overall function.

5.5.1 Power

The power delta is the difference between the target value, P_0, and the summation of all the offers within the solution where there are t offers in the solution. This value is in kW. This value is normalised to produce a power index by dividing the delta by P_0

$$P_D = \frac{\left|\left(\sum_{i=1}^{t} P_i\right) - P_0\right|}{P_0} \tag{5.5}$$

5.5.2 Economic Cost

To calculate the economic cost, each offer is given a value between 0 and 1 where 0 indicates the least expensive and 1 the most expensive offer in the pool of available offers. A normalised index is used as this removes any dependency on cost units: the index simply indicates the cost of each offer relative to all the others in the pool. Note that the members of the pool will change as new offers are received and existing offers accepted. This implies that the cost indices will vary whenever the pool constituents change and must therefore be recalculated before each run. This is done by freezing the pool at the start of each run of the optimisation algorithm. It is important that the algorithm runs against a frozen pool because if it does not then the search runs the risk of settling on a combination of offers some of which might no longer be available. It is therefore necessary to exclude from the pool any offers that will expire before the algorithm completes. Similarly any offers received during the execution of the algorithm are not included within the search. The overall economic cost of the solution is therefore the summation of the cost of each offer, *C*, normalised using the number of offers in the solution, *t*, to give a value between 0 and 1

$$C_0 = \frac{\sum_{i=1}^{t} C_i}{t} \tag{5.6}$$

5.5.3 Reliability

The reliability factor for each node is set to a value between 0 and 1 where 0 indicates reliable and 1 indicates unreliable. This assignment might at first seem counterintuitive but it is necessary because the aim of the algorithm is to minimise the fitness value. Therefore zero represents the most desirable value. The overall reliability of the solution is therefore the summation of the reliability of each node, *R*, normalised using the number of nodes in the solution, *n*, to give a value between 0 and 1

$$R_0 = \frac{\sum_{i=1}^{n} R_i}{n} \tag{5.7}$$

Note that it is assumed that each node may make multiple offers but that the offers are mutually exclusive which implies that the number of nodes, *n*, is the same value as the number of offers, *t*. Future systems might have more flexibility in that they may be able to make multiple offers that are not mutually exclusive, perhaps by sub-dividing the loads under their control to provide finer resolution. The use of different parameters for offers and nodes within the fitness function therefore provides some level of future-proofing to cope with such a scenario.

5.5.4 The Complete Fitness Function

The objective is to ensure that the aggregate offer meets the customer requirement. It is assumed that an undersupply is acceptable if it is the fittest solution and this is why the sign of the delta is ignored. An oversupply implies extra cost that will be reflected in the fitness function cost component, C_0, but could be tolerated if it resulted in a lower cost solution than one closer to the target power value.

The overall fitness function is simply the summation of the three individual parameters described above. However, there is one final requirement when bringing all three parameters together. It is possible that different operating scenarios will place different weights on the three parameters: some may value reliability more highly than cost for example. To accommodate this relative importance of the parameters there is therefore a need for additional scale factors that can be used to adjust the relative importance of the three parameters that make up the fitness function. The intention is to tune the scaling factors in light of observed results from an operational system. The overall fitness function is therefore

$$\Phi = a\frac{\left|\left(\sum_{i=1}^{t} P_i\right) - P_0\right|}{P_0} + b\frac{\sum_{i=1}^{n} R_i}{n} + c\frac{\sum_{i=1}^{t} C_i}{t} \tag{5.8}$$

The objective of the metaheuristic algorithm is to maximise the fitness of the solution by minimising the value produced by the fitness function.

5.6 Description of the Genetic Algorithm

A GA is an iterative algorithm in which a number of simple steps are repeatedly applied to a population of potential solutions, termed *chromosomes*. An iteration is called a *generation*. At the start of each generation a number of evolutionary modifications are applied to the population with two objectives:

1. Generate a new population of chromosomes by propagating the fittest solutions from the previous generation.

2. Randomly cross chromosomes and randomly mutate chromosomes.

After a predefined number of generations, or according to some fitness threshold, the algorithm stops and the fittest chromosome is selected as the result. Random crossings allow the algorithm to search near neighbours whilst random mutations allow the algorithm to escape from local optima. The population size determines the size of the search space considered by the algorithm during each generation, whilst the number of generations determines how exhaustive the search will be. The computational time required is a function of population size and the number of generations. Typically, population size might be limited to tens of chromosomes and generations to a few hundred. The GA described here is based closely on that of Zhang et al. as applied to the optimisation of component values in power electronic circuits (Zhang et al., 2001).

Each potential solution contains a combination of selected offers and is represented as a single chromosome. The chromosome is a two dimensional array of numbers organised as key-value pairs. The key in each pair contains the unique offer identifier and the value is a Boolean symbol where true means that the offer is used by the solution and false that it is not. It is possible for every offer within the offer pool to be used in a solution and so the length of the chromosome – and, hence, the number of key-value pairs – is equal to the total number of offers in the pool, t. The structure of an example chromosome is shown in Table 11. In this example the keys are randomly generated. Note that since each offer can only be used once then each key is unique within the chromosome. The order of the key-value pairs does not matter as solutions are differentiated as alternative combinations of offers, not permutations.

Table 11: structure of a chromosome.

Offer Number	Key	Value
1	013487	True
2	013996	True
3	014004	False
.	.	.
t	023005	True

The following describes the operation of the GA.

Step A: Initialisation of the offer pool

The Control Centre employs fixed offer-lengths of 15, 30 and 60 minutes. Aggregate solutions are similarly arranged such that a 15-minute aggregate solution is constructed from 15-minute offers and similarly for 30- and 60-minute aggregate solutions. Longer aggregate solutions can be constructed from multiple shorter aggregate solutions. This process is separate to the scheduling optimisation discussed here. As an aside, this mechanism also lends itself to more complex load curves since by reducing the offer length the resolution of the load curve can be increased.

Before the GA can be run it is necessary to freeze the offer pool by removing any offers that will become invalid before they can be triggered. Excluding all offers that have an expiry time between the point at which the algorithm is initiated and some point in the future determined by following three constraints achieves this:

- The time the algorithm takes to run.
- The time it takes to decide whether to trigger the event.
- The communications latency in sending the trigger commands to the nodes.

These are sequential actions and therefore the total delay is the sum of the components from each of these three actions. If T_0 is the time at which the optimisation algorithm starts and T_D the time taken by the actions outlined above, then any offer that expires between T_0 and $T_0 + T_D$ must be removed from the pool before the pool is frozen. This process is analogous to the concept of gate closure in the power markets.

The initialisation process for the offer pool is as follows (Figure 21):

A.1 Add all offers to the offer pool according to the duration requirement, i.e. if the requirement is for a 15-minute aggregate solution then add all 15-minute offers to the pool.

A.2 Exclude any offers that have already been selected for use in another aggregate solution.

A.3 Exclude all offers with an expiry time between now, i.e. T_0, and $T_0 + T_D$ where T_D is a pre-set parameter.

A.4 Calculate the absolute economic cost of each offer in the pool.

A.5 Determine the most and least expensive offers in the pool and then assign each offer a relative cost index according to

$$C_n = \frac{\gamma_n - \gamma_l}{\gamma_m - \gamma_l} \tag{5.9}$$

Where C_n is the cost index for the n^{th} offer, γ_m is the absolute cost of the most expensive offer in the pool, γ_l is the absolute cost of the least expensive offer in the pool, and γ_n is the absolute cost of the n^{th} offer in the pool. This will give the index 0 to the least expensive offer and 1 to the most expensive. Note that if the most expensive and least expensive offers have the same absolute cost then the denominator will be zero. This condition should be trapped and the denominator set to 1 in this case to avoid a divide-by-zero error.

A.6 Assign to each offer in the pool the reliability index of the node that made it.

Step B: Initialise parameters

A number of parameters are used to control the GA and must be initialised with appropriate values. (The effect of varying these is examined as part of testing and is covered in Section 6.1.). The process is as follows (Figure 21):

B.1 Set the population size, N_p. A suggested initial value is 50.

B.2 Set the maximum number of generations, $G_{\max}$. A suggested initial value is 250.

B.3 Set the probability of crossover, p_x. The higher the value the more likely crossover will occur. A suggested initial value is 0.15.

B.4 Set the probability of mutation, p_m. The higher the value the more likely mutation will occur. A suggested initial value is 0.25.

B.5 Set the selection probability upper limit, $s_{\max}$, such that $0 < s_{\max} \leq 1$. A higher value means more of the population can be considered for inclusion in the next generation. Setting this value less than 1 prevents the propagation of the worst chromosomes between generations. A suggested initial value is 1 as this will ensure the propagation of all chromosomes.

B.6 Set the selection probability lower limit, $s_{\min}$, such that $0 < s_{\min} < s_{\max}$. Increasing the value of $s_{\min}$ ensures that more of the current population will be considered for inclusion in the next generation. It prevents an arbitrarily low probability value from being selected and thereby excluding all but the fittest chromosomes. This would result in a population with insufficient diversity. A suggested initial value is 0.8.

B.7 Set the fitness threshold, Φ_T. This determines the fitness-stopping criterion. If a solution is found with a fitness value $\leq \Phi_T$ then the algorithm will stop. A suggested initial value is 0.01.

Step C: Initialise the chromosomal population of the GA

The first generation of chromosomes within the population consists of randomly generated solutions created during the population initialisation. At this point all solutions are possible although clearly some will be fitter than others. The objective here, however, is merely to fill the population with candidate solutions. The process proceeds as follows (Figure 21 and Figure 22):

C.1 Set G, the generation counter, equal to 1.

C.2 Generate a new empty chromosome by adding the identifier of every offer in the offer pool to it.

C.3 Parse the entire chromosome and for each offer identifier key within it, randomly decide whether it should be assigned a true or false value.

C.4 Calculate the fitness value, $\Phi[H_n(G)]$, for the chromosome where H_n is the n^{th} chromosome within the G^{th} generation.

C.5 Determine if the created chromosome already exists in the population, if it does discard it and repeat from step C.2.

C.6 If the size of the population is less than N_p then repeat from step C.2.

C.7 Order the chromosomes within the population according to fitness value from smallest to largest.

C.8 Calculate the relative fitness for each chromosome according to

$$\Phi_r\left[H_n(G)\right]=\frac{\Phi\left[H_n(G)\right]}{\sum_{i=1}^{N_p}\Phi\left[H_i(G)\right]} \tag{5.10}$$

This will assign a relative fitness value between 0 and 1 to each chromosome such that the sum of all the relative fitness values is 1.

C.9 Calculate the cumulative fitness for each chromosome according to

$$\Phi_c\left[H_n(G)\right]=\sum_{i=1}^{n}\Phi_r\left[H_i(G)\right] \tag{5.11}$$

The cumulative fitness value produced by the above equation will be equal to the relative fitness value for the best chromosome and equal to 1 for the worst.

C.10 Identify the fittest chromosome in the population, $H_B(G)$, i.e. the one with the smallest fitness value. This is the reference chromosome.

At this point the first generation is complete and the chromosome population has been initialised with N_p unique chromosomes. All chromosomes have an associated fitness value and the reference chromosome for the next generation, $G = 2$, has been selected.

Step D: Selection of chromosomes

At the end of each generation unless a solution satisfying the fitness threshold has been found, or the maximum number of generations have been completed, the population must be reduced to make room for new offspring. Some existing chromosomes will be carried forward and some new ones will be created. The fitness values of the chromosomes in combination with the parameters $s_{\min}$ and $s_{\max}$ are used to determine which chromosomes propagate into the next generation. The selection process is (Figure 22):

D.1 If the maximum number of generations has been reached, $G = G_{\max}$, or the fittest chromosome in the population meets the fitness threshold, $H_B(G) \leq \Phi_T$, then stop and output the fittest chromosome, $H_B(G)$, as the result.

D.2 Increment G.

D.3 Generate a random probability variable, p, between $s_{\min}$ and $s_{\max}$.

D.4 For each chromosome if $\Phi_c[H_n(G)] \leq p$, and the population size is $< N_p$, then the chromosome is selected to be a member of the new generation. The value of $s_{\max}$ therefore controls the threshold for selection of existing chromosomes. Any chromosome with a cumulative fitness value $> s_{\max}$ cannot be selected.

The value of $s_{\min}$ is used to restrict the random generation of low selection probabilities that would lead to too few existing chromosomes meeting the selection criteria thereby greatly reducing population diversity.

D.5 If the population size is < N_p then repeat from step D.4. This means multiple copies of the fitter chromosomes are likely in the new population, which maximises the chances of the fittest chromosomes being used as the seeds for new chromosomes in the next step.

Step E: Reproduction

At the conclusion of the selection step the number of chromosomes within the population will be equal to N_p. However, it will be populated only with copies of chromosomes from the previous generation. New chromosomes need to be created using the techniques of crossover and mutation. Crossover is applied first. The crossover operation involves exchanging the genetic code of two chromosomes that have been split at a randomly determined point within them as illustrated in Figure 19. The process is described below and shown in Figure 23.

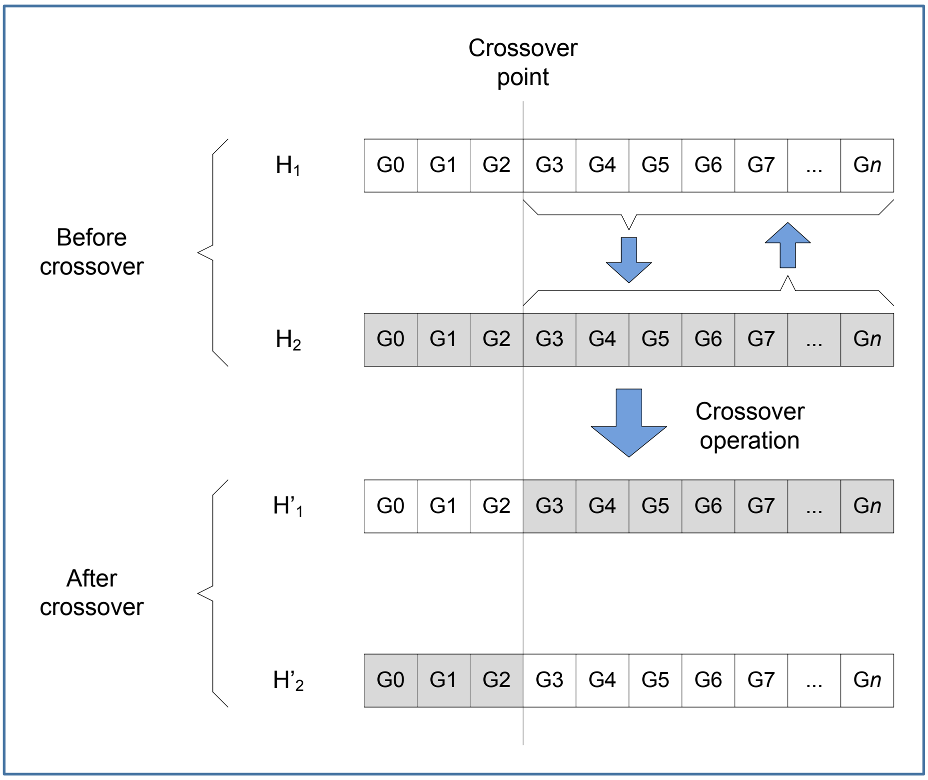

Figure 19: crossover operation.

E.1 For each chromosome within the population, generate a random probability variable, p, between 0 and 1. If $p > p_x$ then skip to the next chromosome, else mark the chromosome as selected for crossover.

E.2 If there is an odd number of chromosomes marked for crossover then unmark the last chromosome selected. Note that it is a little arbitrary as to which chromosome is unmarked and the last is suggested out of convenience. Split all the chromosomes marked for crossover into pairs.

E.3 For each pair of chromosomes select randomly and with equal probability, a crossover point between 1 and the chromosome length - 1. Split each chromosome at the selected point and join the head of the first chromosome to the tail of the second and vice versa. This process is repeated for each pair.

Mutation is also a random process applied to each chromosome in turn. In mutation one value within the chromosome is randomly flipped to create a new chromosome. This process is illustrated in Figure 20.

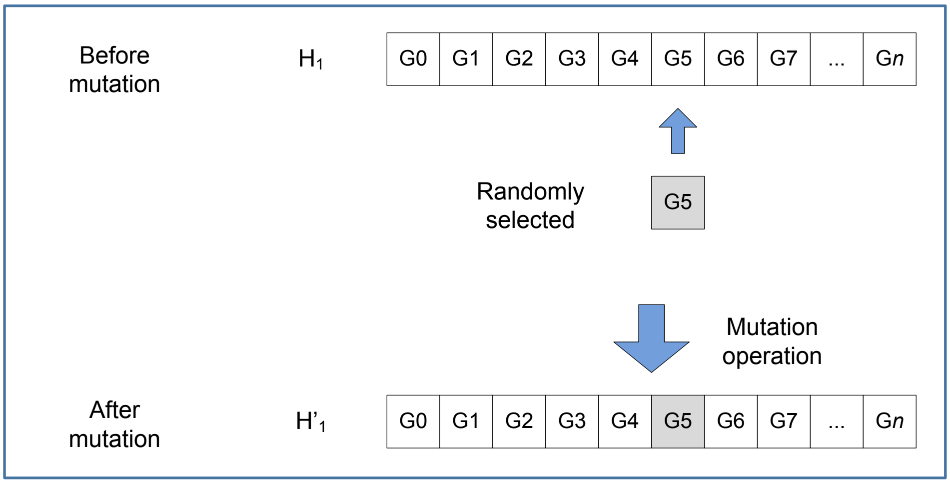

Figure 20: mutation operation.

E.4 For each chromosome within the population, generate a random probability variable, p, between 0 and 1. If $p > p_m$ then skip to the next chromosome, else select randomly and with equal probability, a mutation point within the chromosome between 1 and the chromosome length. Flip the value located at the mutation point and then move on to consider the next chromosome.

Step F: Selection of reference chromosome and application of elitist function

The creation of a new generation is now complete and this step ensures the survival of the fittest chromosome found so far. It is possible that the previous generation's fittest chromosome has not survived into the new generation, and that the fittest chromosome in the new generation is less fit than that of the previous generation. The elitist function compares the fittest chromosomes of the previous and new generations and retains whichever is fitter. It proceeds as follows (see Figure 24):

F.1 Order the chromosomes within the population according to fitness value from smallest to largest.

F.2 Calculate the relative fitness for each chromosome according to (5.10).

F.3 Calculate the cumulative fitness for each chromosome according to (5.11).

F.4 Identify the fittest chromosome in the population, $H_B(G)$. This is the reference chromosome.

F.5 If this is not the first generation, i.e. $G > 1$, then compare the fittest chromosome in the new generation, $H_B(G)$, with that of the previous generation, $H_B(G - 1)$, and if $H_B(G - 1)$ is fitter then replace $H_B(G)$ with $H_B(G - 1)$.

F.6 Repeat from step D.1.

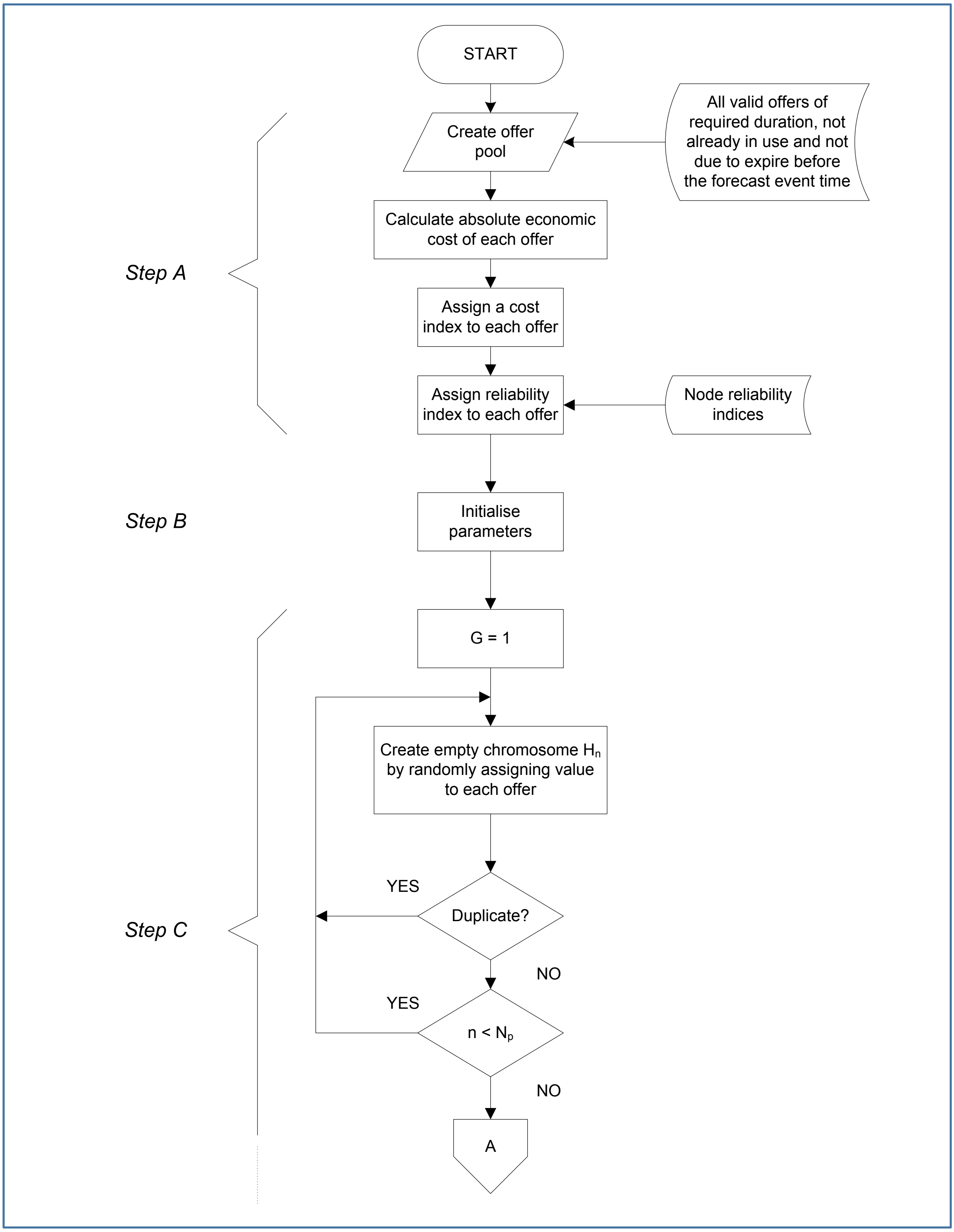

Figure 21: flowchart for GA (1 of 4).

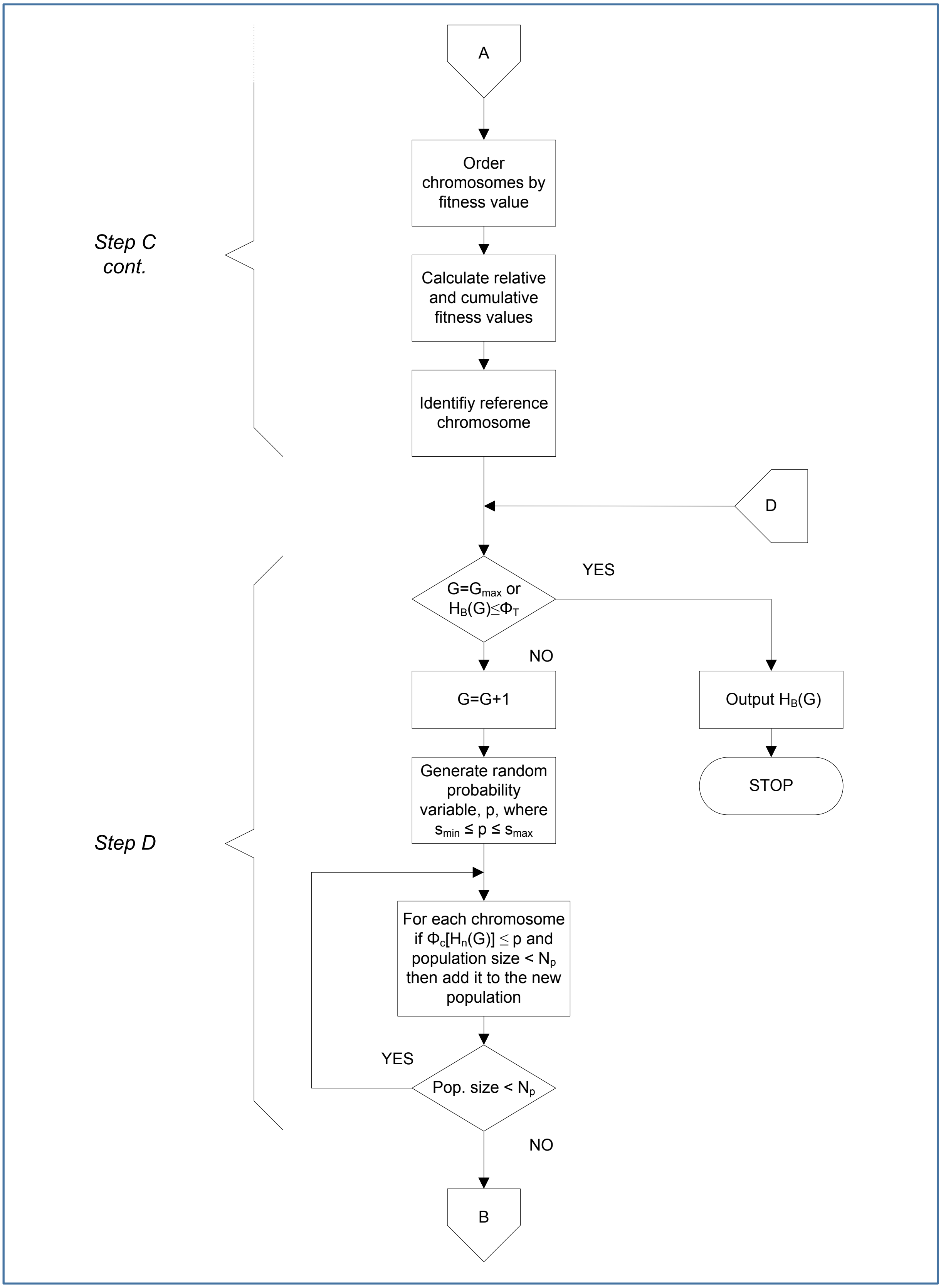

Figure 22: flowchart for GA (2 of 4).

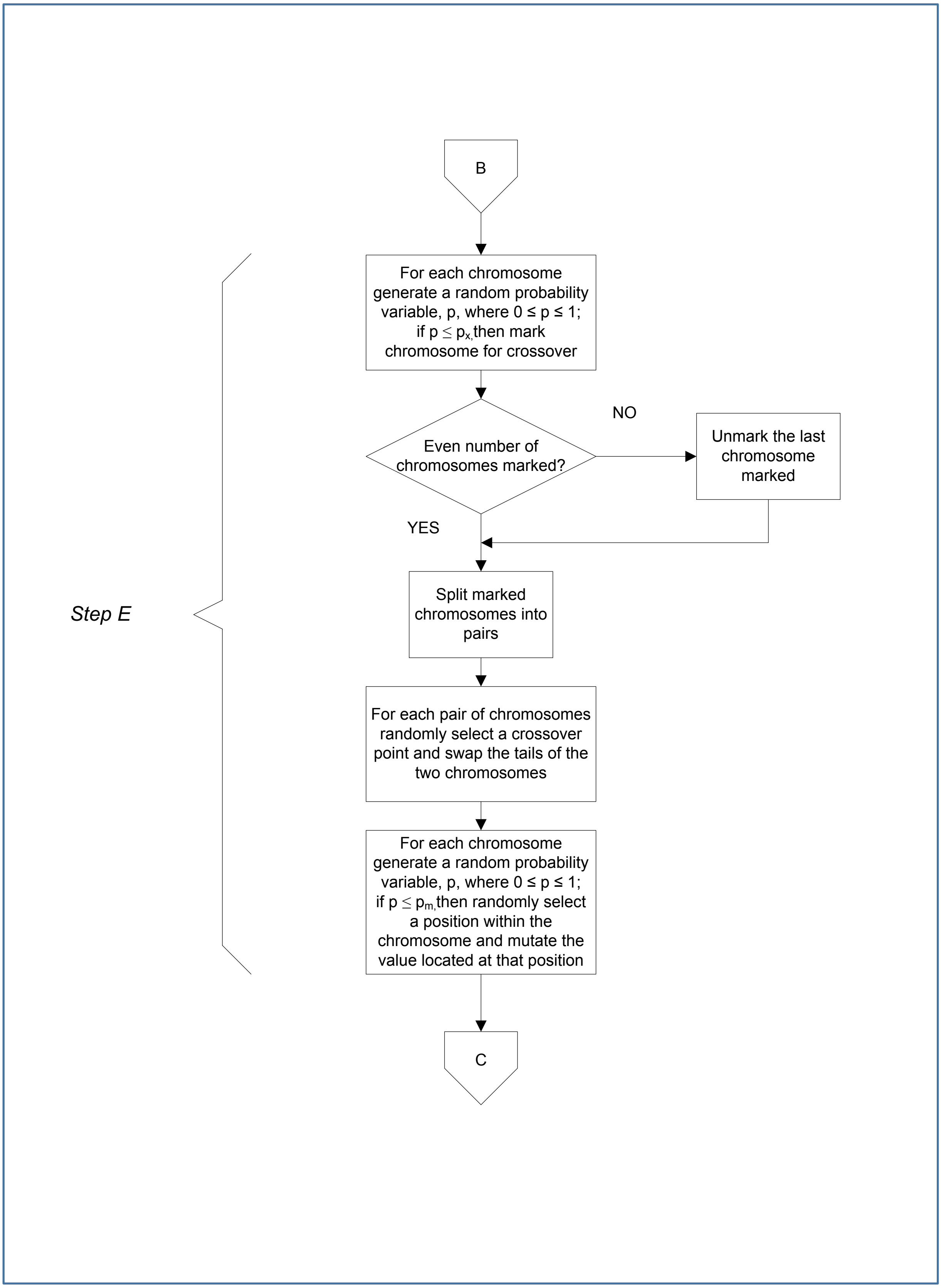

Figure 23: flowchart for GA (3 of 4).

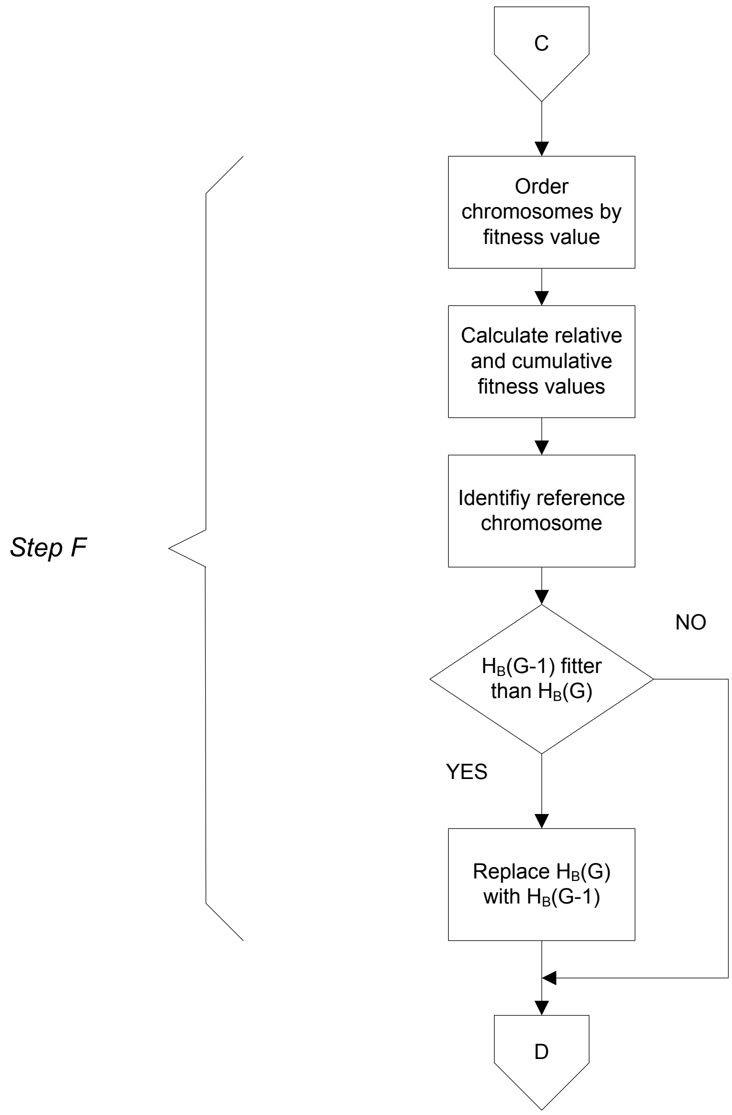

Figure 24: flowchart for GA (4 of 4).

5.7 Summary and Conclusions

This chapter has provided a detailed description of the selection and design of an appropriate algorithm to provide the optimised dispatch functionality required by the FD demonstrator. A literature review revealed that optimal dispatch within this context is a classic scheduling problem. Further consideration of the problem showed that this is most likely an *NP*-hard problem and not solvable in polynomial time. This means that the processing resources required to solve the problem scale exponentially with problem complexity. An algorithm is therefore needed that can search for a near optimal result within an acceptable processing budget. Of the identified candidates a Genetic Algorithm was identified as the most suitable.

The search progresses by retaining the best solutions in each generation and by modifying them to generate new solutions. To measure the relative fitness of each function a fitness function was derived. This function considers the difference between the aggregate power offered by the solution and the target power, the aggregate reliability and the aggregate cost.

A detailed description of the operation of the algorithm is described to facilitate its implementation and testing. The results of testing a software implementation of the algorithm are discussed in the next chapter.

Chapter 6

Analysis of the Results of Testing of the Dispatch Algorithm

6.1 Implementation

As discussed in the previous chapter, the basic structure of the algorithm is adapted from that of Zhang et al (Zhang et al., 2001). This includes the sequence of steps to implement the algorithm: establishing the first generation, reproduction and application of the elitist function. However, the fitness function used here was designed specifically for this application and not adapted from any other author. Similarly the mechanism used to represent combinations of offers within chromosomes, and for representing a population of possible solutions, were designed specifically for this application.

The algorithm was implemented using the Microsoft® .NET Framework in order to test it. This implementation uses a database built using Microsoft SQL Server® 2008 to store the chromosome populations and a .NET-connected application implemented in Microsoft Visual C#® to provide a Web-based interface for controlling the application.[12] The use of Web services allows the database and algorithm to run on a different machine to that which is used to display the Web interface, if required.

[12] Microsoft, SQL Server and Visual C# are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Throughout the implementation of the software application, effort was made to follow the principles of good object-oriented software engineering. This included modularising the application and using defined private or protected object interfaces. This approach allowed an agile-style rapid software development to take place and also ensured that the resulting source code can be readily understood, maintained or extended by other developers. Example source code is included in Appendix B.

In all the tests that follow the application was executed on an Apple® MacBook Pro® computer within a VMware Fusion 5® virtual machine running Microsoft Windows® 7 Professional (64-bit).[13] The virtual machine had one core of a dual core 2.3 GHz Intel® Core™ i5-2415M CPU and 1GB of memory dedicated to it.[14] This does not represent an optimal environment for performance testing but absolute performance data are not the concern here. The interest is in how the algorithm scales and so it is relative data that are relevant. Absolute performance will always be a function of the particular hardware platform and there are so many alternatives that it would be impractical to test them all.

[13] Apple and MacBook Pro are trademarks of Apple Inc., registered in the U.S. and other countries. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. VMware Fusion 5 is a registered trademark of VMware, Inc. in the United States and/or other jurisdictions.

[14] Intel and Intel Core are trademarks of Intel Corporation in the U.S. and/or other countries.

6.2 Testing, Tuning and Performance Analysis

To test the algorithm it was run against a set of test data. The areas that were tested are:

- The algorithm's ability to converge on an acceptable solution.
- The algorithm's ability to find a known globally optimal solution.
- The algorithm's performance.
- The effects on the algorithm's search performance of varying the input parameters.

The analysis is split between verification of the fitness function and comprehensive testing of the GA. Rather than describe the results of each test in isolation from those of the others, greater analytical insight is achieved by combining the results from all the tests and conducting the analysis in three specific areas: accuracy, repeatability and performance. Accordingly the remainder of this section is divided into three subsections that cover the creation of the test data, the tests that were run and finally the analysis of the results obtained.

6.2.1 Creation of the Test Data

To test functionality such as finding a known global optimum it is necessary to perform an exhaustive search to determine the best solution. The data set must be necessarily limited in size to make an exhaustive search possible within a reasonable amount of time. This small data set will be too small to adequately assess performance and so an additional much larger data set is required for this task. Thus two data sets are required. The small data set is given the label TD1. The other, labelled TD2 is used in varying sizes to allow an assessment of scalability. For convenience it is assumed that all offers within the pool are for reductions of 15 minutes and are available to be used. The ability of the algorithm to filter on period and reject offers already in use is not core to correct operation and is not explicitly tested. Since the algorithm is oblivious to whether it is being asked to search for a reduction or increase this functionality is also not tested. The test data characteristics are shown in Table 12.

Table 12: test data characteristics.

	TD1	**TD2**
Number of unique offers	20	500
Nominal power	Randomly assigned in the range 150 - 250 kW	Randomly assigned such that the population is normally distributed in the range 50 - 250 kW with a mean of 110 kW and SD of 40 kW
Offer power	Randomly assigned in the range 10% - 50% of nominal power	
Reliability	Randomly assigned in the range 0 - 1	
Cost	Randomly assigned in the range 1 - 10 per kW offered	

The data are generated assuming nodes with a nominal power between 150 kW and 250 kW inclusive for TD1 and 50 kW and 250 kW inclusive for TD2. This represents a realistic distribution of powers for commercial installations such as HVAC, as seen in the DD-FD project. To create TD2 a normal distribution of node ratings is assumed with a mean rating of 110 kW and standard deviation of 40 kW. This distribution is chosen so as to concentrate the population of nodes around the smaller power sizes to ensure that solutions will need to contain offers from many nodes. There are two reasons for this:

- The first is that it is assumed that within a sufficiently large population of nodes one would see more small sites than large ones and a normal distribution is a good way to represent such a population; and
- The second is that one would expect to see an optimal solution include large offers in preference (as single nodes are likely to be fitter than aggregations of smaller nodes given the effects of accumulating cost and reliability) with the balance made from smaller offers.

The result of applying these criteria is the distribution of the maximum 500 nodes as shown in Figure 25. The small size of the TD1 data set means that a probability distribution is inappropriate and so powers are assigned randomly.

For both TD1 and TD2 to determine the value of the offer made by each node it is assumed that all nodes make an offer to reduce their consumption by between 10% and 50%. These are realistic values based on observations of real sites that were made during the DD-FD project and take into account the fact that installations rarely operate for extended periods at their nominal power. For each node, therefore, a random value between 0.1 and 0.5 is calculated and multiplied by the nominal power to determine the size of the offer. A reliability index is calculated by assigning a random value between 0 and 1. Finally an absolute cost is assigned to each offer by randomly assigning a cost per kW offered between 1 and 10 (currency units are irrelevant for the purpose of testing).

For performance and scalability testing smaller data sets are created from TD2. To reduce the size of the data set intermediate offers are skipped. For example, selecting every other offer creates a 250-offer data set.

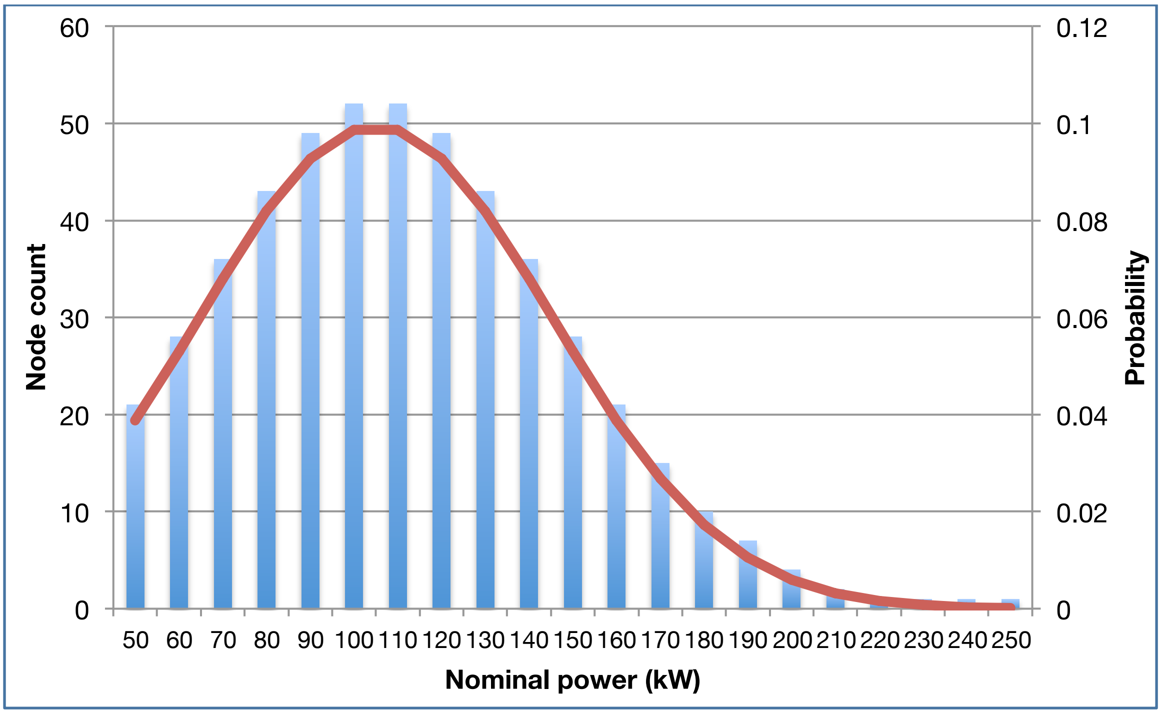

Figure 25: distribution of test node power ratings for TD2.

The algorithm works by selecting offers for inclusion in the solution. Any number of offers may be selected between only one offer and all offers. Thus all combinations of offers represent valid solutions even though some will have very poor fitness. To perform an exhaustive search it is necessary to calculate all possible solutions, calculate the corresponding fitness values using equation (5.8) and then rank the solutions by fitness.

The data set TD1 has 20 offers and therefore has 20 sets of solutions from those containing only one selected offer to the set containing the one solution in which all 20 offers are selected. The number of possible solutions in each set can be calculated using the formula for the number of combinations of r objects from a set of n objects

$$C = \frac{n!}{(n-r)!r!} \tag{5.12}$$

Using equation (5.12) the number of solutions in each solution set can be calculated for TD1 and therefore the total number of solutions in the search space found. This is shown in Table 13 below.

Table 13: number of solutions for each valid number of selected offers for the TD1 data set.

Number of offers selected	Number of solutions	Number of offers selected	Number of solutions
1	20	11	167,960
2	190	12	125,970
3	1,140	13	77,520
4	4,845	14	38,760
5	15,504	15	15,504
6	38,760	16	4,845
7	77,520	17	1,140
8	125,970	18	190
9	167,960	19	20
10	184,756	20	1

Total number of solutions	1,048,575

In a similar way the size of the search space can be calculated for any size of offer pool. This is shown in Table 14.

Table 14: pool characteristics for the test data sets.

	TD1	TD2			
Number of unique offers	20	500	250	100	50
Number of unique solutions	1,048,575	> 3×10^{151}	> 10^{79}	> 10^{30}	> 10^{15}

6.2.2 Test Descriptions

Test 0: The fitness function

Objective

In this test the efficacy of the fitness function as a means of identifying the best solution is tested. This test should show that the fitness function is able to differentiate solutions through a single fitness value that is able to reflect the power delta, cost and reliability parameters. This test should verify that the fitness function correctly identifies the best combination of minimal power delta, low cost and high reliability.

Method

Calculate all possible solutions for the TD1 data set and record the combination of offers in each solution together with the aggregate power, aggregate reliability and aggregate cost (absolute and index). Calculate the fitness value for each solution using equation (5.8) and with the scaling parameters set to their default values of 1 (i.e. equal weighting to each parameter). By analysing the power, reliability and cost of the solutions with the lowest fitness value verify that the order of the solutions does reflect the desirability of each solution in terms of the three parameters.

Test 1: Ability to converge

Objective

In this test the basic ability of the algorithm to find a solution by repeatedly improving its best solution is tested. This test should show that as the algorithm progresses each subsequent population produces fitter solutions until the fitness threshold or maximum number of generations is attained. It does not need to show that the global optimum is found, only that the search continues toward finding it. This test considers all three degrees of freedom: target power, reliability and cost.

Method

Run the algorithm 10 times using test data set TD2 with 100 offers and the following parameters for each run:

P_0	1000 kW	p_m	0.250
N_p	50	p_x	0.150
$G_{\max}$	250	$s_{\min}$	0.8
Φ_T	0	$s_{\max}$	1

Test 2: Ability to find a known global optimum

Objective

In this test the ability of the algorithm to find a known globally optimal solution is tested. It is tested on a relatively small data set, constrained in size to allow an exhaustive search to verify the optimum. This test considers all three degrees of freedom: target power, reliability and cost.

Method

Run the algorithm 50 times using test data set TD1 and the following parameters for each run:

P_0	500 kW	p_m	0.250
N_p	50	p_x	0.150
$G_{\max}$	250	$s_{\min}$	0.8
Φ_T	0	$s_{\max}$	1

Test 3: Performance

Objective

In this test the performance of the algorithm is tested. This test should show that as the number of offers increases adequate performance is maintained and, ideally, that processing time scales linearly with the number of offers. The population size is kept constant and the number of generations is limited to 2,500 as this should be enough to show the convergence as predicted by Test 1 in even the largest 500-offer data set. The larger data sets have correspondingly larger chromosomes and understanding the impact on performance of increasing chromosome length is the primary objective. This test considers all three degrees of freedom: target power, reliability and cost.

Method

The algorithm is run 5 times for each different number of offers (50, 100, 250 and 500) using the following parameters for each run:

P_0	1000 kW	p_m	0.250
N_p	50	p_x	0.150

G_{max}	2500	s_{min}	0.8
Φ_T	0	s_{max}	1

6.2.3 Analysis of the Test Results

Verification of the fitness function

A program was written to calculate all the 1,048,575 possible combinations of solution within the TD1 data set and store them in the database. This program completed in ~ 15 minutes. A second program then calculated the aggregate power, cost and reliability values and the fitness for each solution. This program completed in ~ 42 minutes. Even allowing for inefficiencies in the algorithm used to calculate the solutions, and the potential use of greater processing power, it is clear from this result that an exhaustive search will quickly become infeasible for even a moderately sized offer pool.

The fittest 20 solutions identified by the exhaustive search of TD1 are shown in Table 15. Notice that although solution 124944 achieves exactly the target power of 500 kW it is only the 18th best solution. Examining the aggregate offers reveals that solution 124944 has an absolute aggregate cost of 3,267 and an aggregate reliability of 3.21. The best solution, 125154, has an absolute aggregate cost of 3,287 and an aggregate reliability of 2.71. Thus with equal weighting of the parameters the enhanced reliability of solution 125154 counters the marginal increase in cost and power delta relative to solution 124944. This shows that the fitness function is combining all three parameters to produce the overall fitness value.

Table 15: fittest 20 solutions for the TD1 data set.

Identifier	125154	125187	97034	125183	226485	159791	120421	110062	159627	109740	70526	226824	226823	346166	125184	51338	43109	124944	393505	123710
Offer 01						✓			✓		✓									
Offer 02			✓											✓						
Offer 03					✓	✓		✓	✓	✓	✓	✓	✓	✓			✓		✓	
Offer 04	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Offer 05							✓													
Offer 06																				
Offer 07																				✓
Offer 08	✓	✓	✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Offer 09					✓				✓	✓								✓		
Offer 10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
Offer 11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Offer 12	✓				✓	✓		✓				✓	✓	✓				✓	✓	
Offer 13	✓	✓	✓	✓			✓	✓		✓	✓				✓	✓	✓	✓		✓
Offer 14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Offer 15																				
Offer 16				✓															✓	
Offer 17													✓		✓					
Offer 18												✓								
Offer 19																				
Offer 20		✓												✓					✓	
Aggregate Power (kW)	504	496	499	499	491	495	499	488	499	492	496	489	488	502	505	476	488	500	502	491
Fitness	0.2860	0.2863	0.2867	0.2899	0.2931	0.2933	0.2936	0.2940	0.2976	0.2984	0.2985	0.2988	0.2989	0.2992	0.3001	0.3006	0.3010	0.3021	0.3024	0.3031
Aggregate Cost	3287	3251	3160	3252	2754	2702	3114	3047	2666	3011	2959	2834	2826	2823	3323	3091	3055	3267	2915	3136
Aggregate Cost Index	2.8496	2.8168	2.7338	2.8177	2.3428	2.2954	2.6919	2.6308	2.2625	2.5980	2.5506	2.4157	2.4084	2.3847	2.8824	2.6919	2.6591	2.8314	2.4686	2.7119
Aggregate Reliability	2.71	2.75	2.96	2.94	3.16	3.37	3.14	2.77	3.65	3.05	3.26	3.12	3.09	3.52	2.92	2.36	2.88	3.21	3.50	2.99

If operational experience shows that equal weighting is inappropriate then the scaling factors can be adjusted as required. Increasing the scaling factor of a parameter increases its impact on the fitness value. Table 16 shows some example values applied to the fittest 20 solutions identified in Table 15.

Table 16: solutions from Table 15 with modified scalars.

				A	B	C
			a (power)	6	0.8	0.8
			b (reliability)	0.6	2	0.8
			c (cost)	0.6	0.8	3
Identifier	**Power (kW)**	**Reliability**	**Absolute Cost**			
125154	504	2.71	3287	0.2148	0.3914	0.5422
125187	496	2.75	3251	0.2150	0.3941	0.5390
97034	499	2.96	3160	0.1828	0.4070	0.5301
125183	499	2.94	3252	0.1847	0.4083	0.5419
226485	491	3.16	2754	0.2731	0.4241	0.4922
159791	495	3.37	2702	0.2300	0.4369	0.4871
120421	499	3.14	3114	0.1870	0.4233	0.5310
110062	488	2.77	3047	0.3060	0.4014	0.5246
159627	499	3.65	**2666**	0.1894	0.4571	**0.4870**
109740	492	3.05	3011	0.2654	0.4217	0.5245
70526	496	3.26	2959	0.2223	0.4344	0.5194
226824	489	3.12	2834	0.2981	0.4262	0.5048
226823	488	3.09	2826	0.3090	0.4245	0.5041
346166	502	3.52	2823	0.2011	0.4506	0.5017
125184	505	2.92	3323	0.2341	0.4153	0.5572
51338	476	**2.36**	3091	0.4396	**0.3821**	0.5366
43109	488	2.88	3055	0.3102	0.4136	0.5333
124944	**500**	3.21	3267	**0.1812**	0.4343	0.5531
393505	502	3.5	2915	0.2031	0.4519	0.5135
123710	491	2.99	3136	0.2791	0.4219	0.5408

The three columns A, B and C represent three separate runs of the algorithm, each with the scaling factor values shown in the relevant column. In Column A the scaling factors are chosen such that power is the most important parameter. With these values the function correctly identifies the only solution with exactly the right target power. Column B has scaling factors chosen to weight reliability and correctly identifies the most reliable solution. Column C does the same for cost.

Accuracy

The ability of the algorithm to find a solution that is as close as possible to the global optimum, within an acceptable processing budget, is of paramount importance. The objective of Test 1 was to verify the basic functionality of the algorithm and that it would iteratively improve its estimate of the optimal solution. A plot of the fitness value of the best solution in each generation for each of the ten runs is shown in Figure 26.

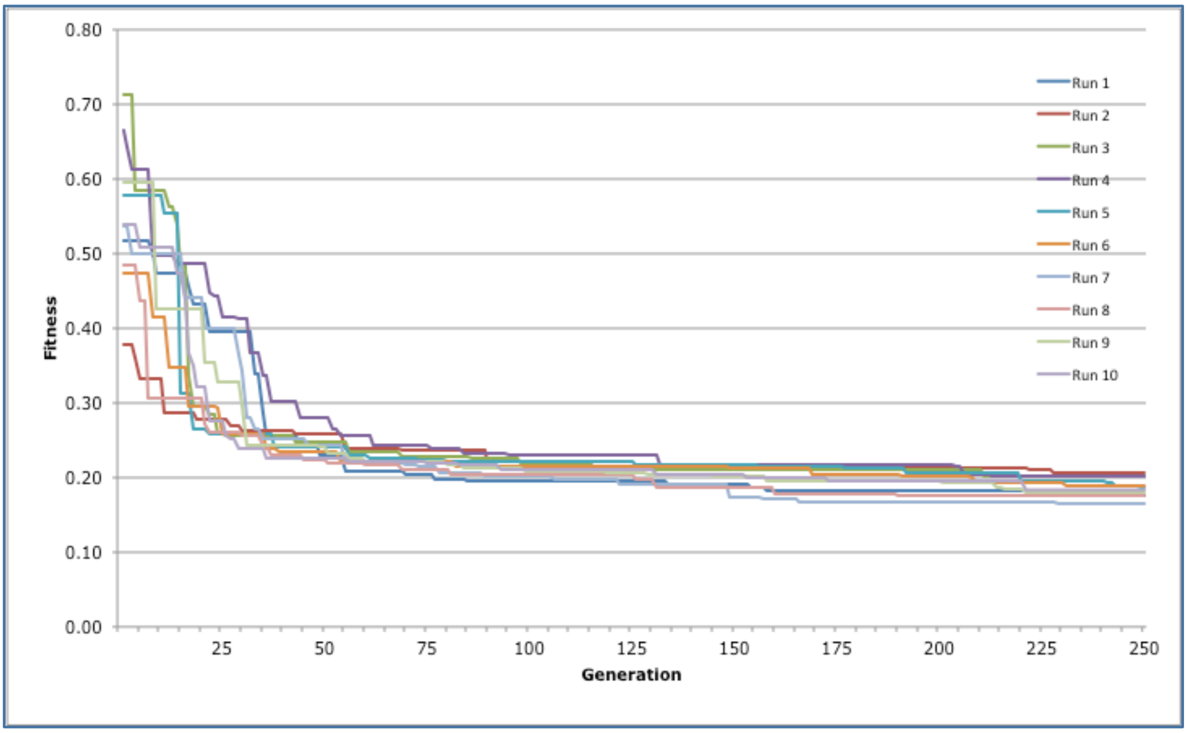

Figure 26: the best fitness by generation for each of the 10 runs in Test 1.

It can be seen that whilst there is a relatively large range of starting values the values rapidly converge on a value of around 0.20 with relatively little improvement shown in any run after ~ 75 generations. By ~ 150 generations most runs have produced a solution very close to the eventual answer. This shows that the algorithm is indeed performing an iterative improvement given a random starting point.

The results do not mean that running the algorithm for longer would not improve the answer, only that the improvements become smaller and take longer as the number of generations completed increases, suggesting diminishing returns on investment in processing time. An analysis of the rate of change in the fitness value taken from the much longer test runs used in Test 3 supports this conclusion (see Figure 27).

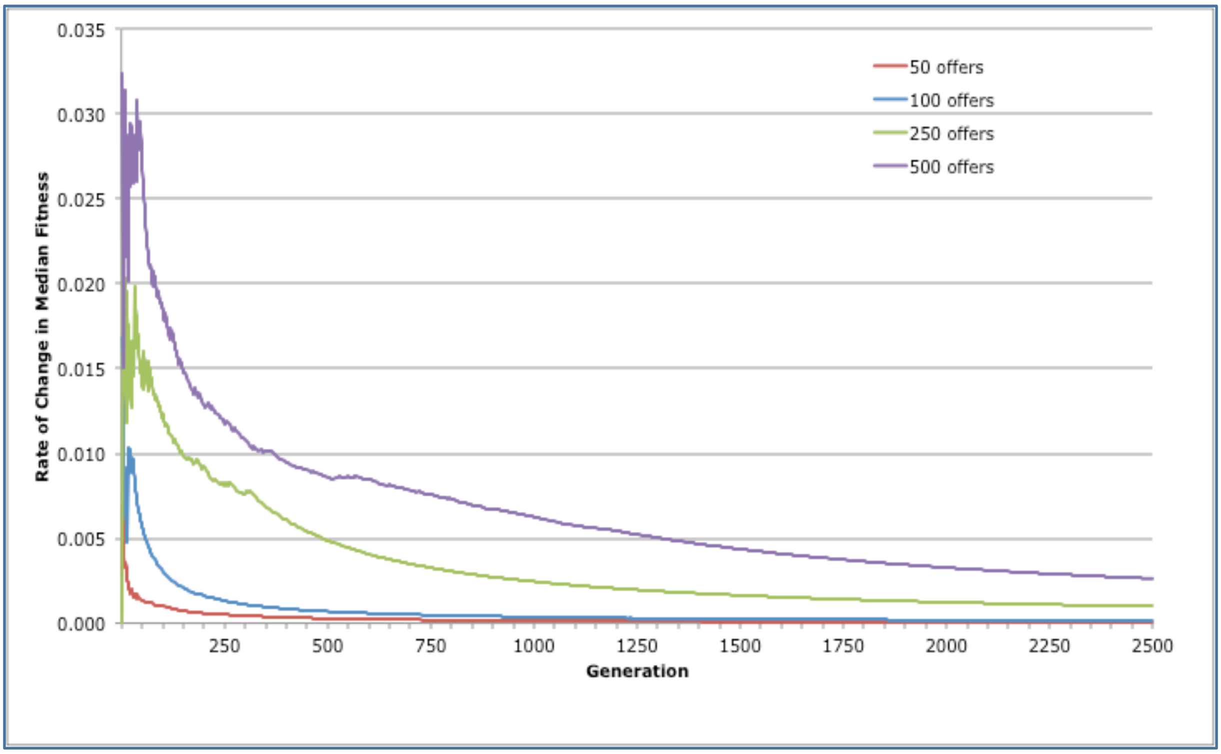

Figure 27: the rate of change in the median fitness value

for each data set used in Test 3.

This chart reveals that for the larger data sets improvement does continue in the high generation numbers but that the rate of change of this improvement decreases rapidly after the first few hundred generations, suggesting that there may be little advantage in consuming valuable processing time letting the algorithm continue in the hope of finding a significantly better solution. It would, however, be interesting to see if this could be improved by using larger populations with larger pool sizes. This will be considered further below.

The error range within a population (defined here as the difference between the best solution in a particular generation and the worst) should also decrease as fitter solutions are selectively propagated into future generations and begin to dominate. This is countered to some degree by the level of crossover and mutation, which act to increase diversity irrespective of whether the new chromosomes are better or not. The change in error range is shown in Figure 28 (for the sake of clarity for Run 1 of Test 1 only). As observed above, there is very little improvement in fitness after generation number 55. The error range shows a rapid narrowing as predicted but also decreases very little after this generation.

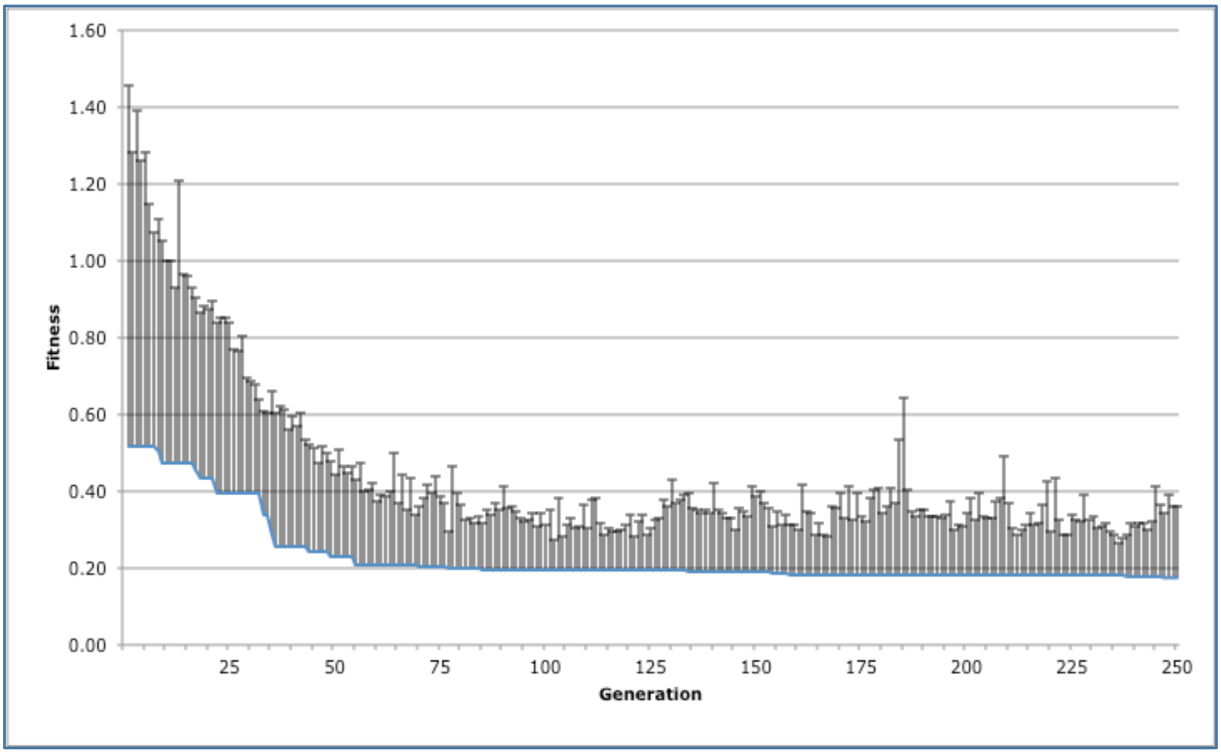

Figure 28: fitness improvement by generation for Run 1 of Test 1.

For the TD1 data set the solution space is sufficiently small to be able to perform an exhaustive search. Test 2 was designed to allow the ability of the algorithm to find this global optimum to be determined. The best solution found in each run was analysed to see if it was one of the known top 20 solutions as detailed in Table 15. The results are shown in Figure 29. This chart shows each of the 20 fittest solutions and the number of times that the algorithm found each one. The fittest solution is on the left with the least fit on the right. The purpose of this chart is to show how often the algorithm successfully got within ~5% of the best value for the fitness. It can be seen that the best solution was found more times than any other: 13 times compared to seven for the next most frequently found solution. The solutions found are clustered within the top 10 solutions with only 14 solutions outside of the top 10 being selected, and only 8 outside of the top 20. Thus, the algorithm has found one of the top 10 solutions 72% of the time, one of the top 5 solutions 58% of the time and the best solution 26% of the time.

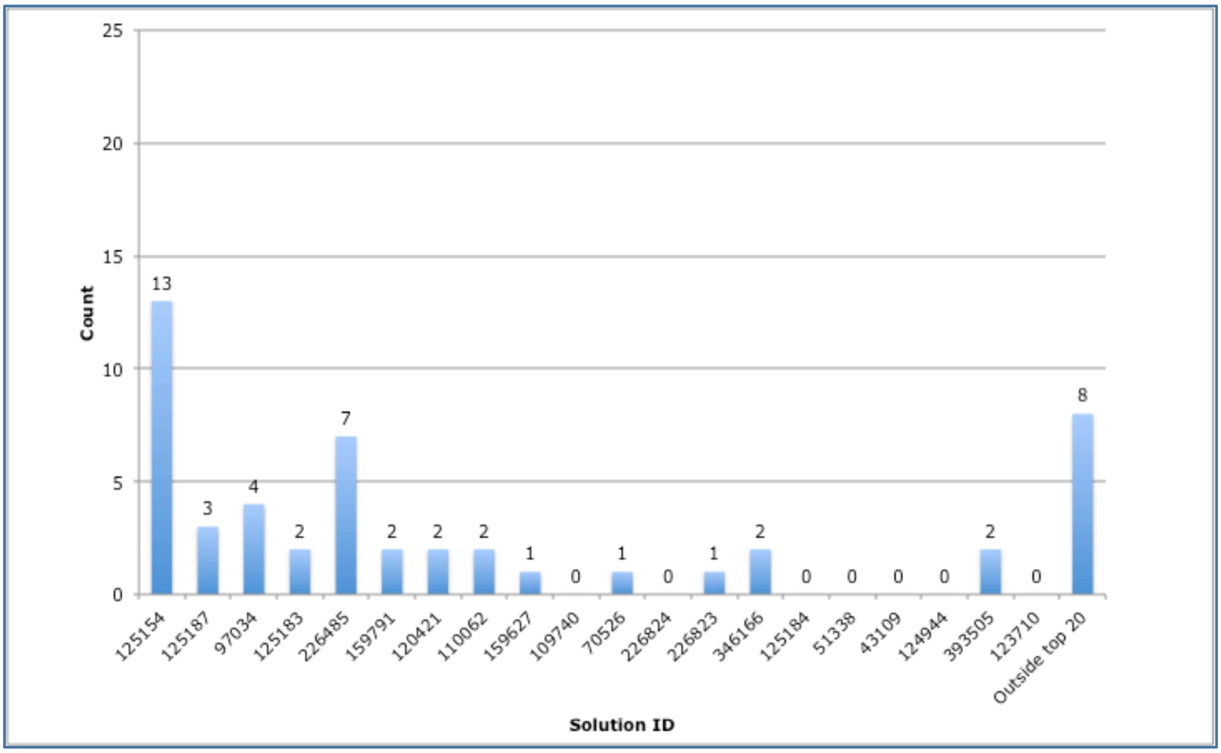

Figure 29: count of the number of times each of the known top 20 solutions was found by the GA in Test 2.

The algorithm was run with a maximum of 250 generations and a population size of 50 chromosomes. Therefore the absolute maximum number of unique solutions that could ever be considered is 12 × 250 = 12,500 although in practice the non-zero value for s_{min} and the application of the elitist function mean that a certain number of chromosomes will always be propagated between generations thereby limiting the solution space. Thus, in any one run the algorithm can only ever consider <12% of the search space. The starting position is randomly chosen and there is no seeding or weighting. From these results it is clear that the algorithm is performing a directed search and, at least in this case, can be expected to find a near optimal answer from any random starting position. There is no reason to assume that a similar result would not be found with any size data set as long as a sufficiently large population and number of generations are allowed. Since the solution space scales linearly with the number of generations and the population size but the search space scales exponentially with the number of offers this is clearly an important characteristic of the GA and will be considered further in the section on performance.

Repeatability

With a large data set it is not necessary that the algorithm always returns the same solution but it is important that all the solutions returned over a number of independent runs provide solutions of similar fitness given defined $G_{\max}$ and N_p. For the ten runs executed in Test 1 an analysis of the eventual solutions reveals substantial diversity in the offers used as shown in Figure 30. Whilst some offers are used by more than half of the solutions, no offer is used by all the solutions. This suggests that the algorithm is discovering a number of local optima. It could be that the algorithm is finding it difficult to move further within the search space once a local optimum has been found, or it could be that there are actually a large number of local optima which vary very little from the global optimum.

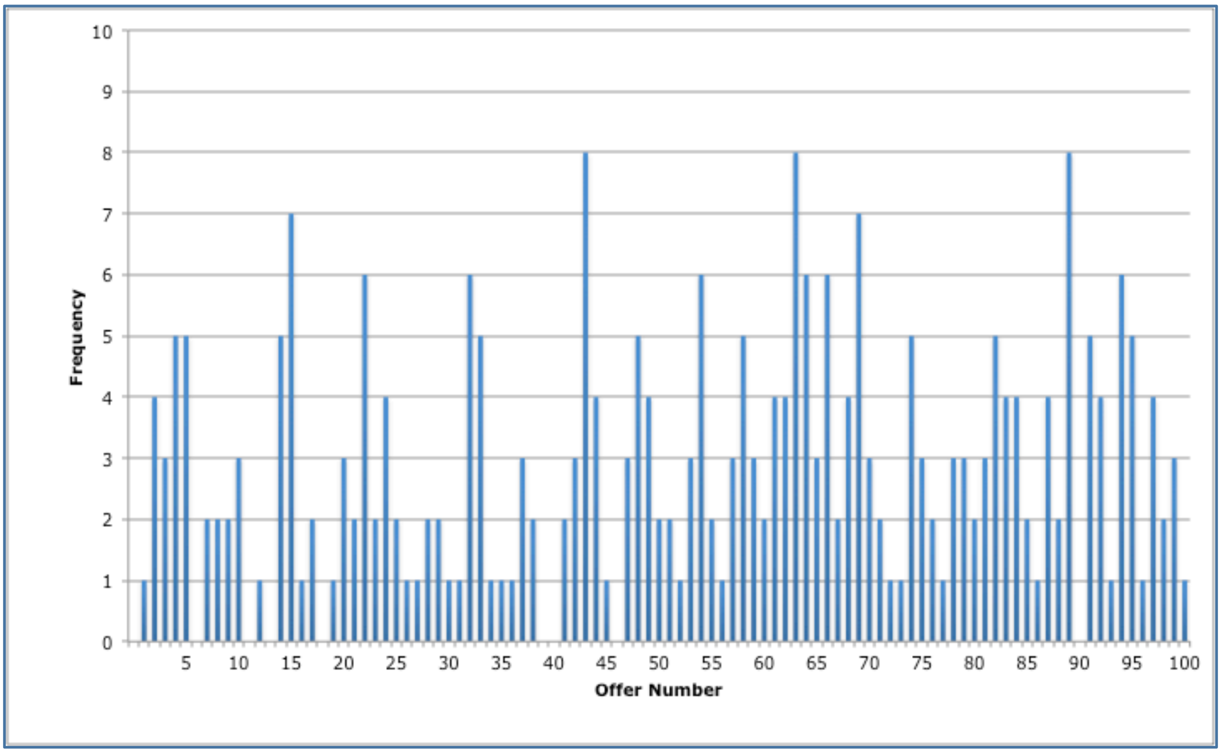

Figure 30: frequency of use of individual offers within the best 10 solutions in Test 1.

The GA is designed to avoid the first problem and there is sufficient variation due to mutation and crossover to ensure that it is able to escape from any local optimum. This suggests then that it is the second possibility that is more likely. There is relatively little variation in cost and reliability, as one would expect there to be only minor differences between different nodes. There are likely to be, therefore, a large number of similar offers within the pool and therefore a large number of possible combinations having broadly similar values for reliability and cost for a given aggregate power. It is therefore highly likely that there are many solutions that vary very little in fitness value and, therefore, a number of local optima. This is supported by the analysis of how many offers are used by each solution as shown in Figure 31. The number used varies very little. Between 27 and 30 offers are used in each solution even though the final solutions vary greatly across all the runs.

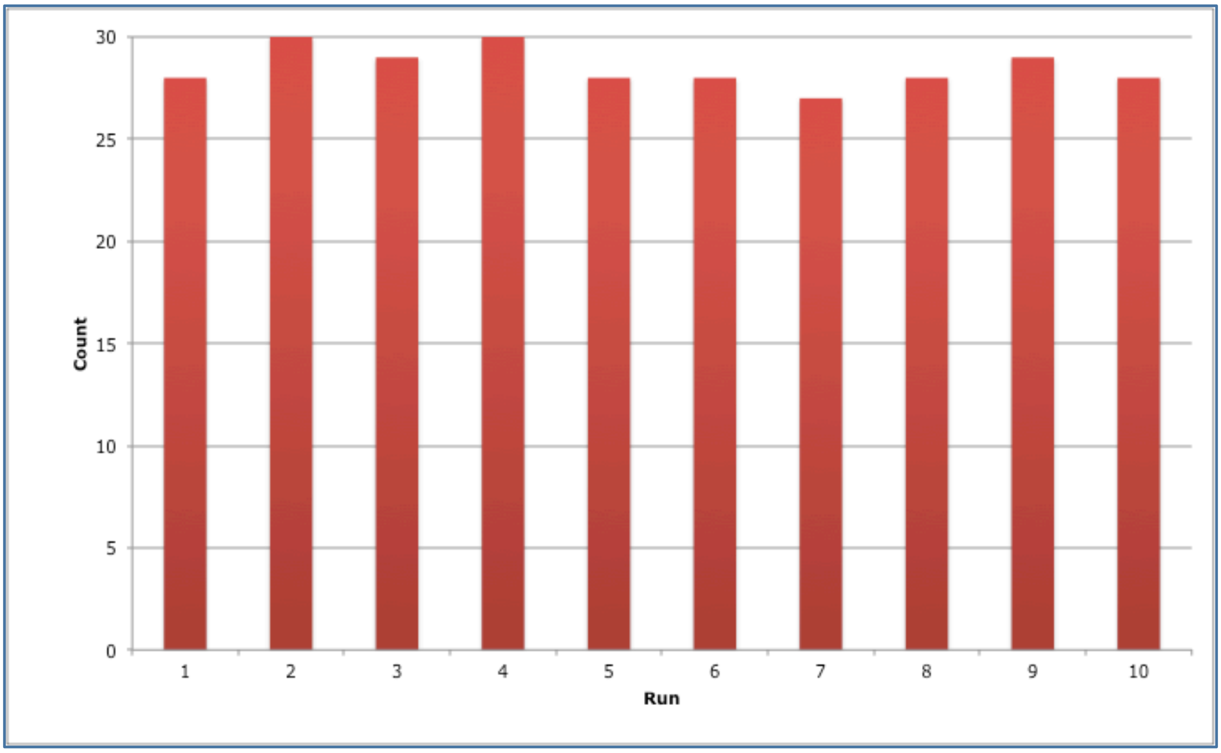

Figure 31: count of the number of offers used in each of the best solutions in Test 1.

It is very likely that in any practical system the total population of offers will contain many similar offers and that the target power will be many times greater than the single largest offer so a similar result would be expected in an operational system. This is useful as it implies that many different nodes will be used and the random nature of the algorithm will effectively prevent certain nodes being used more than others, thereby ensuring that all participants receive a return and are incentivised to continue participation.

Performance

Figure 26 and Figure 27 suggest that allowing the algorithm to run for a very large number of generations is unlikely to yield significantly better solutions than capping it a point shortly after the cusp of the curve as depicted in Figure 27. Clearly the optimum number of generations is greater for larger offer pools; this will be considered in more detail below. The median fitness curves calculated from the results of Test 3 are shown in Figure 32. For each data set the median fitness was calculated for each of the 2,500 generations over all five runs. As one would expect more generations are required to achieve convergence as the size of the offer pool increases. This is because the larger number of possible combinations means that the aggregate fitness is larger to begin with and it takes longer (more mutations and crossovers) for combinations to be created that include more beneficial offers.

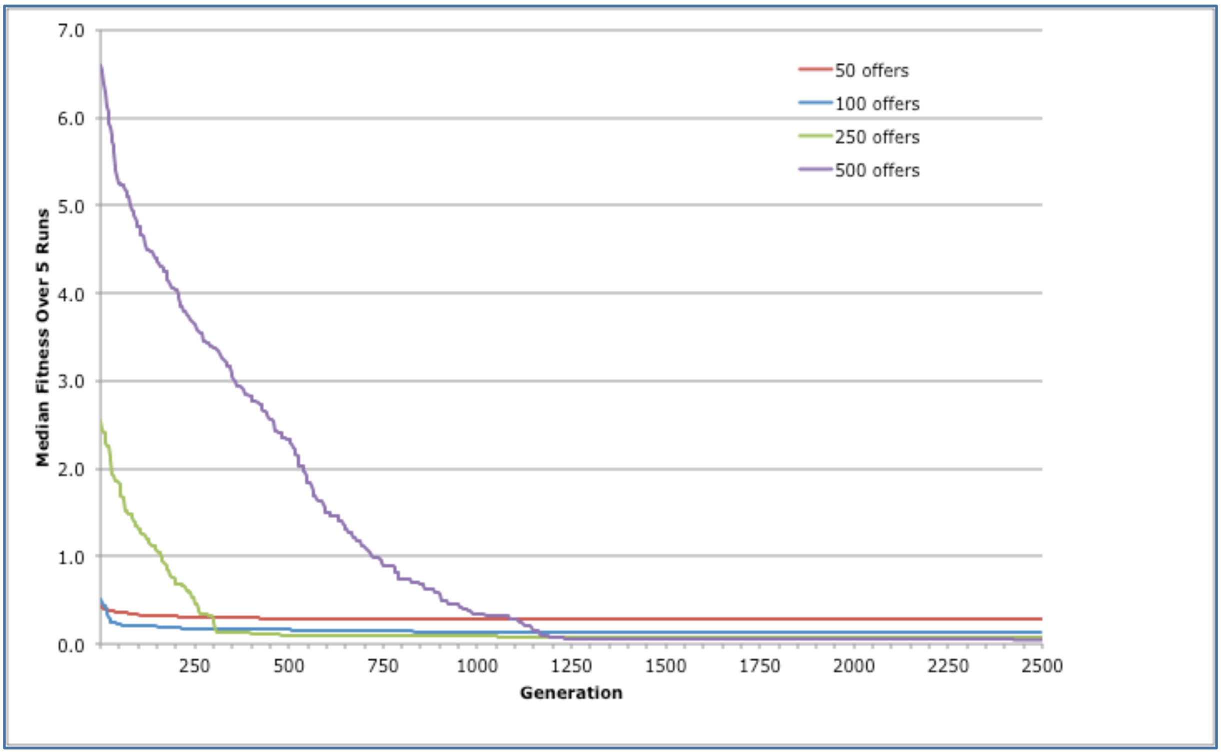

Figure 32: median fitness by generation for each data set in Test 3.

The relationship between the optimum number of generations and the offer pool size is not, however, linear (see Figure 33). This chart shows the number of generations needed for the rate of change in the fitness value to fall to < 0.003 against the size of the offer pool. The threshold chosen is to some extent arbitrary but is informed by the curves in Figure 27. This shows that the slope of the curve actually increases with increasing offer pool size. This suggests that as the size of the offer pool increases relatively more additional generations will be required to obtain an acceptable solution. The curve shown in Figure 33 has few data points and it would be useful to run Test 3 with other offer pool sizes to obtain more data points and a more accurate picture of the relationship. Unfortunately this has not been possible due to time constraints and is a suggested area for further research. The curve as it is shown can be used as an estimate for the optimum number of generations, with a population size of 50, for different offer pool sizes between 50 and 500.

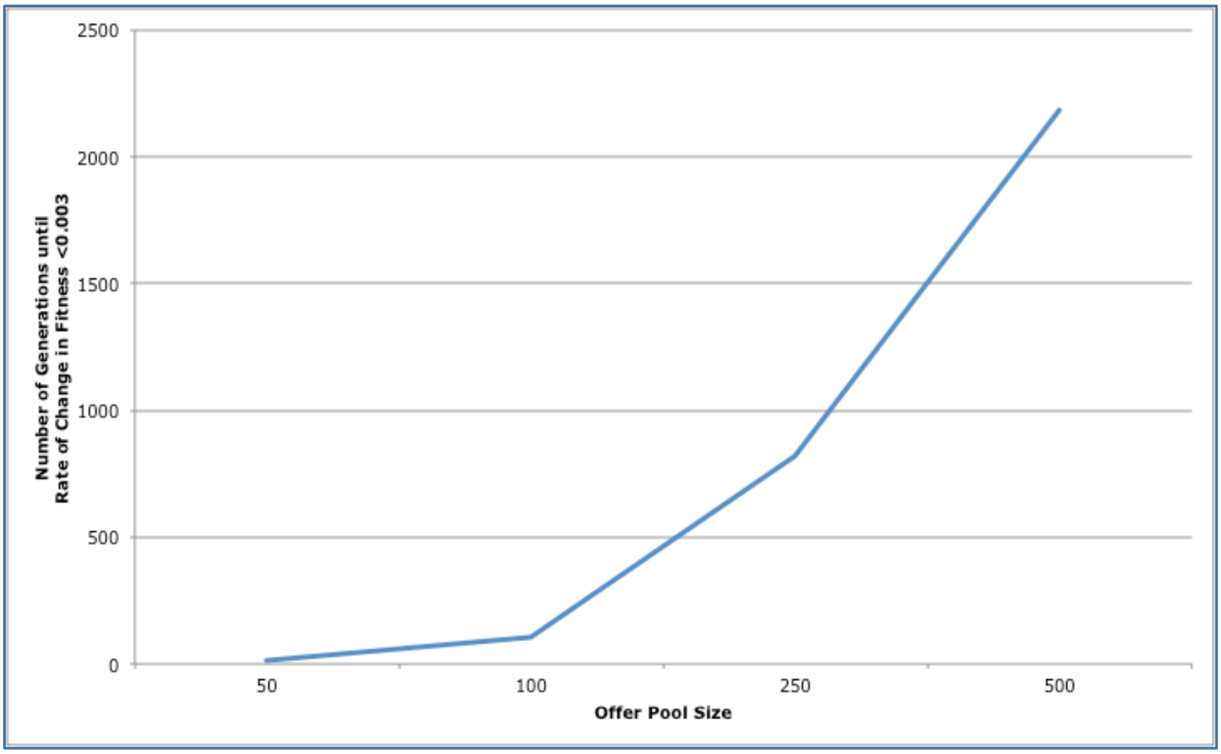

Figure 33: relationship between the required number of generations to achieve an acceptable fitness and the offer pool size derived from the results of Test 3.

Restricting the number of generations potentially releases processing time that could be used to perform multiple runs. This is only useful if by so doing the algorithm will consider more of the available search space and is more likely to find a better solution. This would be expected given the random starting point and is supported by the test results obtained in Test 1. Figure 34 shows that just over 4,000 solutions are considered in each run. If there are 250 generations of 50 chromosomes then the maximum possible search space is 12,500 solutions. It is an interesting observation that each run consistently considers approximately the same number of chromosomes, however, further analysis shows that across all ten runs 41,218 unique solutions were considered indicating that there is very little replication of solutions by run. This greatly supports the conclusion that it is better to restrict the number of generations and employ multiple runs to use the available processor time rather than letting the algorithm run for a large number of generations with very little incremental improvement.

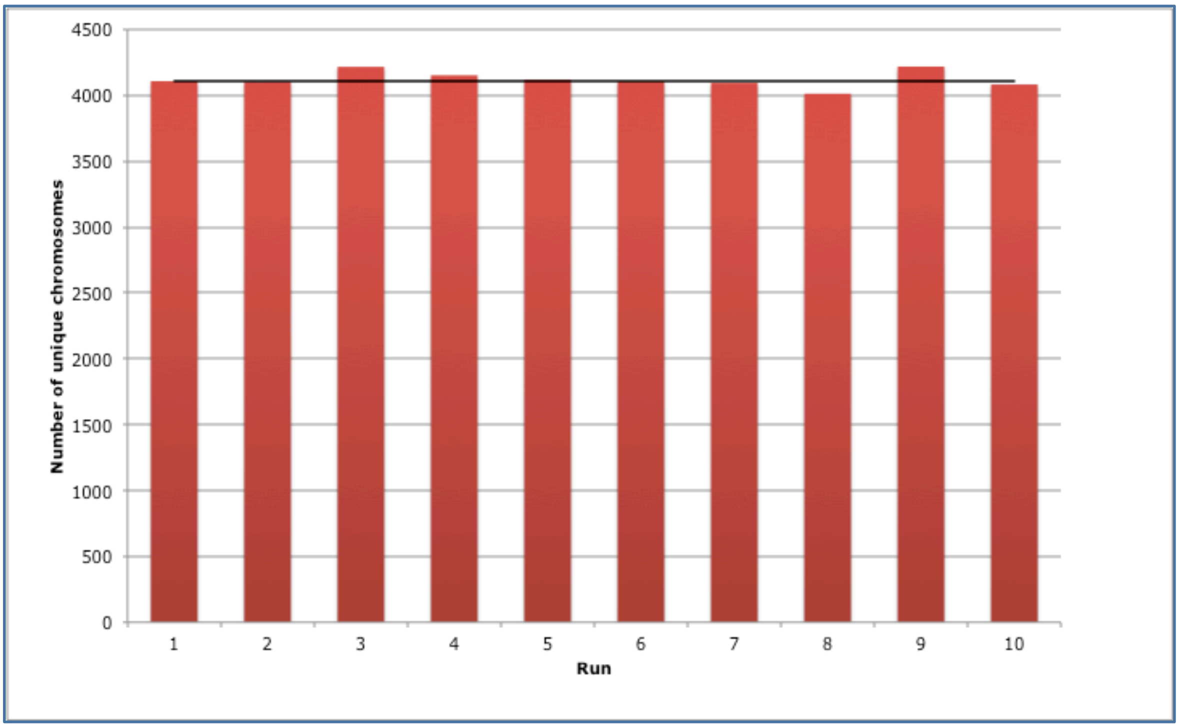

Figure 34: number of solutions considered in each run in Test 1
(median shown as a solid black line).

The final aspect of performance to be considered is the time taken to complete a run. This is highly dependent on the computer but some benchmarks are provided here for the computing configuration used in these tests. The durations of the test runs in Test 3 (defined as the time taken to reach a rate of change in fitness value of < 0.003) were recorded and are reproduced in Table 17. Although the run duration increases as the size of the offer pool increases it is not a simple multiple of the offer size. A single run against the 500-offer data set takes approximately twice as long as that for the 50-offer data set even though the pool is ten times bigger. This suggests that even for very large pool sizes the algorithm still achieves adequate performance. However, a note of caution is necessary when considering execution times, as it is difficult to estimate the contributions made by calculation, database access, I/O operations, etc. It is therefore sensible not to infer too much from this simple test about how the processing time scales with offer size.

Table 17: execution times for Test 3.

No. of Offers	Run	Duration (seconds)	No. of Offers	Run	Duration (seconds)
50	1	78.365	100	1	90.395
	2	80.903		2	85.622
	3	80.160		3	80.121
	4	76.348		4	79.945
	5	75.589		5	84.316
	Ave.	78.273		Ave.	84.080
250	1	110.997	500	1	161.244
	2	107.448		2	162.815
	3	106.092		3	156.811
	4	107.220		4	162.693
	5	114.318		5	177.843
	Ave.	109.215		Ave.	164.281

In order to try to characterise performance a number of tests were run in which some of the parameters governing the algorithm were varied. Of particular interest is what effect varying the population size, crossover probability, mutation probability, or selection threshold has on the algorithm's performance. To establish this a number of tests were run using the 250-offer version of the TD2 data set with a target power of 1000 kW. In each case the fitness threshold was set to zero and $s_{\max}$ to one. The test configurations are shown in Table 18 and the results in Figure 35.

Table 18: performance testing configurations.

Test	$s_{\min}$	N_p	p_x	p_m
A	0.8	50	0.15	0.25
B	0.8	100	0.15	0.25
C	0.8	150	0.15	0.25
D	0.8	50	0.5	0.25
E	0.8	50	0.5	0.5
F	1	50	0.15	0.25
G	1	100	0.15	0.25

With reference to Figure 35 it is clear that Tests A to E all converge with each other after approximately 275 generations. In these tests $s_{\min}$ is held at 0.8 and N_p, p_x and p_m are varied. Tests A to C have constant crossover and mutation probabilities and show that as the population size is increased from 50 to 100 to 150 (Tests A, B and C respectively) there is a broadly similar response for this size data set (250 offers). Increasing the probability of crossover (Test D compared with Test A) suppresses the dynamic range of the initial variation but after the first few generations results in a similar response. Increasing both the probability of crossover and mutation (Test E) further suppresses these initial excursions, but thereafter provides little gain. There seems therefore for this data set to be little value increasing the population size beyond 50 or increasing the probabilities of crossover or mutation.

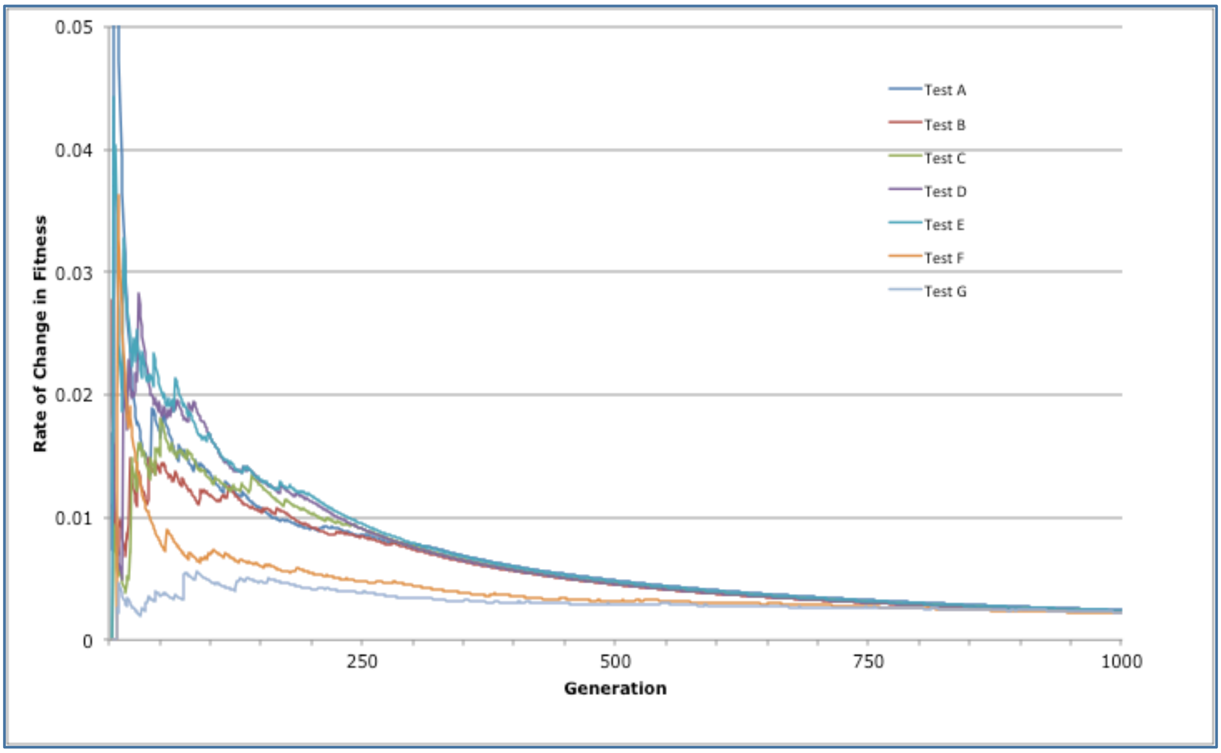

Figure 35: results of performance testing with the TD2, 250-offer data set.

All of Tests A to E ultimately converge on roughly the same value. However, if the selection minimum threshold is increased (Tests F and G) then the algorithm starts to find fitter solutions much quicker. This value of $s_{\min}$ maximises diversity in the population by preventing the duplication of chromosomes in the following population. With a value of 1 every chromosome will be carried forward into the next generation before the application of mutation and crossover. The best performance is provided by Test G, which has a population size of 100. These results indicate that the critical parameters are the population size and the selection threshold. Further they indicate that the best performance is achieved by avoiding duplicate chromosomes in the following generation (i.e. $s_{\min} = 1.0$).

This is an interesting result and perhaps indicates something about the nature of this problem that is different to that considered by Zhang et al. As has already been mentioned, in this problem there are likely to be a large number of solutions that are close to the global optimum and therefore a broad and flat distribution of fitness values. It is therefore better to follow many paths rather than concentrate on only a few, as would happen with a lower value for the selection threshold.

To test this assertion, and to understand if a larger data set requires a larger population size, tests were run on the TD2, 500-offer data set as shown in Table 19. The results are shown in Figure 36. Again the performance is shown to improve as the population size is increased from 50 (Test H) to 100 (Test I). Increasing the population size beyond 100 (in this case to 200 as in Test J) shows a slight decrease in performance in the early generations but slightly better performance for the next 100 or so generations. The greater population size does not seem to offer a sufficiently superior performance to justify its use.

Table 19: performance testing configurations
with the TD2, 500-offer data set.

Test	$s_{\min}$	$s_{\max}$	N_p	p_x	p_m
H	1	1	50	0.15	0.25
I	1	1	100	0.15	0.25
J	1	1	200	0.15	0.25

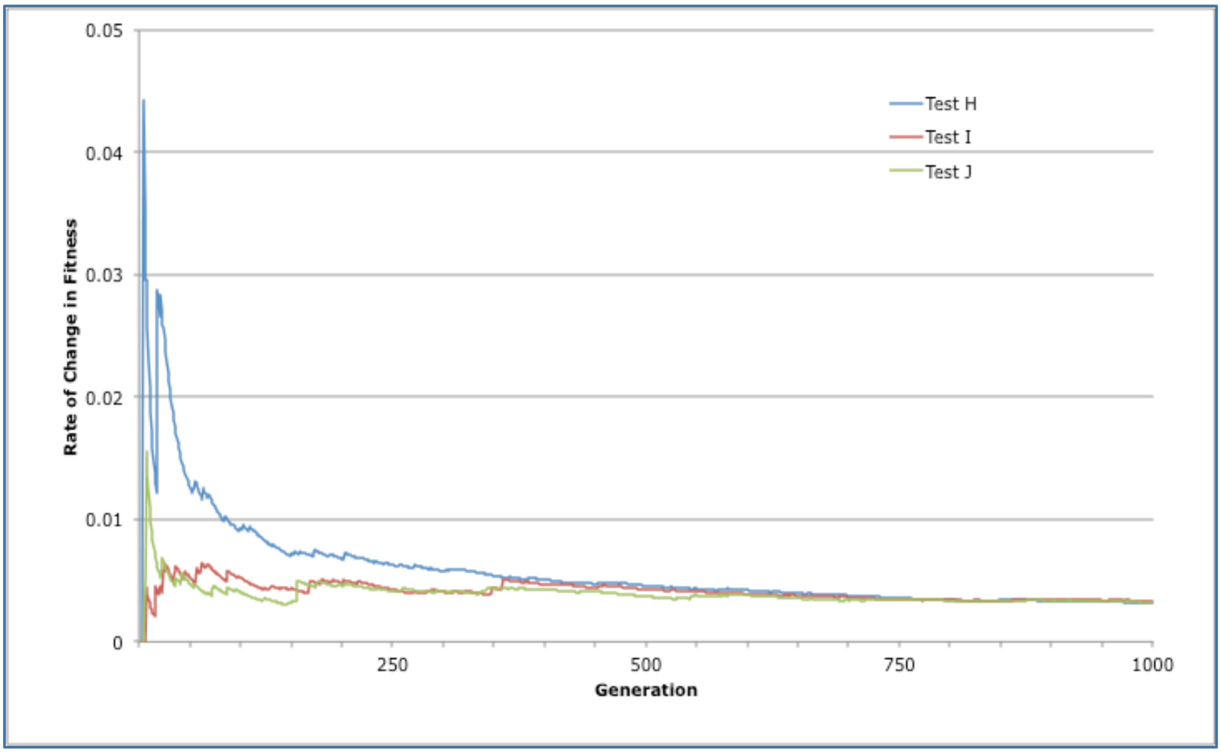

Figure 36: results of performance testing with

the TD2, 500-offer data set.

It is also interesting to look at the execution times for each of these tests as shown in Figure 37. This shows that in this test environment the execution time is influenced by the population size: if the population is doubled then the execution time approximately doubles too (for example, compare Test A with Test B). Larger data sets mean larger chromosomes. For example, Test I is the same as Test G but has double the offers. This test takes ~2.8 times longer to execute. These results indicate that it is critical to restrict both the population size and the offer pool size to maximise performance, as one would expect, as increasing either will result in increased access to the database used to hold them. This is a function of the implementation used here and indicates that care should be taken in this aspect of any implementation, perhaps by using more in-memory data thereby reducing the number of discrete database accesses.

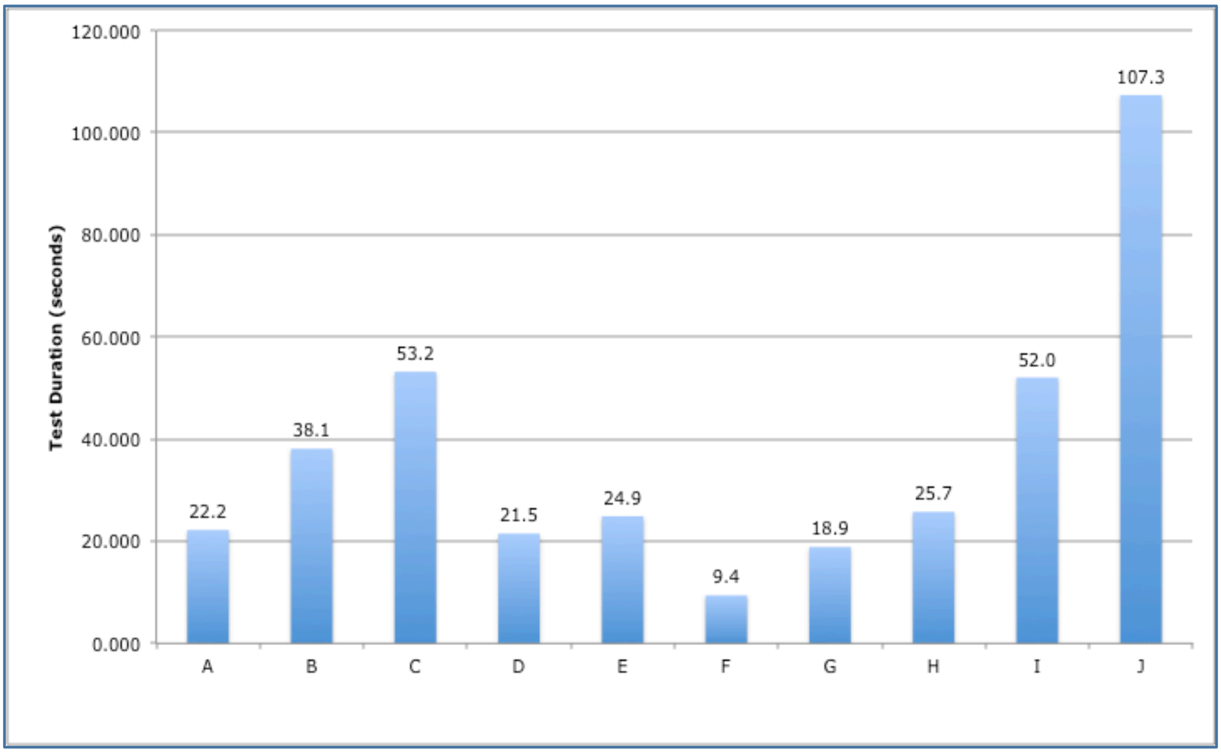

Figure 37: execution times for the performance tests.

In summary, the results obtained from the testing done here suggest a good choice of parameter values is as follows:

N_p	100	p_m	0.25
$G_{\max}$	600	p_x	0.15
$s_{\min}$	1.0	$s_{\max}$	1.0

6.3 Summary and Conclusions

The algorithm as described in Chapter 5 was implemented in a commercial, off-the-shelf computing environment for testing. Test data were created by considering the characteristics of participant loads within the DD-FD demonstrator to ensure that it is representative. Data sets of different sizes were created to facilitate comparative analysis with respect to performance.

The first test is to verify the operation of the fitness function upon which the search depends. Testing on a data set in which all the possible solutions had been calculated showed that the fitness function correctly ranks candidate solutions.

Further testing aimed to investigate accuracy, repeatability and performance. The results of testing the algorithm show that the algorithm successfully converges on an optimal solution after several hundred generations. The actual number of generations required to achieve convergence varies according to the size of the offer pool but with pools of up to 500 offers it does not exceed ~ 600 generations. Allowing the algorithm to run for longer does improve the answer but incremental improvements are so small as to not justify the additional processing time. Independent runs are shown to consider different portions of the search space and it is better to undertake a number of separate runs with constrained generations, and then select the best solution across all the runs. This is a more effective strategy and produces a better answer than one, long run.

The testing has demonstrated the ability of the algorithm to find a known optimal solution in a data set sufficiently small to permit an exhaustive search. The algorithm successfully found the global optimum more often than any other solution and found one of the best five solutions more than half of the time. Whilst it is not possible to be certain that this accuracy will be maintained with other, larger offer pools there is no reason to assume it will not, so long as the algorithm is allowed to run for sufficient time. The best stopping criterion is shown to be using the rate of change of fitness rather than an absolute fitness value.

With the test data used here having offer pools varying between 50 and 500 offers, a population of ~ 100 is shown to provide the optimal balance between speed and rate of convergence. The probabilities of crossover and mutation are also shown not to significantly alter the rate of convergence. The selection threshold minimum, however, does have a significant effect. In these results it is shown that maintaining the selection threshold minimum at 1.0 significantly increases the rate of convergence as opposed to a lower value of, say, 0.8. This indicates that for this class of problem maintaining diversity within the chromosome population is important for maximising the effectiveness of the search.

The algorithm has been shown to provide good performance even on moderately powerful consumer computing hardware; completing 1000 generations on a 500-offer data set in just 52 seconds. However such assessments of processing speed are highly dependent on the implementation and care must be exercised in inferring future performance. In this application frequent access is made to a SQL database, which is likely to have a significant impact on performance. Further improvements could be obtained by implementing the algorithm using more efficient database techniques and by adopting a multi-threaded architecture.

Chapter 7

A Methodology for Modelling Structural Barriers Within Industrial Systems using SysML

7.1 Introduction

The analysis in Chapter 3 identified the need for both a demonstration to address the perception of risk associated with FD technology, and also the need to identify where structural constraints will lead to implementation barriers. The hypothesis is that because the use of demand side services for balancing requires a change to the "predict and provide" operating paradigm then it is very likely that the existing industrial structure cannot easily accommodate such services as a core operational component. The electrical power industry displays evidence of tight coupling to government policy, risk aversion and conservatism. It is resistant to technological discontinuities such as that represented by FD. To find a way of overcoming these obstacles it is likely that lessons may be learnt from other disciplines. In particular because the problem is likely to be structural then it is necessary to consider it holistically.

In this chapter an initial outline for an approach to the problem using Systems Engineering is presented. This research provides the foundation for the development of a structured approach to modelling stakeholder relationships within complex industrial organisations such that the analyst can ensure that all relevant relationships and constraints are considered; and that incentives aimed at eliminating barriers are appropriate. The latter point is essential to reduce the risk of the creation of unintended consequences (undesirable emergent behaviour in Systems Engineering terms) that may act to undermine the policy objectives.

Although the research presented here provides the methodological foundation it requires further development and testing to provide the full analytical capability required. Such development forms the basis for future research in this area.

7.2 The Requirements of an Analytical Framework

The industrial system that is the GB electricity system is a complex system formed from a web of interactions between a large number of stakeholders and assets. Socio-economic rules as well as technical constraints govern these relationships. Socio-economic rules can be subjective and ambiguous and are often a complex mix of legal rules and social constraints driven by economic and social policy, together with cultural convention. Such a complex picture is not uncommon when considering enterprises. To understand barriers it is therefore essential to understand what is already in place. The analysis framework must be able to capture both the socio-economic rules and the technical constraints. This analysis must be holistic and sufficiently detailed such that all the points at which the new service must engage can be identified and characterised. To analyse an industry and how it operates is a major undertaking. Therefore to make the problem tractable the research presented in this chapter develops a methodology based on the use of the DD-FD project as a case study. The objective is to make this methodology sufficiently generic that it can be applied to the wider industry and, potentially, in other industrial contexts also.

The aim is to develop a methodology for modelling relationships between stakeholders in such a way that when new services are introduced the effects on these relationships can be identified. Ways of mitigating effects that create barriers can then be developed and tested on the model such that efforts to lobby stakeholders for change can be supported with evidence. This is especially important when such change might require, for example, primary legislation. Such changes have wide-ranging impact and require substantial stakeholder investment. A resistance to change without evidence in support of it is therefore understandable.

7.3 A Review of Literature on Modelling Stakeholder Relationships

There is considerable literature in the area of organisational modelling and there are a number of recognised frameworks. These include:

- Zachman Framework (Zachman, 2008)
- System Dynamics (Sterman, 2000)
- Role Activity Diagramming (RAD) with or without Process Architecture Diagramming (PAD) (e.g. Ould, 2004; Ould, 2005)
- Soft Systems Methodology (SSM) (Checkland, 1998)
- Unified Modeling Language / Systems Modeling Language (UML / SysML) (Jacobson et al., 1994; Kruchten, 2000; Friedenthal et al., 2008)

Some of these are more complete than others. For example, Zachman explicitly states that his framework is an ontology, or structure, for describing an enterprise but is not a methodology (which, as he states, is a process) (Zachman, 2008). Some frameworks try to be as flexible as possible without imposing rigid ontologies, within which the modelling technique must be exercised, e.g. SSM (Checkland, 1998). Others attempt to provide a coherent and consistent ontology for its description, e.g. RAD / PAD and UML / SysML. Similarly, some frameworks focus on only structure (e.g. the Zachman Framework) and others only on behaviour (e.g. System Dynamics

(Hitchins, 2010)). It is possible to contrast the different capabilities of these frameworks using these criteria as shown in Table 20.

Table 20: comparison chart for alternative modelling frameworks.

Framework	Structure	Behaviour	Ontology	
			Strict	Flexible
Zachman	✓		✓	
System Dynamics		✓		✓
RAD / PAD		✓		✓
SSM	✓	✓		✓
UML / SysML	✓	✓	✓	

To effectively model relationships it is necessary to capture both structure (who is connected to who and the static description of the relationship) and the behaviour (the dynamic description of the relationship). From Table 20 it is immediately apparent that Zachman, Systems Dynamics and RAD / PAD cannot offer a solution in isolation as Zachman captures only structure and not behaviour whilst the opposite is true for both Systems Dynamics and RAD / PAD. Only SSM and/or UML / SysML can capture both, the difference between them is whether a strict or flexible ontology is required. Both of these techniques have their foundations in Systems Engineering lending credibility to the notion that Systems Engineering tools and techniques can be useful in this research context.

The objective of both SSM and UML / SysML is to provide a methodology founded on a holistic approach: the principle that in an ever more complex and inter-connected world the only effective route to a complete understanding is through consideration of the whole within its environment. This, termed *systems thinking*, is a key principle for understanding the operation and structure of complex socio-economic or socio-technical systems (Hitchins, 2003).

SSM is a technique often used as a methodology for guiding this systems thinking (Jackson, 2001). Dirker et al. (2008) used systems thinking in conjunction with SSM to examine a particular aspect of the market for industrial control and instrumentation products. The use of SSM allowed consideration of a high-level view and design of an appropriate channel management methodology for the introduction of a new product. Ferrari et al. (2002) applied SSM to understanding an automotive parts supply business and noted the advantage the methodology provides in capturing behavioural variables. Such a facility could be particularly important when considering stakeholder relationships.

SSM has also been applied to aspects of the energy industry. Fawkes (1987) used it to examine the problem of the management of energy use in buildings, Fielden and Jacques (1998) to consider energy policy in smaller self-contained communities and Neves et al. (2004 and 2009) to examine efficient decision making with respect to the implementation of energy efficiency initiatives. In the latter two papers by Neves et al. SSM has been successfully applied with the aim of informing policy. Fawkes, meanwhile, explicitly notes that SSM provides the starting point and not the final solution.

The point made by Fawkes is important and one echoed by other researchers who have sought to use SSM to define a problem but have then looked to use other modelling approaches to refine the model of that problem. Neves et al. (2009) and Petkov et al. (2007) have combined SSM with Multi-Criteria Decision Analysis (MCDA) whilst others have combined SSM with use-case modelling, a component of the UML modelling approach (Bustard et al., 1998; Bustard et al., 2000). Lane and Oliva (1998) combined SSM with System Dynamics but noted that in their approach

SSM is not being used as a "front end" to System Dynamics. In this paper System Dynamics is used to ensure "dynamic coherence" and is described as a synthesis of the two methodologies.

Whilst SSM provides the flexibility to capture a wide variety of relationships and entities via the use of rich pictures, it does not provide the ability to transcribe these rich pictures into models of greater fidelity that can be used for detailed analysis. In contrast UML (see the Object Management Group (OMG); www.omg.org) provides a very structured language with strong consistency checking that can be used to build a detailed and highly descriptive model.

The OMG has been very active in standardising methodologies and tools to provide a complete suite of tools to assist in the design and deployment of complex IT systems and business processes. The OMG states that it has an objective of unifying a number of aspects of IT-centric systems (Object Management Group, 2010b):

> "UML, along with the Meta Object Facility (MOF™), also provides a key foundation for OMG's Model-Driven Architecture®, which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution."

An important aspect of UML is that it is a visual language. By using diagrams and supporting text it supports the modelling of, and communicating about, systems (Si Alhir, 2003). It grew out of software engineering and a perceived need to use a general purpose language for modelling software systems and how they interact with their environment. However, the language has powerful capabilities that mean that it can be used as a general-purpose tool for modelling processes and relationships between entities in any organisational setting. Indeed proponents of

UML and the Rational Unified Process (RUP) argue that UML can be used to model businesses without any other framework (Eriksson and Penker, 2000; Kruchten, 2000).

However, Odeh and Kamm (2003) argue that there are dangers in trying to superimpose a software engineering framework onto business processes. In particular, Odeh and Kamm take issue with the idea of business engineering through use-case modelling. They criticise the approach adopted by Jacobson et al. (1994) as this approach does not take account of interactions between actors that take place separate to interactions with the IT system. Their viewpoint is centred on the system and in such a viewpoint their criticism is valid. However, if the viewpoint is shifted to that of the process then there seems little reason to assume that interactions between actors – as valid components of the overall process – cannot also be captured using UML. This is particularly relevant when UML is being used without any obvious IT system as the focus.

In Bustard et al. (1998 and 2000) the symbiosis of SSM in support of business improvement and use-case modelling in support of user-centred functional modelling is explored. They conclude that much is gained from the synergy of the two approaches; both as a more complete description of the problem and also as a crosscheck on the results of each approach. They propose an overall approach of "describing system behaviour through scenarios expressed as event or action sequences" which improves UML analysis using an SSM front-end analysis. In their case the conclusion is that SSM linked to UML via scenarios is a valuable means of enhancing UML.

Maij et al. (2002) reached similar conclusions when modelling the use of ICT in healthcare. In their study they allied UML to the Dynamic Essential Modelling of Organizations (DEMO) methodology. DEMO was used as the front-end analysis tool with UML use-cases used to describe the interactions with ICT systems. They note that combining the two approaches is not "mechanical" noting "end-users play a vital part in the process of linking the DEMO models with the use-cases". This then indicates a need for bridging the gap between the formalised and standardised UML, together with the tool support available for it, and the less formalised and less standardised approaches to business modelling, such as SSM and DEMO.

Wiering et al. (2004) do not dismiss the use of UML alone for business modelling but do acknowledge its inaccessibility for non-IT specialists and the need to couple the UML world with the business-modelling world. Their proposal is also to supplement UML but their objective is slightly different. They aim to make the UML toolset more accessible rather than necessarily address its weaknesses when applied to business modelling. Their solution is the ArchiMate metamodel that can be translated into UML. This provides tools for modelling business organisations at a level of abstraction higher than that available within UML, but in a way that can be readily translated into UML.

The work described above seems to support the view of Odeh and Kamm that UML alone is insufficient for business modelling. However, this seems to be more a problem of presenting the technical details of a UML model in an abstracted form that non-IT specialists can comprehend, rather than deficiencies in UML *per se*. Some take the approach of using another modelling tool prior to UML, such as SSM or DEMO, whilst others choose to extend UML by providing an abstraction layer such as ArchiMate to hide the detail.

From the literature it seems clear that there has been little success in applying UML alone to the domain of organisational modelling. In the cases highlighted above it was necessary to augment UML with other more flexible approaches to overcome the constraints that the precise semantics entail. However, one method that appears absent from the literature is to extend UML itself with capabilities to overcome the limitations identified above, when applied to this type of problem. The language explicitly allows its extension via profiles. Recognising UML's deficiencies when applied to more general systems problems than those typically encountered in IT, the OMG and the International Council on Systems Engineering (INCOSE) published in 2003 a set of requirements for extending UML to support systems engineering. This resulted in 2007 in the SysML specification (Object Management Group, 2010a).

This extension of UML would seem to offer all the advantages of UML (tool support, precision, consistency, etc.) but without the limitations that the authors cited above have encountered. However, experience of using SysML for organisational modelling seems to be absent in the literature. This is possibly because of the relative youth of the SysML specification. Whilst SSM offers a tried-and-tested approach to modelling organisations, it cannot provide the detail necessary to move to the system implementation phase. SysML, however, appears to potentially offer the capability the OMG envisions when considering business modelling. It provides the strict semantics of UML but with extensions to provide the flexibility to capture human relationships as well as technological ones (Friedenthal et al., 2008).

7.4 Barriers as an Emergent Property of Relationships

The implementation of any new product or service requires interfacing its technology, commercial framework and business processes with what is already there – at least cost. Barriers prevent or hinder this implementation. Examples might be the absence of a necessary protocol or interface, a legal provision preventing the formation of a necessary commercial relationship or a contractual provision that also acts to prevent a relationship from forming. In some cases the barrier identified might just involve a renegotiation of a relationship between two contracting parties in order to facilitate implementation, or it might be insurmountable without a policy or legislative change. In the latter case this would necessitate action by the relevant governmental body.

A barrier arises where two stakeholders are required to establish a new relationship, or modify an existing one, in a manner that is incompatible with existing rules or conventions. The barrier can be thought of as a property of the relationship that emerges when an existing relationship is extended, or a new one established, to support a new objective: it emerges from the interaction of the parties within the framework of the relationship and cannot be deduced by looking at either party in isolation. It is at the interface created by the relationship that the barrier occurs.

Relationships are key to the successful implementation of a technology such as FD as it is by its nature a service that will touch on a number of areas of, and stakeholders within, the electricity supply system. It will create new stakeholders, such as the Aggregator, and new relationships. The technology provides a communication channel to consumption nodes and provides a much more granular view of their behaviour, behaviour that can now be actively influenced and controlled in a much more dynamic fashion than was previously possible.

Barriers can prevent or restrict relationships through a number of mechanisms including:

- **Technical:** absence of an agreed format, standard or means for the exchange of an essential commodity (e.g. data).
- **Legal / regulatory:** a rule or regulation preventing two parties from exchanging an essential commodity because to do so would breach some legal provision.

- **Commercial:** a conflict arising from the existence of an existing contract or arrangement that acts to create disincentives to implementation.

Under the categories of technical and commercial barriers it is unlikely that outright barriers to implementation exist. These barriers are more likely to impose constraints that necessitate costly workarounds thereby negatively impacting the business case for the product, service or technology when it is in its most vulnerable, embryonic phase. Resolution might, for example, involve a renegotiation of a relationship between two contracting parties in order to facilitate implementation. In such cases resolution is likely to be entirely within the control of the industry.

Under the category of legal/regulatory, however, it is entirely possible that barriers exist that whilst they might be surmountable the changes necessary might require changes to regulation and/or legislation. Proceeding with implementation may simply not be possible under the legal framework as it is. Resolution of such barriers is only possible through the action of government in response to appropriate lobbying on the part of the relevant industry stakeholders. This is invariably a long and tortuous process fraught with risk. The inability of the industry alone to resolve these barriers means that they create a significant impediment to further development. The objective of this analysis framework is to provide a mechanism by which early insight into such barriers can be obtained such that remedial action can be taken to ease the implementation and reduce cost.

7.5 Using SysML as an Enterprise Analysis Framework to Model Relationships within the DD-FD Project

SysML is a diagrammatic language for modelling both the structure and behaviour of the system under consideration (Friedenthal et al., 2008). Within the application context of this research the *system* is the network of stakeholder relationships from which barriers are expected to arise. Those barriers should become evident as emergent properties within the SysML model. The information to be modelled is composed of both objective, precise information about the interface or relationship and also subjective, potentially ambiguous information.

The structure relates both to who has a relationship with who (entities can be people, organisations or information systems) and also the details of the relationship, such as how it is instantiated. Behaviour relates to how a transaction takes place and is analogous to a protocol in communication systems. Objective information relates to interfaces that are tightly defined such as those between information systems, subjective information relates to looser interfaces that typically exist between people. In such situations rather than explicitly defining the nature of the transaction the intelligence of the parties on either side is heavily relied upon to make sense of it.

The DD-FD system was conceived as part of a collaborative project between a group comprising academic, industrial and governmental partners[15]. The project successfully produced a demonstration FD system capable of controlling HVAC plant installed in a number of commercial buildings within London. The architecture of the FD system is described in detail in Chapter 4 but the demonstration project also provided an excellent case study for the purposes of this strand of research.

[15] Technology Strategy Board project number 200083 started 1st January 2009 and ended 31st December 2010.

The knowledge that is the foundation for the modelling is largely contained within the stakeholders within the industry (in particular the people within the organisations), plus the relevant literature such as written statutes, operating procedures and policy statements. Capturing this knowledge involves a process of literature review (as described in detail in Chapters 2 and 3) and discussions with individuals within the problem domain that have knowledge of the relevant relationships. A stakeholder map was produced to map the relationships and identify the key stakeholders (see Figure 38).

The stakeholder map identifies the principal stakeholders and their roles or key characteristics. For example, the SO (National Grid, one of the project partners) is responsible for system balancing. For it, the key issues are trust that service providers will deliver what they have contracted to deliver, and the business case since it has a statutory obligation to minimise the cost of operating the system. Thus, for FD services to form an active part of its balancing portfolio the services must be reliable and cost competitive with alternatives (principally generation). The SO also has a relationship with other potential FD customers such as energy traders, DNOs and energy suppliers. All of these may compete with the SO for the procurement of FD services according to their own requirements and priorities.

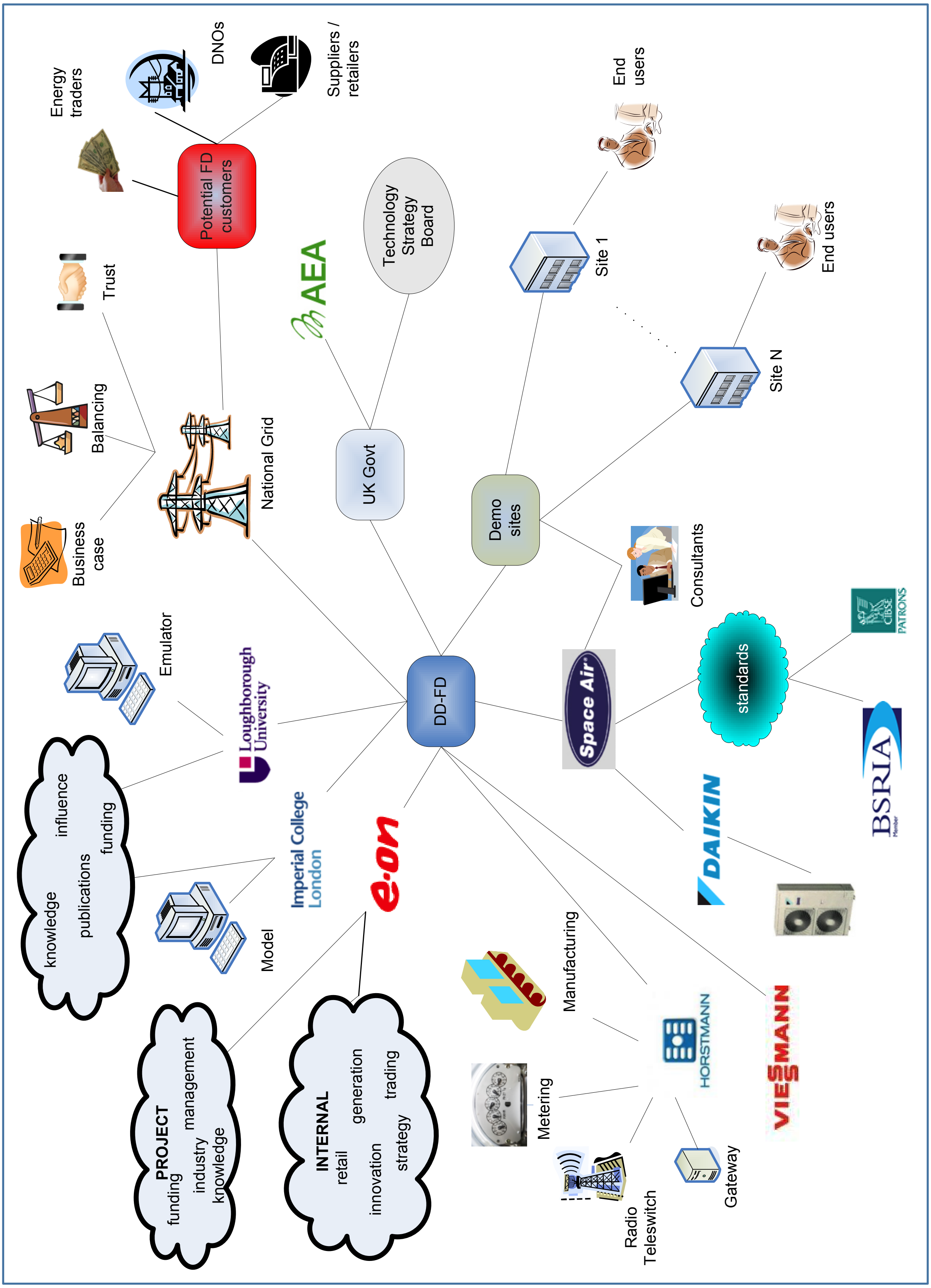

Figure 38: stakeholder map for the DD-FD project.

In the case of the DD-FD project E.ON is the sole supplier of aggregated flexibility. The individual flexibility suppliers are the owners of the assets that are used to provide the flexibility. In this project that is a number of demonstration sites who agreed to have the DD-FD control equipment installed in order to utilise the flexibility available from their HVAC equipment. The end users of the facilities at the demonstration sites are the ones impacted by the use of the assets to provide flexibility. Finally, there are the suppliers of the infrastructure used to provide the services including the assets under control; the communications, gateway and metering infrastructure; and the software models used to characterise the services. Even in such a relatively small technical trial such as this one it is clear that there are a large number of stakeholders and a correspondingly large number of relationships to consider. To make the problem tractable a structured approach is required and this is described below.

SysML extends UML and is therefore also a diagrammatic language. It contains a number of diagram types that together allow the full structure and behaviour of a system to be modelled (see Figure 39). It also includes the ability to model requirements, although that capability is not used here. The white triangle in Figure 39 indicates specialisation, e.g. an activity diagram is a specialised type of behaviour diagram.

Although powerful, the semantics and application of SysML can be confusing. To test the application of the language in this context this research has focused on capturing the existing DD-FD system in SysML based on the stakeholder diagram shown in Figure 38. This provides valuable experience in using the language on a more tractable problem. To identify barriers to implementation the focus of analysis must be the interfaces between entities within the system. It is at the point at which items need to be transferred from one entity to another that incompatibilities and constraints might occur. It is these that give rise to implementation barriers. As the

DD-FD system is a specialisation of a more generic FD system, modelling the DD-FD system might also help illuminate where barriers might occur in a production system.

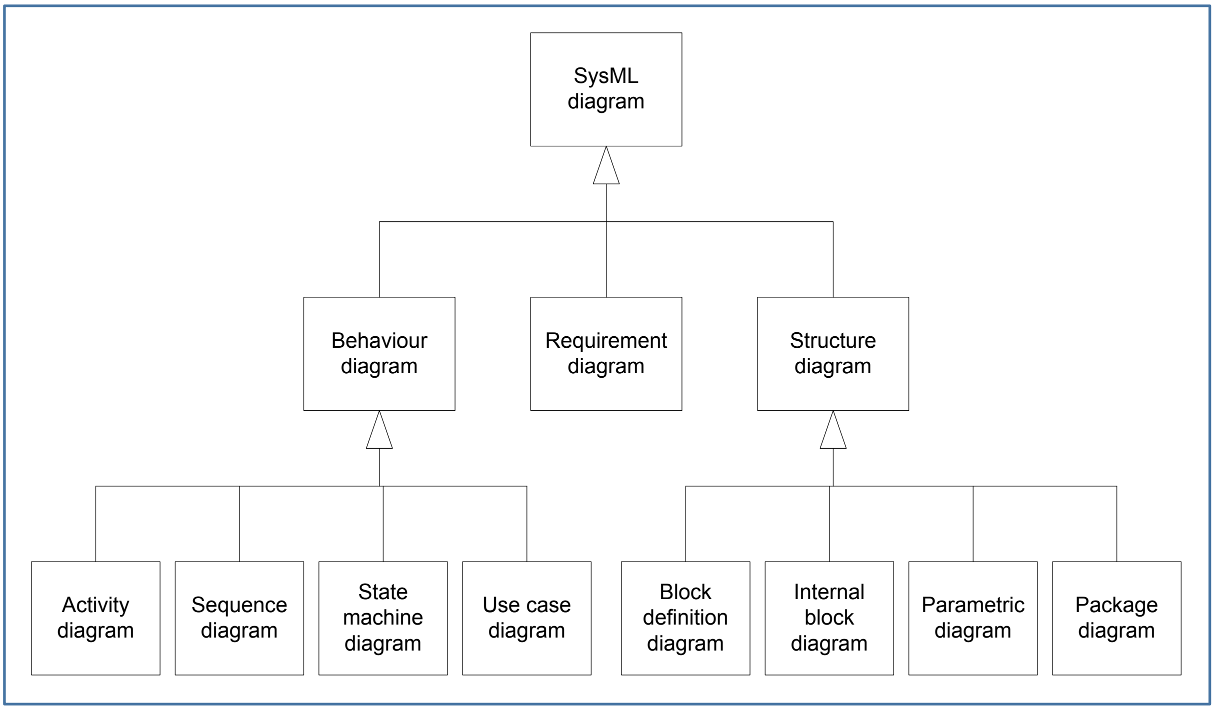

Figure 39: SysML diagram types (Source: Friedenthal et al., 2008).

7.5.1 DD-FD Structure

As discussed above, the objective is to capture both the structure and behaviour of relationships. Modelling structure begins with the block definition diagram (BDD) of the system. The BDD for the structure of the DD-FD system is shown in Figure 40. The diagram shows the principal components of the system and three types of association between them: composition, specialisation and use. A solid diamond on the end of an arrow connecting two blocks shows composition. The diamond is attached to the superior block and the arrow points to the part. This allows a hierarchical composition to be represented. Thus the `ddfdSystem` is composed of four blocks: `ctrlCentre`, `commsNwk`, `demoSite` and `emulator`.

Where no number is shown at the end of the composition association then it is assumed that only one block can be associated. Multiplicity is shown by a * thus a `ddfdSystem` can be composed of a minimum of one and an unbounded maximum number of `demoSite` blocks.

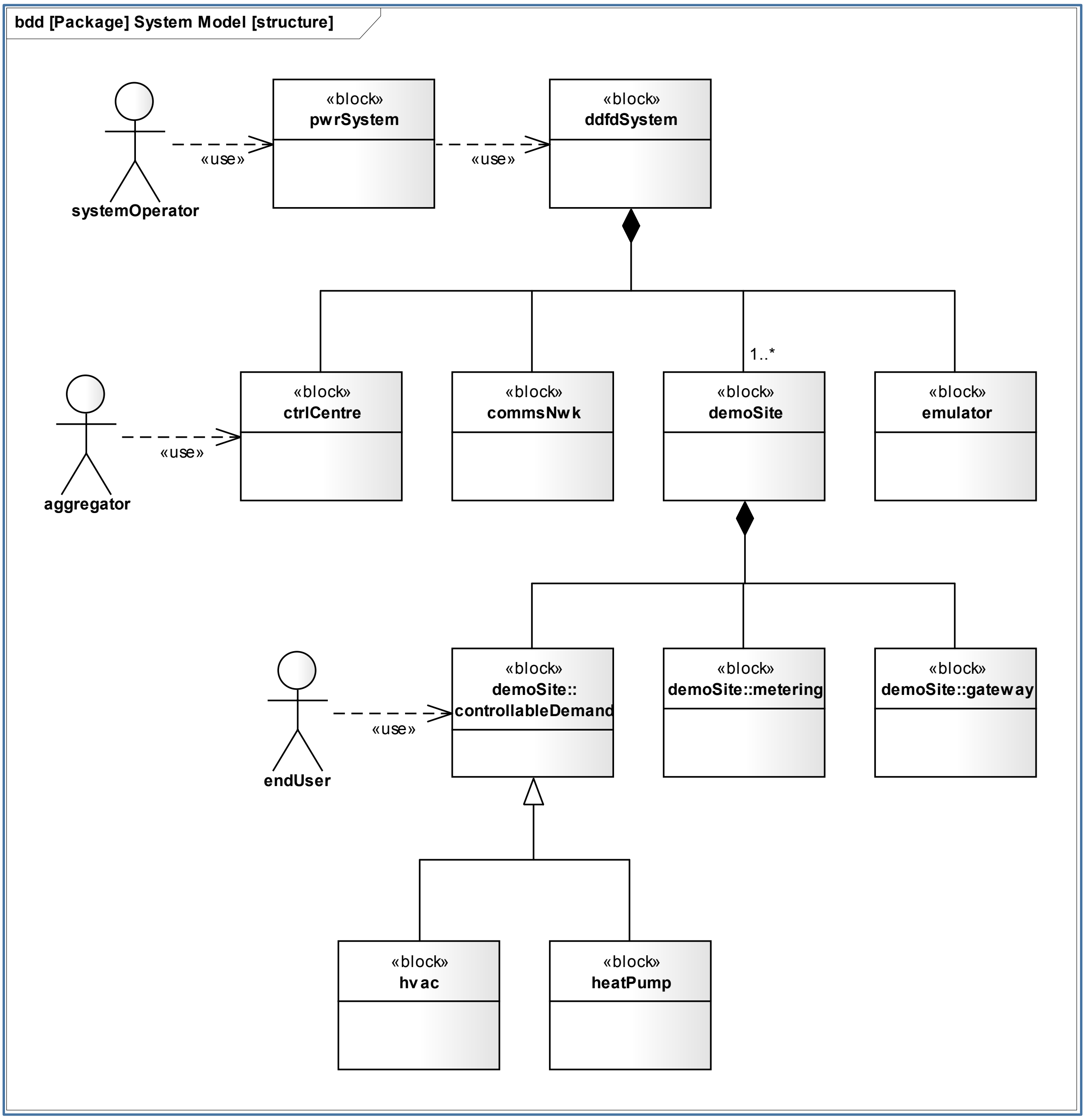

Figure 40: SysML block definition diagram for the DD-FD system.

Specialisation is shown by an empty triangle. The triangle is attached to the more general block. Thus both `hvac` and `heatPump` are specialisations of a `controllableDemand`. Finally, a dashed arrow with the keyword `<<use>>` attached indicates usage. The arrow points from the client of the functionality to the supplier of it. For example, in Figure 40 an `endUser` uses a controllable demand.

7.5.2 DD-FD Behaviour

Functionality determines how entities are used and what they are used for. Further, it determines what must be passed across interfaces. If compatible interfaces cannot be established, or if it is not possible to pass across an interface something required by the functionality, then this will constitute a barrier to implementation. So the next step is to capture the relevant behaviour in terms of use cases. The DD-FD system has two principal external actors:

- The `systemOperator` that balances the electricity system.
- The `endUser` who is subject to the effects of FD events.

The SO must balance the electricity system by exercising functionality provided by the electricity system to control both generation and demand. The DD-FD system provides the SO with the ability to change the power demand of controllable loads within the DD-FD system. The SO does this by issuing power change requests in response to offers to do so made by the DD-FD system. These two use cases are shown in Figure 41.

The use-cases exercised by the SO both include other use cases internal to the system. The `makeOfferToSO` use case includes the use case in which the control centre uses functionality provided by the gateway to get an offer from each site. Similarly the `chgPwr` use case includes the use case in which the control centre uses functionality provided by the gateway to initiate an event at a specific site. Figure 41 shows one additional use case: `abortEvent`. This is the only functionality that can be exercised by the user located at a site. It is included within the power change request use case as it is also possible for the SO to request termination of an event via the Control Centre operator.

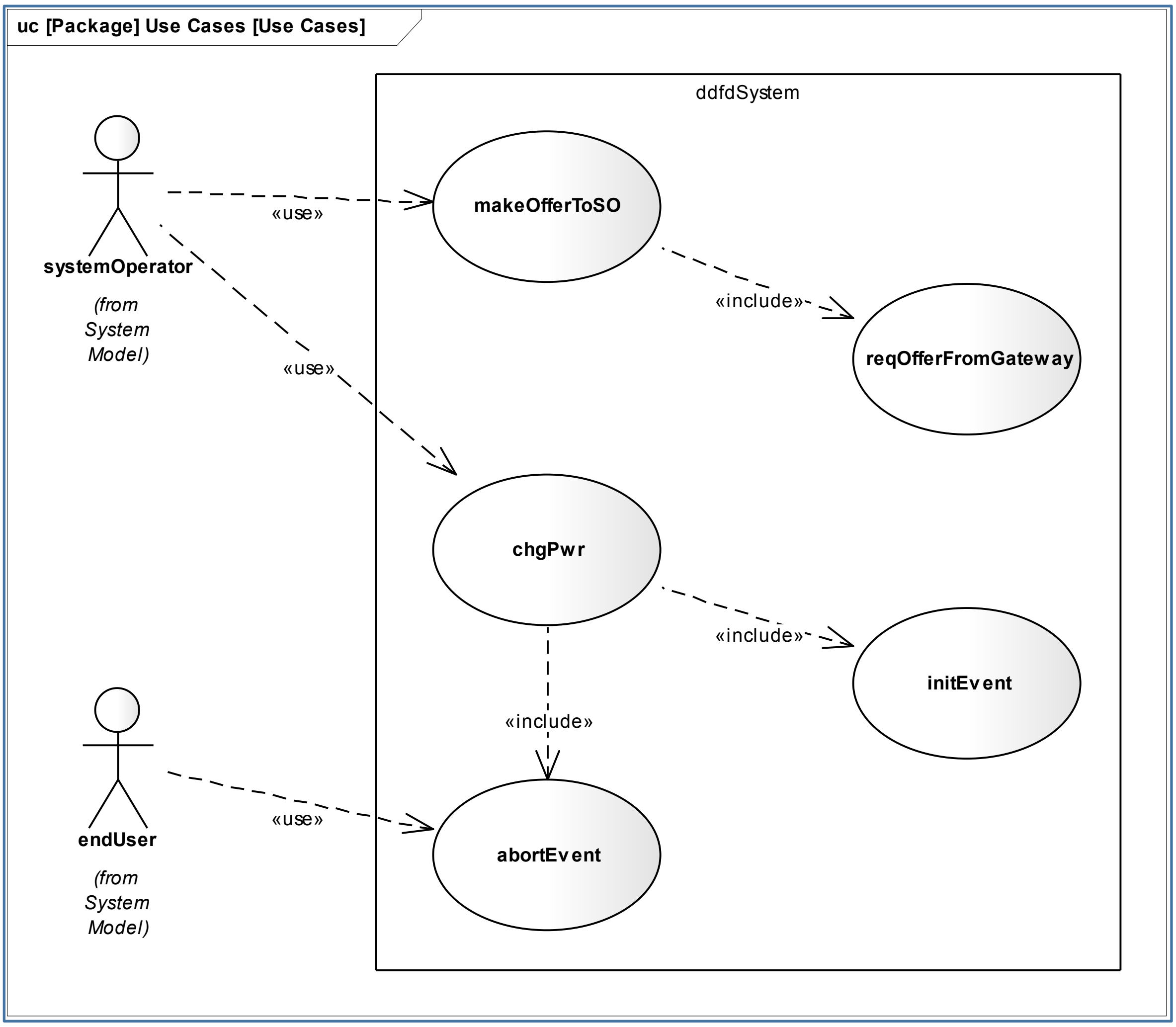

Figure 41: main use cases within the DD-FD system.

7.5.3 Modelling Relationships

Relationships can be between people, between machines or between people and machines. It is expected that those interfaces involving machines will be the easiest to capture, as the machines will likely dictate quite strict rules governing the interaction. Items flow across interfaces. These items could be energy, money, information or almost anything. If a technical, commercial or regulatory incompatibility prevents that flow then this highlights a barrier to implementation. The nature of the flow is characterised by the functionality that is expressed via the interface.

By considering the use cases of Figure 41 and how they might be implemented on the structure shown in Figure 40 it is possible to determine the key interfaces within the system. For example, the `makeOfferToSO` use case requires the following steps:

1. Measure the power.
2. Calculate an offer.
3. Send the offer to the Control Centre.
4. Consolidate offers within the Control Centre.
5. Calculate an aggregate offer.
6. Send the aggregate offer to the SO.

Within this sequence of events are a number of relationships instantiated over interfaces; each of which transfers one or more item(s) between entities. They are summarised in Table 21.

Table 21: interfaces used in the `makeOfferToSO` use case.

A End	B End	Class
Meter	Gateway	M2M
Gateway	Control Centre	M2M
Control Centre	SO	M2P

Using SysML the structure of the interfaces can be captured using an internal block diagram (IBD). In an IBD the transfer of items is captured using an *item flow*. For example, the interfaces between the meter and gateway are internal to the `demoSite` block. The IBD for this block is shown in Figure 42.

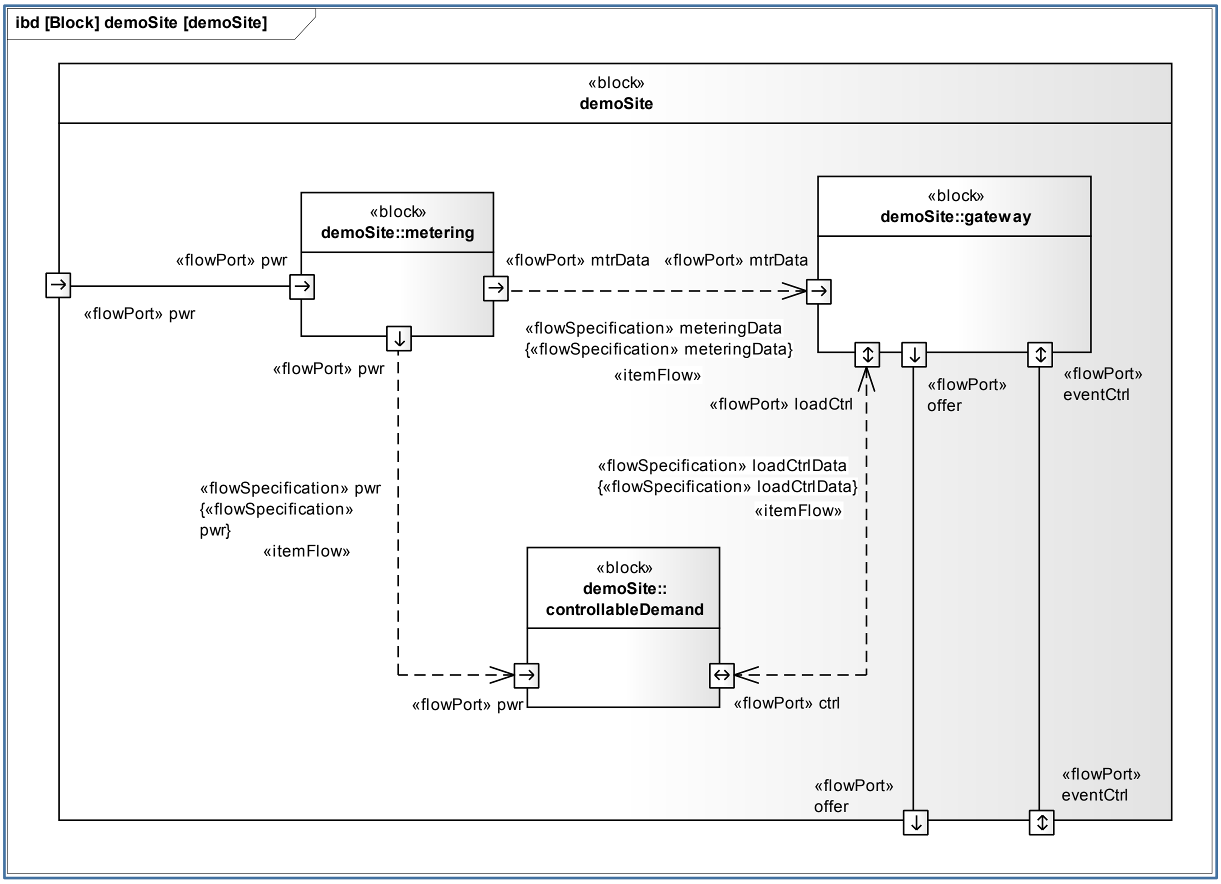

Figure 42: internal block diagram for the demo site block.

From Figure 42 it is possible to identify the external interfaces to the demo site and what item(s) are required to be transferred across them. There are two interfaces into the demo site: one for the power feed and one for the communication signal to control an event (from the use cases this control signal must either initiate or abort an event). There is also one interface out of the demo site: a communication signal containing the offer and the power consumption data.

The above discussion is helpful in capturing the detail of how the DD-FD system works and demonstrates how the technical, i.e. machine-to-machine (M2M) or machine-to-person (M2P), relationships can be modelled. It does not, however, provide an insight into how to capture the relationships between people. These are an integral part of the operating domain of the system and a potential source of barriers. As discussed in the literature review earlier in this chapter, little evidence of the application of SysML to the modelling of such relationships could be found in the literature. To gain insight into the language's applicability an example follows. In the following discussion the modelling of a value chain is considered. This involves two relationships: between the SO and the Aggregator, and between the Aggregator and the Flexibility Provider. Referring back to the stakeholder map of Figure 38, within DD-FD these are National Grid, E.ON and the end users respectively. As the DD-FD project did not include commercial arrangements the following example is hypothetical.

The overall value chain is shown in the BDD in Figure 43. This shows that the value chain is composed of two commercial relationships. A commercial relationship is likely to be quite generic so in this model the two separate relationships are properties of the `valueChain`, and are typed using a generic commercial relationship defined as the block `commRelationship`. As the commercial relationship is between legal entities they are shown as actors with a reference association to the `commRelationship` block. In the relationship there is always a client and a supplier. The supplier supplies some tradable item to the client via the commercial relationship and the client compensates the supplier for it. This is usually with money but does not always have to be so. It could be some other item on which the supplier places a value. This implies that there are two separate item flows that form the commercial relationship. Those item flows will have various rules governing their structure and behaviour. Some rules are negotiated between the client and the supplier whilst an external authority imposes some upon them. For example, in a retail relationship where a consumer (the client) buys something from a shop (the supplier) there is an externally imposed requirement for the shop to add VAT to the transaction.

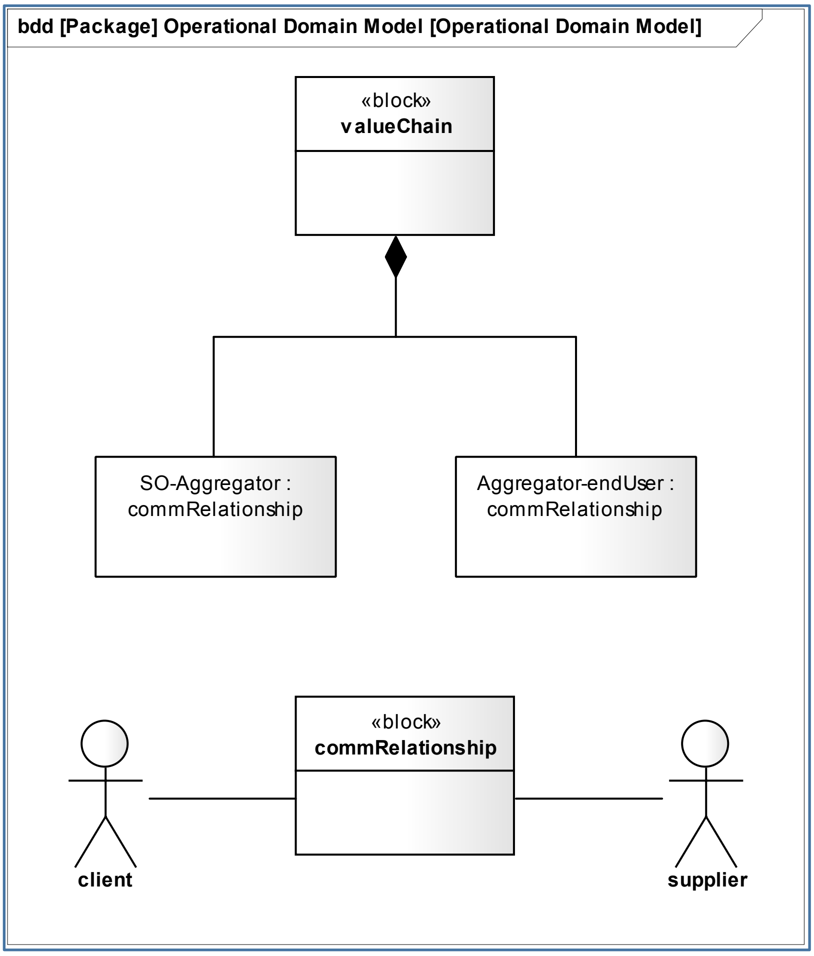

Figure 43: SysML block definition diagram for the value chain.

The reason to use properties rather than specialisation is that it is expected that the internal structure and behaviour of the two relationships will be the same; only the attribute values are expected to differ. This may not be a correct assumption but it forms a good working assumption. The next step is to consider the internal structure of a commercial relationship. This is shown using an IBD in Figure 44. The relationship contains two parts: one representing a mechanism for executing a financial transfer and one for an offer transfer. These could be further decomposed as necessary to provide greater detail as to how the transfers are accomplished and the exact detail of the items that flow.

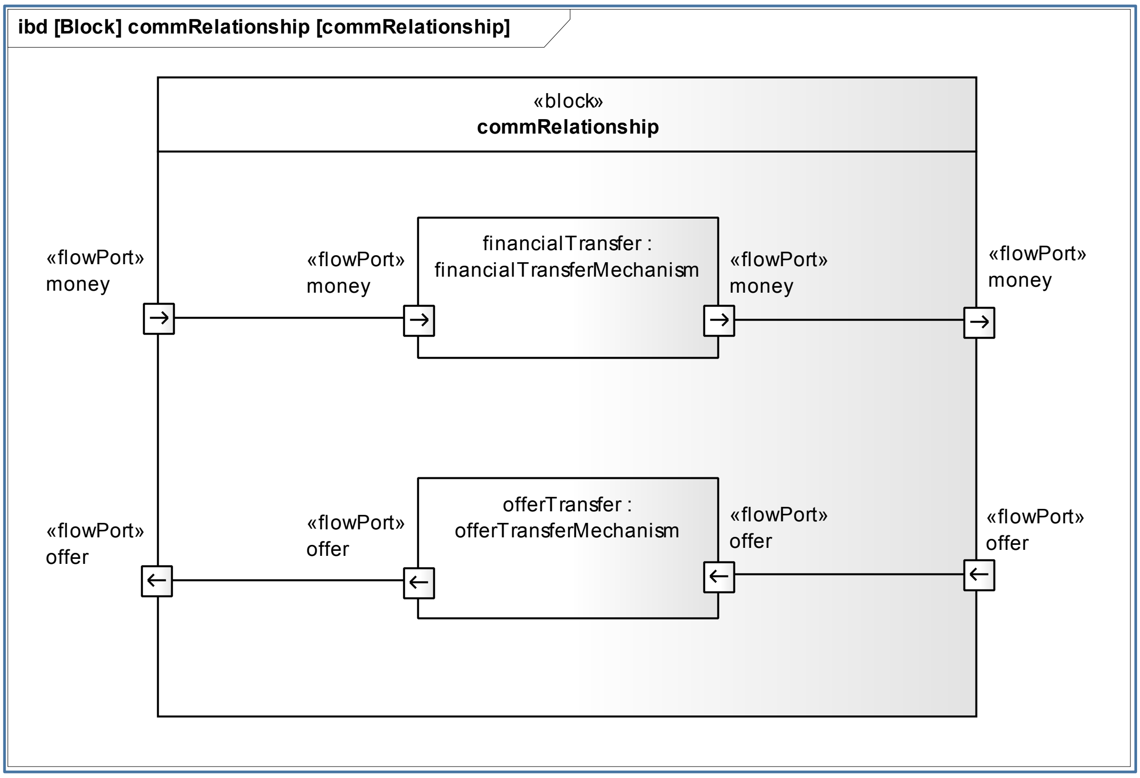

Figure 44: internal block diagram for a commercial relationship.

The model developed so far allows the capture of the structure and behaviour of relationships but it does not provide any insight into how constraints on the relationship might be modelled. SysML does provide the capability to model constraints using a *constraint block* however these are intended to specify constraints in terms of mathematical formulae or constants. For example, in the commercial relationship a constraint might be specified that states the value of the money item cannot be < 0 as this would specify a refund rather than a payment. This is too restrictive for the kind of constraints that are likely to be encountered in this application. A better solution is the use of free text attached to the relevant parts of the diagram. This precludes the possibility of generating an executable model but as that is not the objective here, that is not a concern.

As an example, consider the DD-FD system and the commercial relationship between the End User and the Aggregator. In this relationship the End User makes offers of flexibility to the Aggregator. In DD-FD such offers must be for a fixed period of 15, 30 or 60 minutes. This constraint is captured on the SysML diagram as a constraint note attached to the offer transfer mechanism as shown in Figure 45. This technique can be used to add any number of constraints to the model.

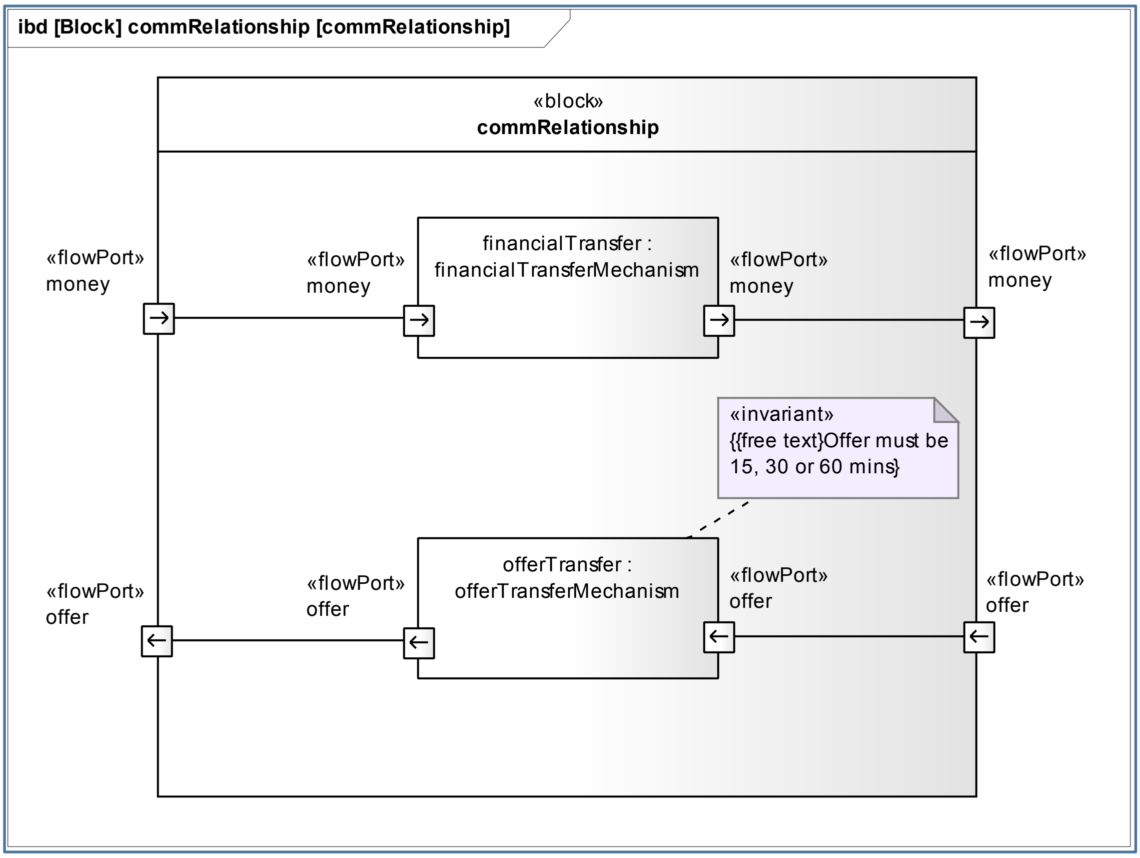

Figure 45: adding constraints as constraint notes to a SysML diagram.

Using the techniques outlined above it is possible to continue to expand the model by adding other relationships. Where constraints can be parameterised they should be added using constraint blocks, where they cannot constraint notes could be used. In this way a model of the system of relationships can be built and then changes added to reflect the addition of FD. By considering the constraints it should be possible to see where barriers occur.

7.6 Summary and Conclusions

The research presented in this chapter provides an initial outline of a framework within which the hypothesis that there exist structural barriers to the implementation of FD services can be tested. This hypothesis assumes that the implementation of FD services presents a challenge to the way the market operates today. Barriers are categorised as being technical, legal/regulatory or commercial; and are identified as being emergent properties of relationships between stakeholders.

To examine these barriers it is necessary to develop a framework for modelling relationships between stakeholders in such a way that the changes required to support FD services can be applied and any emergent barriers analysed. The foundations of such a framework have been developed based on the application of Systems Engineering techniques and utilising the Systems Modelling Language (SysML). The application of SysML to enterprise relationship modelling represents a novel application of the language.

The DD-FD project was used as a case study to allow the development of the methodology within a bounded and tractable problem. Using SysML block definition diagrams, internal block diagrams and use case diagrams it was possible to show how both the structure and behaviour of relationships within the DD-FD project may be modelled. Crucially, constraints are added to model the limits on relationships. Constraint notes can be used for those constraints that it is not possible to represent mathematically. Constraints are key indicators of the likely presence of barriers.

Chapter 8

Conclusions

8.1 Research Objective

The starting point for this research was the question:

> What is preventing a policy commitment to, and investment in, the large scale adoption of demand flexibility as a core component of system balancing within the future GB electrical energy system?

The hypothesis was that this is likely due to a perception that its adoption is for some reason too risky. The objective was to examine whether this is so and, if it is, to consider some ways by which this can be addressed. The research was to be conducted using Systems Engineering tools or techniques where possible.

8.2 Research Conclusions

Electricity currently accounts for only 18% of overall energy needs within the GB. However, climate change policy means that this is expected to grow as increasing amounts of transportation and heating migrate to electrical power. That power is most likely to be derived from a generation mix in which intermittent renewables have a significant role (DECC, 2010). This presents a challenge to system operation and offers an opportunity for Flexible Demand to provide a significant proportion of balancing services at relatively low cost. There is a compelling case for these Flexible

Demand services as they offer a complementary and cost-effective balancing facility to that of peaking plant and storage.

This thesis introduces the concept of the trading of flexibility as a new value chain within the energy market, one in which consumers can actively trade their willingness to be flexible with consumption – whilst retaining overall control – in return for financial compensation from other industry stakeholders. This new value chain facilitates the participation of consumers as active energy managers rather than passive consumers. The retention of control by the consumer, central to the concept of Flexible Demand, is a key mechanism for mitigating consumer reluctance and apathy.

A historical analysis of the industry shows that there have been two substantial structural changes in the post-war period: nationalisation in 1947 and privatisation in 1989. Both were ideologically driven. Considering the development of the industry since privatisation in 1989 reveals that the industry structure of Generators, Operators (Transmission, Distribution and System) and Suppliers of electricity, trading energy via bilateral agreements and wholesale markets, is a direct result of legislation and regulation imposed by Government. Furthermore the development of wind power provides a case study in the power of politically based incentives. Wind generation was restricted to a niche experimental area for more than two decades until climate mitigation policy created incentives that has resulted in explosive growth in both installed capacity and the size of turbines. This analysis leads to the following conclusions:

- **Conclusion 1:** There are precedents for ideologically driven large-scale structural change within the industry.
- **Conclusion 2:** There is a tight coupling between Government macroeconomic policy and the structure of the GB electricity industry.

- **Conclusion 3:** The commercial attractiveness of the large-scale implementation of new technology is massively increased with appropriate political backing.

Flexible Demand challenges the "predict and provide" paradigm. The participation of consumers as active energy managers does not fit well with this paradigm as it introduces bidirectional flows of information and money into an industrial structure that is essentially unidirectional. It is, therefore, likely to require structural change to support it. The conclusions stated above mean if Flexible Demand is to be deployed at a scale where it can provide a significant component of balancing services then it will need political backing. However, large-scale change can be difficult for a traditionally conservative industry. It can lead to a perception that the risk involved is too great. This also applies in the political arena where a fear of adverse consumer reaction can add to the perceived risk.

The most effective method for overcoming this perception is to demonstrate the viability of the proposed change. By demonstrating that any risk associated with it can be effectively mitigated then the perceptual barriers that are preventing the creation of the necessary incentives can be removed. These insights are summarised in the following conclusions:

- **Conclusion 4:** A perception of risk can act as a barrier to the creation of necessary incentives.

- **Conclusion 5:** A demonstration of the viability of the proposed change can remove the perceptual barrier.

Thus the answer to the initial question is that a perception of risk leads to a lack of political backing and of the incentives necessary to initiate the required structural changes that will permit the deployment of Flexible Demand on a large scale. By demonstrating its viability such perceptual barriers can be overcome.

Once the risk has been demonstrated to be acceptable there still remains the question of appropriate incentives. These incentives need to result in changes to the structure of relationships between stakeholders that allow for the implementation of Flexible Demand. By analysing these relationships within a structured modelling framework appropriate incentives can be designed.

- **Conclusion 6:** A framework for the analysis of stakeholder relationships will facilitate the design of appropriate incentives.

A system to demonstrate the technical viability of Flexible Demand was developed as part of the DD-FD project. This thesis is concerned with the development of one of the key components of that system: an algorithm for the optimal dispatch of loads. Using Systems Engineering techniques a set of requirements were derived. An extensive review of literature concerning combinatorial optimisation was undertaken. Together with an analysis of the problem it was identified as very likely to be *NP*-hard and therefore the required processing resources will scale exponentially with problem size. A review of scheduling approaches and search algorithms identified that a Genetic Algorithm is the most appropriate for the problem considered here.

- **Conclusion 7:** The processing requirements of an optimised dispatch algorithm in Flexible Demand are likely to scale exponentially with the number of offers.

- **Conclusion 8:** A Genetic Algorithm is the most appropriate type of search algorithm for the optimised dispatch of loads in this demonstrator.

A fitness function is presented that takes into account the difference in aggregate power between the solution and the target, a reliability index, and an economic cost index for each combination. This fitness function was implemented within a Genetic Algorithm and tested. The algorithm presented in this thesis was shown to successfully converge on a solution close to, or to be, the optimum within a reasonable processing budget for test sets up to 500 offers. Furthermore, it was shown to successfully find the global optimum in a test set that was sufficiently small to be able to identify the global optimum using an exhaustive search. This demonstrates that both the algorithm and the fitness function operate as intended.

- **Conclusion 9:** The Genetic Algorithm presented here is shown to converge on an optimal solution within a reasonable processing budget.

- **Conclusion 10:** The Genetic Algorithm presented here is capable of identifying with excellent repeatability either the global optimum or a solution very close to it.

Even when a demonstration successfully shows the viability of Flexible Demand there remains the need to address any barriers to implementation that might exist. A review of alternative modelling approaches identified no clear candidate for the modelling of both the structure and behaviour of relationships for the purpose described here. The use of the Systems Engineering language SysML was identified as a possible solution although applying it to this type of problem represents a novel application.

- **Conclusion 11:** The novel use of SysML for relationship modelling offers a framework for the analysis of implementation barriers.

An initial outline of a framework for modelling relationships has been developed using the DD-FD project as a case study. The use of constraints to illuminate barriers to the implementation of new services is suggested.

8.3 Claims of Originality

The application of systems thinking is essential in establishing the research context:

- There is a close coupling between macroeconomic policy and the structure of the GB electricity supply industry with the result that policy and not technology is the principal inhibitor to change.
- The perceived risk arising from the operational challenge presented by the use of Flexible Demand is most likely the principal reason for the lack of appropriate incentives.
- There is a need for effective demonstration to overcome risk perception.

Within this context this research provides the following novel contributions:

- Identification of the need for a framework for analysing stakeholder relationships to facilitate the design of incentives.
- Design and demonstration of a Genetic Algorithm for the optimised dispatch of large numbers of loads to produce a desired aggregate effect.
- An initial outline for the design and application of a Systems Engineering framework based on SysML for the analysis of implementation barriers within enterprises.

As a result of the success of this research the concepts and algorithms presented in this thesis are in the process of being incorporated into operational systems currently under development within E.ON New Build and Technology Limited.

8.4 Suggestions for Further Work

8.4.1 Optimised Dispatch

Incorporating More Complex Offers

The offers processed by the scheduling algorithm are relatively simple in that the duration is fixed and the response is assumed to be rectangular, i.e. a fixed power change for a fixed period (see Figure 17). It is likely that in an operational system there will be nodes that are able to offer more complex responses: perhaps multi-stage responses with defined ramps for the transitions such as that shown in Figure 46. It is also possible that nodes can be instructed to provide a response different to that offered, but within the bounds of the offer. For example, a node might offer a reduction of 10 kW for 30 minutes but able to provide a reduction of say 8 kW for 22 minutes. The mechanism by which variable offers are communicated is out of scope for the scheduling algorithm, but if such variable offers are to be utilised then the scheduling algorithm must be capable of processing them. This will likely require substantial modification to the structure of the chromosome and fitness function.

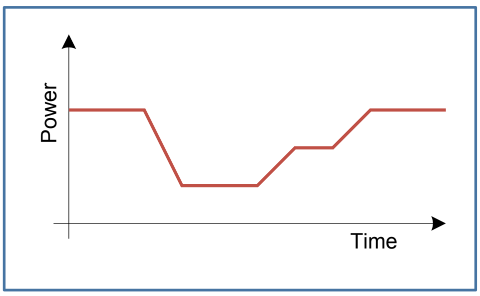

Figure 46: example complex consumption curve.

Alternative Fitness Functions

The fitness function includes the three elements that are relevant to an assessment of the adequacy of a solution to the scheduling problem: power, cost and reliability. The calculation of the reliability and economic cost indices may benefit from further study, as might the assessment of the values to be assigned to the scaling factors used within the fitness function.

Multiple Mutations

Larger offer pools mean longer chromosomes and with only a single mutation occurring in any one chromosome in any particular generation this means that the effect of the mutation is likely to reduce as the size of the offer pool increases. With larger offer pools it may be more effective to mutate more than one chromosome value during each mutation. This might help to increase diversity and reduce the number of generations required to get to an acceptable solution.

Offer Pool Pre-processor

Performance improvements can be achieved through constraining the size of the offer pool. In addition to excluding offers that will expire as is described in Chapter 5, a pre-processor might, for example, remove all offers in excess of the target power or preferentially select larger offers first as to do so is likely to produce a more economic and reliable solution than a combination of large numbers of small offers.

8.4.2 Structural Modelling using SysML

Further Testing

A basic methodology for modelling stakeholder relationships has been developed and tested on the simple test case of the DD-FD system. However, this approach needs to be tested on a more complex system to verify its general applicability and scalability. To do this it would be useful to first fully model all the relevant aspects of the DD-FD system and then expand the model by adding relationships between the DD-FD system and the external context.

Develop a Library

Many relationships are likely to have common characteristics and a library of models that can be imported and customised would be a useful tool. This concept of design and re-use is central to MBSE and SysML lends itself readily to this task.

Constraints

The approach to modelling constraints that is presented here requires the use of free text within constraint nodes. A more formal approach using an appropriate language such as the Object Constraint Language (OCL) (see www.omg.org/spec/OCL) or a programming language such as Java might provide a level of semantic rigour that assists in modelling and analysis.

Bibliography

1. AALAMI, H., YOUSEFI, G. R. & MOGHADAM, M. P. 2008. A MADM-based Support System for DR Programs. *In:* 2008 Proceedings of the 43rd International Universities Power Engineering Conference, Vol. 1-3.

2. ALBADI, M. H. & EL-SAADANY, E. F. 2008. A summary of demand response in electricity markets. *Electric Power Systems Research,* 78.

3. AMATO, A., DI LECCE, V. & PIURI, V. 2007. A smart distributed measurement data management system for DSM. *In:* 2007 IEEE Instrumentation and Measurement Technology, IMTC 2007 - Synergy of Science and Technology in Instrumentation and Measurement, 1-3 May 2007, Warsaw, Poland: Institute of Electrical and Electronics Engineers.

4. BABU, P. R. & DIVYA, V. P. S. 2008. Application of ANN and DSM techniques for peak load management a case study. *In:* 2008 IEEE International Conference on Sustainable Energy Technologies, ICSET 2008, 24-27 November 2008, Singapore: Institute of Electrical and Electronics Engineers Computer Society.

5. BBC NEWS. 2012. *Hundreds of millions without power in India* [Online]. BBC. Available: http://www.bbc.co.uk/news/world-asia-india-19060279 [Accessed 5 Sep 2012].

6. BELHOMME, R., SEBASTIAN, M., DIOP, A., ENTEM, M., BOUFFARD, F., VALTORTA, G., SIMONE, A. D., CERERO, R., YUEN, C., KARKKAINEN, S. & FRITZ, W. 2009. ADDRESS Technical and Commercial Conceptual Architectures - Core Document. *In:* BELHOMME, R. & BOUFFARD, F. (eds.). European Community FP7.

7. BOSHELL, F. & VELOZA, O. P. 2008. Review of Developed Demand Side Management Programs Including Different Concepts and their Results. *2008 IEEE/PES TRANSMISSION AND DISTRIBUTION CONFERENCE AND EXPOSITION: LATIN AMERICA, VOLS 1 AND 2; Transmission and Distribution Conference and Exposition-Latin America.* Institute of Electrical and Electronics Engineers.

8. BOYKOFF, M. T. & BOYKOFF, J. M. 2007. Climate change and journalistic norms: A case-study of US mass-media coverage. *Geoforum,* 38**,** 1190-1204.

9. BRAITHWAIT, S. D. 2003. Demand Response Is Important - But Let's Not Oversell (or Over-Price) It. *The Electricity Journal,* 16**,** 52-64.

10. BSI 2005. BS EN ISO 7730:2005 - Ergonomics of the thermal environment. Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria. UK: BSI.

11. BURGE, S. 2008. ELP061 Systems Approach, Analysis and Concepts. (A lecture series given at Loughborough University in October 2008.)

12. BUSTARD, D. W., HE, Z. & WILKIE, F. G. 2000. Linking soft systems and use-case modelling through scenarios. *Interacting with Computers,* 13**,** 97-110.

13. BUSTARD, D. W., HE, Z., WILKIE, F. G. & SPRAGUE, R. H., JR. 1999. Soft systems and use-case modelling: mutually supportive or mutually exclusive? *In:* SPRAGUE, R. H., JR., ed. Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32: Institute of Electrical and Electronics Engineers Computer Society.

14. CALABRESE, M., DI LECCE, V. & PIURI, V. 2007. ANN residential load classifier for intelligent DSM system. *2007 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, CIMSA*, 27-29 June 2007. Ostuni, Italy: Institute of Electrical and Electronics Engineers Computer Society.

15. CASCIO, V., BERNASCONI, S., SAUBA, G., MENDIGUTXIA, J. & KUNG, A. 2006. Integration of digital appliances in Demand Side Management systems. *In:* 14th Mediterranean Conference on Control and Automation, MED'06, 28-30 June 2006. Ancona, Italy: Institute of Electrical and Electronics Engineers Computer Society, Mediterranean Control Association.

16. CHARDON, A., ALMÉN, O., LEWIS, P. E., STROMBACK, J. & CHÂTEAU, B. 2008. Demand Response: a decisive breakthrough for Europe. Paris, France: Capgemini.

17. CHECKLAND, P. B. 1998. *Systems Thinking, Systems Practice,* UK, John Wiley & Sons Ltd.

18. CHILDS, D. 2006. *Britain since 1945: a political history*, Routledge.

19. COMMISSION OF THE EUROPEAN COMMUNITIES 2008. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions — 20-20 by 2020 — Europe’s Climate Change Opportunity. Brussels, Belgium: Commission of the European Communities.

20. CORNWALL, N. 2007. Cash-out review 2007: "An independent perspective". London: Cornwall Energy Associates.

21. CRUICKSHANK, R. F. Year. Orchestrating utility supply and demand in real-time via the internet, home networks smart appliances. *In:* 10th IASTED International Conference on Power and Energy Systems, PES 2008, 16-18 April 2008. Baltimore, MD, USA: Acta Press.

22. DECC. 2009. Digest of United Kingdom energy statistics (DUKES): 60th Anniversary Edition. London: Department of Energy and Climate Change.

23. DECC. 2010. 2050 Pathways Analysis. London, UK.

24. DECC. 2011. *SMART METERING IMPLEMENTATION PROGRAMME: PROSPECTUS* [Online]. DECC. Available: http://www.decc.gov.uk/en/content/cms/consultations/smart_mtr_imp/smart_mtr_imp.aspx [Accessed 5 Sep 2012].

25. DECC. 2013. 2013ed. Digest of United Kingdom energy statistics (DUKES). London: Department of Energy and Climate Change.

26. DEINDL, M., BLOCK, C., VAHIDOV, R. & NEUMANN, D. 2008. Load shifting agents for automated demand side management in micro energy grids. *In:* 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2008, 20-24 October 2008. Venice, Italy: Institute of Electrical and Electronics Engineers Computer Society.

27. DIRKER, H. G., PRETORIUS, L. & PRETORIUS, J. H. C. 2008. *A Systems Thinking Approach to Manage Distribution Channels in the Control and Instrumentation Product Market with Multi-Faceted Product Lines.* New York: Institute of Electrical and Electronics Engineers.

28. DIWEKAR, U. M. 2008. *Introduction to Applied Optimization*, Springer.

29. ELEXON. 2012. The Electricity Trading Arrangements: A Beginner's Guide. ELEXON. 17 December 2012. Available: http://www.elexon.co.uk/wp-content/uploads/2012/12/electricity_trading_arrangements_beginners_guide_v3.0.pdf

30. ERIKSSON, H. E. & PENKER, M. 2000. *Business Modelling with UML*, New York, Wiley.

31. FARHANGI, H. 2010. The path of the smart grid. *Power and Energy Magazine*, 8, 18-28.

32. FAWKES, S. 1987. Soft-systems model of energy management and checklists for energy managers. *Applied Energy*, 27, 229-241.

33. FERRARI, F. M., FARES, C. B. & MARTINELLI, D. P. 2002. The systemic approach of SSM: The case of a Brazilian company. *Systemic Practice and Action Research*, 15, 51-66.

34. FIELDEN, D. & JACQUES, J. K. 1998. Systemic approach to energy rationalisation in island communities. *International Journal of Energy Research*, 22, 107-129.

35. FIRST HYDRO COMPANY. 2012. Electric Mountain: Nature At Its Most Powerful [Online]. Available: http://www.fhc.co.uk/downloads/pdf/EM Teachers Pack Pre Visit English v4.pdf [Accessed 5 Sep 2012].

36. FRIEDENTHAL, S., MOORE, A. & STEINER, R. 2008. *A Practical Guide to SysML*, Morgan Kaufmann.

37. FROST & SULLIVAN, 2012a. Electric Energy Storage Systems to Enhance Electric Grid Reliability and Support Expanded Renewable Energy Usage, States Frost & Sullivan. Available: http://www.prnewswire.com/news-

releases/electric-energy-storage-systems-to-enhance-electric-grid-reliability-and-support-expanded-renewable-energy-usage-states-frost--sullivan-147722215.html.

38. GAS TO POWER JOURNAL, 2012c. RWE npower to start 2,000-MW Pembroke CCGT in September. *Gas to Power Journal* [Online]. Available: http://gastopowerjournal.com/index.php/projectsafinance/item/761-rwe-npower-to-start-2000-mw-pembroke-ccgt-in-september [Accessed 5 Sep 2012].

39. HITCHINS, D. K. 2003. *Advanced systems thinking, engineering, and management,* Norwood, MA: Artech House.

40. HITCHINS, D. K. 2010. *All Things "Systems"* [Online]. Derek Hitchins. Available: http://www.hitchins.net/Systems.html [Accessed 05 Oct 2010].

41. HUI, L., CHONGQING, K. & QING, X. 2004. Price based decision making for demand side management considering customer satisfaction index. *Power System Technology,* 28**,** 1.

42. IEA 2012. [Online]. Available: http://www.ieadsm.org/Content.aspx?ID=3#_ftn1 [Accessed 5 Sep 2012].

43. IPCC 2007. Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. *In:* CORE WRITING TEAM, PACHAURI, R. K. & REISINGER, A. (eds.). Geneva, Switzerland.

44. JACKSON, M. C. 2001. Critical systems thinking and practice. *European Journal of Operational Research,* 128**,** 233-244.

45. JACOBSON, I., ERICSSON, M. & JACOBSON, A. 1994. *The Object Advantage: Business Process Reengineering with Object Technology,* Reading, MA, Addison-Wesley.

46. JUNJI, Q. W., WANG, L. & YUDONG TANG, Y. Z. 2003. Determination and analysis of TOU (time-of-use) power price based on DSM (demand side management) and MCP (marketing clearing price). *In:* 6th International Conference on Advances in Power System Control, Operation and Management. Proceedings. APSCOM 2003. IEE.

47. KHAN, T. 2009. The Electricity Trading Arrangements: A beginner's guide.

48. KRUCHTEN, P. 2000. *The Rational Unified Process: an Introduction,* Reading, MA, Addison-Wesley.

49. KUNDU, S., MAHATO, M., MAHANTY, B. & ACHARYYA, S. 2008. Comparative Performance of Simulated Annealing and Genetic Algorithm in Solving Nurse Scheduling Problem. *International MultiConference of Engineers and Computer Scientists.* Hong Kong.

50. KUPZOG, F. 2006. Self-controlled exploitation of energy cost saving potentials by implementing distributed demand side management. *In:* 2006 IEEE International Conference on Industrial Informatics, INDIN'06, 16-18 August 2006. Singapore: Institute of Electrical and Electronics Engineers Computer Society.

51. LANE, D. C. & OLIVA, R. 1998. The greater whole: towards a synthesis of system dynamics and soft systems methodology. *European Journal of Operational Research,* 107, 214-235.

52. LITTLECHILD, S. 2007. Electricity Cash Out Arrangements. London.

53. LITTLECHILD, S. 2010. The Creation of a Market for Retail Electricity Supply. Cambridge University.

54. MAIJ, E., TOUSSAINT, P. J., KALSHOVEN, M., POERSCHKE, M. & ZWETSLOOT-SCHONK, J. H. M. 2002. Use cases and DEMO: aligning functional features of ICT-infrastructure to business processes. *International Journal of Medical Informatics,* 65**,** 179-191.

55. MANIKAS, T. W. & CAIN, J. T. 1996. Genetic Algorithms vs. Simulated Annealing: A Comparison of Approaches for Solving the Circuit Partitioning Problem. Pittsburgh, USA: University of Pittsburgh.

56. MICHIELS, W., AARTS, E. & KORST, J. 2007. *Theoretical Aspects of Local Search,* Eindhoven, Springer.

57. MOURA, P. S. & DE ALMEIDA, A. T. 2010. Multi-objective optimization of a mixed renewable system with demand-side management. *Renewable and Sustainable Energy Reviews,* 14**,** 1461-1468.

58. NATIONAL GRID 2006. Procurement Guidelines.

59. NATIONAL GRID 2009a. Balancing Principles Statement.

60. NATIONAL GRID 2009b. Operating the Electricity Transmission Networks in 2020: Initial consultation. London: National Grid.

61. NATIONAL GRID 2010. National Electricity Transmission System Performance Report 2009 - 2010.

62. NATIONAL GRID. 2011a. *60 minute frequency trace* [Online]. Available: http://www.nationalgrid.com/uk/Electricity/Data/Realtime/Frequency/Freq60.htm [Accessed 29 Mar 2011].

63. NATIONAL GRID. 2011b. *About the gas network* [Online]. Available: http://www.nationalgrid.com/uk/Gas/About/How+Gas+is+Delivered/ [Accessed 14 Mar 2011].

64. NEVES, L. M. P., DIAS, L. C., ANTUNES, C. H. & MARTINS, A. G. 2009. Structuring an MCDA model using SSM: a case study in energy efficiency. *European Journal of Operational Research,* 199**,** 834-845.

65. NEVES, L. M. P., MARTINS, A. G., ANTUNES, C. H. & DIAS, L. C. 2004. Using SSM to rethink the analysis of energy efficiency initiatives. *Journal of the Operational Research Society,* 55**,** 968-975.

66. OBJECT MANAGEMENT GROUP. 2010a. *SysML* [Online]. Available: http://www.omgsysml.org/ [Accessed 27 Sep 2010].

67. OBJECT MANAGEMENT GROUP. 2010b. *UML* [Online]. Available: http://www.uml.org/ [Accessed 27 Sep 2010].

68. ODEH, M. & KAMM, R. 2003. Bridging the gap between business models and system models. *Information and Software Technology,* 45, 1053-1060.

69. OFGEM. 2009. *Markets* [Online]. London: Ofgem. Available: http://www.ofgem.gov.uk/Markets/Pages/Markets.aspx [Accessed 10 Mar 2009].

70. OFGEM 2010. Project Discovery: Options for delivering secure and sustainable energy supplies. London.

71. OSBORNE, J. & WARRIER, D. 2007. A Primer on Demand Response. USA: Thomas Weisel Partners.

72. OULD, M. A. 2004. Riva: A rigorous approach for Business Process Management. Venice Consulting Ltd.

73. OULD, M. A. 2005. *Business Process Management: A Rigorous Approach*, British Computer Society.

74. PETKOV, D., PETKOVA, O., ANDREW, T. & NEPAL, T. 2007. Mixing multiple criteria decision making with soft systems thinking techniques for decision support in complex situations. *Decision Support Systems,* 43**,** 1615-1629.

75. PRACA, I., MORAIS, H., RAMOS, C., VALE, Z. & KHODR, H. 2008. Multi-agent electricity market simulation with dynamic strategies Virtual Power Producers. *In:* IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES, 20-24 July 2008. Pittsburgh, PA, USA: Institute of Electrical and Electronics Engineers Computer Society.

76. RAMANATHAN, B. & VITTAL, V. 2008. A Framework for Evaluation of Advanced Direct Load Control With Minimum Disruption. *IEEE Transactions on Power Systems,* 23**,** 1681.

77. RUIZ, N., COBELO, I. & OYARZABAL, J. 2009. A Direct Load Control Model for Virtual Power Plant Management. *IEEE Transactions on Power Systems,* 24**,** 959.

78. SI ALHIR, S. 2003. *Learning UML,* Sebastopol, USA, O'Reilly & Associates, Inc.

79. STADLER, M., KRAUSE, W., SONNENSCHEIN, M. & VOGEL, U. 2009. Modelling and evaluation of control schemes for enhancing load shift of electricity demand for cooling devices. *Environmental Modelling & Software,* 24**,** 285.

80. STERMAN, J. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*, McGraw-Hill Higher Education.

81. STERN, N. 2007. *The Economics of Climate Change: The Stern Review*, Cambridge University Press.

82. STRBAC, G. 2008. Demand side management: Benefits and challenges. *Energy Policy*, 36**,** 4419-4426.

83. TANG, Y., SONG, H., HU, F. & ZOU, Y. Year. Investigation on TOU pricing principles. *In:* 2005 IEEE/PES Transmission and Distribution Conference and Exhibition - Asia and Pacific, 15-18 August 2005. Dalian, China: Institute of Electrical and Electronics Engineers.

84. THE ECONOMIST, 2012b. Packing some power. *The Economist* [Online], 2012. Available: http://www.economist.com/node/21548495?frsc=dg|a.

85. TOVEY, N. K. 2004. Recent Changes in the Electricity Markets in the UK. *In:* Proceedings of 3rd International Conference on the Operational Experience and Practice of the European Electricity Markets, May 2004. Moscow, Russia.

86. TOVEY, N. K. Year. Developments in the Electricity Markets in the UK: the move towards BETTA. *In:* 4th International Conference on the Operational Experience and Practice of the European Electricity Markets, May 2005. Moscow, Russia.

87. UK GOVERNMENT 1947. Electricity Act 1947. London, United Kingdom.

88. UK GOVERNMENT 1957. Electricity Act 1957. London, United Kingdom.

89. UK GOVERNMENT 1989. Electricity Act 1989. London, United Kingdom.

90. UK GOVERNMENT 2000. Utilities Act 2000. London, United Kingdom.

91. UK GOVERNMENT 2004. Energy Act 2004. London, United Kingdom.

92. UK GOVERNMENT 2008a. Climate Change Act 2008. London, United Kingdom.

93. UK GOVERNMENT 2008b. Energy Act 2008. London, United Kingdom.

94. UK GOVERNMENT 2009a. Carbon Budgets Order 2009. London, United Kingdom.

95. UK GOVERNMENT 2009b. Climate Change Act 2008 (2020 Target, Credit Limit and Definitions) Order 2009. London, United Kingdom.

96. VOGEL, U. & SONNENSCHEIN, M. 2007. Optimization of Adaptive Consumers to a Time-varying Electricity Supply. *Information Technologies in Environmental Engineering.* Heidelberg, Germany: Springer Berlin.

97. WIERING, M. J., BONSANGUE, M. M., VAN BUUREN, R., GROENEWEGEN, L. P. J., JONKERS, H. & LANKHORST, M. M. 2004. Investigating the mapping of an Enterprise Description Language into UML 2.0. *Electronic Notes in Theoretical Computer Science,* 101, 155-179.

98. WOO, C. K., LLOYD, D. & TISHLER, A. 2003. Electricity market reform failures: UK, Norway, Alberta and California. *Energy Policy,* 31**,** 1103-1115.

99. YU, N. & YU, J.-L. 2006. Optimal TOU decision considering demand response model. *In:* 2006 International Conference on Power System Technology, POWERCON2006, 22-26 October 2006. Chongqing, China: Institute of Electrical and Electronics Engineers.

100. ZACHMAN, J. A. 2008. *The Zachman Framework™: The Official Concise Definition* [Online]. Zachman International. Available:

http://www.zachmaninternational.us/index.php/the-zachman-framework [Accessed 4 Oct 2010].

101. ZHANG, J., CHUNG, H. S. H., LO, W. L., HUI, S. Y. R. & WU, A. 2001. Decoupled Optimization of Power Electronics Circuits Using Genetic Algorithms. *In:* CHAMBERS, L. (ed.) *The Practical Handbook of Genetic Algorithms Applications.* 2nd ed.: Chapman & Hall/CRC.

102. ZHANG, W. & FELIACHI, A. 2003. Residential load control through real-time pricing signals. *In:* Proceedings of the 35th Southeastern Symposium on System Theory, 2003.

Appendix A: Published Papers

The following paper was published during the course of this programme of research:

HODGSON, G., THOMSON, M. & CLIFFORD, C. 2011. A systems engineering approach to resolving structural barriers to the implementation of demand response. *In:* Proceedings of the 8th International Conference on the European Energy Market (EEM 2011), 25-27 May 2011. Zagreb, Croatia: Institute of Electrical and Electronics Engineers.

Appendix B: Example Source Code

The application described in Chapter 5 uses object-oriented programming techniques. Classes are used to implement the objects of an offer, a pool of offers, a chromosome and a population of chromosomes. The relevant class definitions is described below.

Offer

The `Offer` class holds the information that constitutes an offer from a load.

```
public class Offer
{
```

Enumerated types are used to define the types of offer and the offer status. Use of enumerations helps clarity elsewhere in the code.

```
    // public variables and constants
    public enum OfferType { Invalid = -1, Reduction = 0, Increase = 1 };
    public enum OfferStatus { Invalid = -1, Available = 0, InUse = 1 };
```

All values are held in private variables accessed via `get` and `set` methods.

```
    // property accessors
    public string OfferId { get; set; }         // the offer's identifier
    public OfferType Type { get; set; }         // reduction or increase
    public double Value { get; set; }           // the power value
    public int Period { get; set; }             // the amount of time that
the offer is for (e.g. 15 minutes)
    public DateTime Expiry { get; set; }        // expiry time of the
offer
    public OfferStatus FdStatus { get; set; }   // whether the offer is
available for use or has already been used in another event
    public double AbsCost { get; set; }         // absolute cost of the
offer
    public double CostIndex { get; set; }       // relative cost of the
offer based on the contents of the pool
    public double Reliability { get; set; }     // relative reliability
(0...1 where 0 is most reliable)
```

A single default constructor creates an empty offer with all internal values initialised to the default values. The default date is arbitrarily chosen to be 1 January 2000.

```
    // constructors
    public Offer()
    {
        OfferId = null;
        Type = OfferType.Invalid;
        Value = 0;
        Period = 0;
        Expiry = new DateTime(2000, 01, 01, 0, 0, 0, DateTimeKind.Utc);
        FdStatus = OfferStatus.Invalid;
        AbsCost = double.MaxValue;
        CostIndex = 1;
        Reliability = 1;
    }
}
```

Pool of Offers

The `OfferPool` class is used to hold a pool of offers.

```
public class OfferPool
{
```

Each offer is contained in an instance of the `Offer` class, which are held in a .NET Dictionary indexed by the unique offer identifier.

```
    // local variables
    private Dictionary<string, Offer> pool;      // used to hold the offers
    private int poolSize;                        // holds the number of
offers in the pool
    private int period;
    DateTime minExpiry;

    public int PoolSize { get { return poolSize; } }
```

A single constructor is provided which builds the pool from a specified test data set. The required period (15, 30 or 60) must also be supplied together with a value for the minimum allowable time difference between an offer's expiry time and the current time. This allows for latency in processing and executing events. Note that for testing purposes the time is fixed at 1 June 2012 11:55:00 UTC as this aligns with the test data that were created for the purpose of testing the algorithm.

```
    // constructor
    public OfferPool(int _period, int _expiryDelta, string _dataset)
    {
        period = _period;

        // ** work out the minimum expiry time **
        //minExpiry = DateTime.UtcNow;
// uncomment for operational use
        minExpiry = new DateTime(2012, 6, 1, 11, 55, 00,
DateTimeKind.Utc);    // test timestamp aligned with test data
        minExpiry = minExpiry.AddMinutes(_expiryDelta);

        // ** create a new list of offers with a default capacity **
        pool = new Dictionary<string, Offer>();

        // ** fetch the offers from the database **
        switch (_dataset)
        {
            case "TD1":
                buildTd1();
                break;
            case "TD2_50":
                build50();
                break;
            case "TD2_100":
                build100();
                break;
            case "TD2_250":
                build250();
                break;
            case "TD2_500":
                build500();
                break;
            default:
                buildOther();
                break;
        }
        poolSize = pool.Count;
```

Once the pool has been constructed the relative cost of each offer is calculated.

```
        // ** calculate the cost indices **
        // determine the most and least expensive offers
        if (poolSize > 0)
        {
            double mostExpensive = 0;
            double leastExpensive = double.MaxValue;
            foreach (string id in pool.Keys)
            {
                if (pool[id].AbsCost < leastExpensive)
                {
                    leastExpensive = pool[id].AbsCost;
                }
                if (pool[id].AbsCost > mostExpensive)
                {
                    mostExpensive = pool[id].AbsCost;
                }
            }
            // assign cost index to each offer
            double costDelta = mostExpensive - leastExpensive;
            // if most expensive and least expensive are equal then delta
is zero so catch this
            if (costDelta == 0) { costDelta = 1; }
            foreach (string id in pool.Keys)
            {
                pool[id].CostIndex = (pool[id].AbsCost - leastExpensive) /
costDelta;
            }
        }
    }
```

A number of methods allow access to parameters of the individual offers within the pool.

```
    // fetch the power value stored in the specified offer
    public double OfferPoolOfferPwr(string _offerId)
    {
        return (double)pool[_offerId].Value;
    }

    // fetch the reliability index stored in the specified offer
    public double OfferPoolOfferRel(string _offerId)
    {
        return pool[_offerId].Reliability;
    }

    // fetch the cost index stored in the specified offer
    public double OfferPoolOfferCost(string _offerId)
    {
```

```
        return pool[_offerId].CostIndex;
    }
```

A method is provided to return a list of the identifiers of all offers within the pool.

```
    // return list of offer ids
    public List<string> OfferIdList()
    {
        List<string> returnedIdList = new List<string>();
        foreach (string id in pool.Keys)
        {
            returnedIdList.Add(id);
        }
        return returnedIdList;
    }
```

The following routine builds a pool from the TD1 dataset. Similar routines are used for the other datasets but are omitted for clarity.

```
    // build the offer pool for the TD1 dataset
    private void buildTd1()
    {
        // initialise the interface to the database
        OfferData.TD1DataTable table = new OfferData.TD1DataTable();
        OfferDataTableAdapters.TD1TableAdapter tableAdapter = new
OfferDataTableAdapters.TD1TableAdapter();
        // fetch the list of offers from the database
        tableAdapter.Fill(table);

        // ** build the pool **
        foreach (OfferData.TD1Row row in table)
        {
            Offer newOffer = new Offer();
            newOffer.OfferId = row.OfferId;
            newOffer.Type = Offer.OfferType.Reduction;      // only
reduction offers currently handled
            newOffer.Value = row.Offer;
            newOffer.Period = row.Period;
            newOffer.Expiry = row.ExpiryTime;
            newOffer.AbsCost = row.Cost;
            newOffer.Reliability = row.Reliability;

            if (row.FdStatus == 0) { newOffer.FdStatus =
Offer.OfferStatus.Available; }     // 0 indicates offer not in use by
another event
            // add if the offer is of the right duration, will not expire
before trigger received and is not currently in use
            if (newOffer.Period == period && newOffer.Expiry > minExpiry
&& newOffer.FdStatus == Offer.OfferStatus.Available)
            {
```

```
                pool.Add(newOffer.OfferId, newOffer);
            }
        }
    }
}
```

Chromosome

The GaChromosome class is used to hold a single chromosome.

```
public class GaChromosome
{
    // local constants
    private const double a = 1;     // scalar for normalised power
    private const double b = 1;     // scalar for reliability index
    private const double c = 1;     // scalar for cost index

    // local variables
    private Guid identifier;                        // unique identifier
for the chromosome
```

The chromosome is held in a Dictionary indexed with the offer identifiers. A Boolean value indicates whether the chromosome actively uses the offer or not. A true value means that it does.

```
    private Dictionary<string, bool> chromosome;    // dictionary to hold
chromosome
    private double fitness, relFitness, cumFitness; // fitness parameters
```

A string is used to hold a shorthand representation of the chromosome as a string of T (for true) and F (for false) characters. This is used by some of the chromosome processing routines as a quicker mechanism for manipulating the chromosome.

```
    private string chromosomeString;                // string holding the
chromosome signature
    private Random rnd;                             // random number
generator, unique to each chromosome
    private OfferPool pool;                         // pointer to the
offer pool
    private double tgtPwr;                          // the target power
(used in fitness calculation)
    private double aggPwr;                          // the aggregate power
of this solution

    // property accessors
    public Guid ChromosomeId { get { return identifier; } }
```

```
    public double Fitness { get { return fitness; } }
    public double RelFitness { get { return relFitness; } set { relFitness
= value; } }
    public double CumFitness { get { return cumFitness; } set { cumFitness
= value; } }
    public string ChromosomeString { get { return chromosomeString; } }
    public double AggPwr { get { return aggPwr; } }
    public double A { get { return a; } }
    public double B { get { return b; } }
    public double C { get { return c; } }
```

A constructor is provided to build a new chromosome and randomly choose which offers are used.

```
    // ** constructor **
    // This constructor creates a new chromosome and initialises it with
random values
    public GaChromosome(OfferPool _pool, double _tgtPwr)
    {
        // store initialisation parameters
        pool = _pool;
        tgtPwr = _tgtPwr;

        // assign a new identifier to this chromosome
        identifier = Guid.NewGuid();

        // initialize dictionary to hold offers
        chromosome = new Dictionary<string, bool>(_pool.PoolSize);

        // create a random number generator
        rnd = new Random();

        // add offers and randomly assign values
        foreach (string s in _pool.OfferIdList())
        {
            if (rnd.NextDouble() < 0.5)
            {
                chromosome.Add(s, false);
            }
            else
            {
                chromosome.Add(s, true);
            }
        }

        // create chromosome string
        this.calcString();

        // calculate fitness
        this.calcFitness();
```

```
    }
```

The following method returns the value (true or false) within the chromosome for a specified offer.

```
    // query value of specified offer in chromosome
    public bool GaChromosomeQueryOfferStatus(string _offerId)
    {
        return chromosome[_offerId];
    }
```

This method allows the contents of the chromosome to be overwritten with those of another chromosome. The resulting chromosome is a clone of the specified chromosome. This method is used for manipulation of chromosomes within a population.

```
    // replace the contents of this chromosome with those of the given
chromosome
    public void GaChromosomeReplace(GaChromosome _c)
    {
        // copy the ID
        identifier = _c.ChromosomeId;

        // copy chromosome contents
        Dictionary<string, bool> newChromosome = new Dictionary<string,
bool>(chromosome.Count);
        foreach (string s in chromosome.Keys)
        {
            newChromosome.Add(s, _c.GaChromosomeQueryOfferStatus(s));
        }
        chromosome = newChromosome;

        // copy chromosome string
        chromosomeString = _c.ChromosomeString;

        // copy values
        aggPwr = _c.AggPwr;
        fitness = _c.Fitness;
        relFitness = _c.RelFitness;
        cumFitness = _c.CumFitness;
    }
```

This method is another manipulation routine and forces a chromosome to adopt a new unique identifier.

```
    // force a new ID
```

```
    public void GaChromosomeChangeId()
    {
        identifier = Guid.NewGuid();
    }
```

The following method is used to mutate the chromosome by randomly flipping one of its values.

```
    // mutate chromosome
    public void GaChromosomeMutate()
    {
        // randomly select a mutation point between start and end
        int x = rnd.Next(chromosome.Count - 1);
        // select relevant offer ID
        string s = chromosome.ElementAt(x).Key;
        // flip offer status at mutation point
        if (chromosome[s]) { chromosome[s] = false; } else { chromosome[s]
= true; }

        // update chromosome string
        calcString();

        // update fitness value
        calcFitness();
    }
```

The next method implements the crossover function between this chromosome and another that is specified as an input parameter. The position at which the splice is to take place is also specified.

```
    // crossover chromosome
    // _c is the chromosome that is being crossed with this one
    // position is the zero-based crossover point, everything from this
position onwards is crossed
    public void GaChromosomeXover(GaChromosome _c, int position)
    {
        // everything from position onwards will be crossed
        for (int i = position; i < chromosome.Count; i++)
        {
            string offerid = chromosome.ElementAt(i).Key;
            chromosome[offerid] =
_c.GaChromosomeQueryOfferStatus(offerid);
        }

        // update chromosome string
        calcString();

        // update fitness value
```

```
        calcFitness();
    }
```

The following method is used to calculate the fitness of the chromosome by implementing the fitness function described in Chapter 5.

```
    // calculate fitness value
    private void calcFitness()
    {
        //initialise local variables
        double rel = 0;                     // sum of reliability indices
        double cost = 0;                    // sum of cost indices
        int t = pool.PoolSize;              // number of offers within the
pool
        int n = pool.PoolSize;              // number of unique nodes within
the pool

        // reset aggregate power
        aggPwr = 0;

        // get the list of IDs within the chromosome
        Dictionary<string, bool>.KeyCollection idList = chromosome.Keys;

        // iterate through offers to build summations
        foreach (string k in idList)
        {
            // if the offer is used in this chromosome then add it to the
summations
            if (chromosome[k])
            {
                // update summations
                aggPwr += pool.OfferPoolOfferPwr(k);
                rel += pool.OfferPoolOfferRel(k);
                cost += pool.OfferPoolOfferCost(k);
            }
        }

        // calculate value
        fitness = (a * Math.Abs(aggPwr - tgtPwr) / tgtPwr) + (b * rel / n)
+ (c * cost / t);
    }
```

The final method builds the shorthand string representation of the chromosome.

```
    // calculate chromosome string
    private void calcString()
    {
        // clear any existing content
        chromosomeString = "";
        foreach (string s in chromosome.Keys)
```

```
        {
            if (chromosome[s]) { chromosomeString += "T"; } else {
chromosomeString += "F"; }
        }
    }
}
```

Population of Chromosomes

The Population class is used to hold a complete population of chromosomes. Each chromosome within the population is represented by an instance of the GaChromosome class described above.

```
public class Population
{
    // local variables
    private int g;                                          // generation
counter
    private Dictionary<Guid, GaChromosome> chromPop;        // population of
chromosomes
```

The reference chromosome, i.e. the fittest chromosome in the population, is held in the refChromosome property.

```
    private GaChromosome refChromosome;                     // the reference
chromosome
    private Guid[] orderedList;                             // used to hold a
list of chromosome IDs ordered by fitness from best [0] to worst [n]

    // property accessors
    public int Gen { get { return g; } }
    public GaChromosome RefChromosome { get { return refChromosome; } }
```

There are two constructors: one to build the first generation and one that builds subsequent generations. This first constructor builds a generation randomly populated with chromosomes that are all unique.

```
    // ** constructors **
    // This method builds the first generation
    public Population(OfferPool _pool, double _tgtPwr, int _popSize)
       {
        // if the offer pool is empty then nothing to be done
        if (_pool.PoolSize == 0) { return; }

        // initialise a random number generator
```

```
        Random rnd = new Random();

        // ** Step 2: initialise parameters **
        // parameters are set as constants above
        // ** Step 3: initialise the chromosomal population of the GA **
        // set generation counter to 1
        g = 1;
        // create a dictionary to hold the chromosome population
        chromPop = new Dictionary<Guid, GaChromosome>();
        // add chromosomes
        while (chromPop.Count < _popSize)
        {
            bool unique = true;        // flag used when checking if the
chromosome already exists in the population
            // create a new chromosome
            GaChromosome newChromosome = new GaChromosome(_pool, _tgtPwr);
            // check that this chromosome is unique and add it if it is
            foreach (Guid key in chromPop.Keys)
            {
                if (newChromosome.ChromosomeString ==
chromPop[key].ChromosomeString)
                {
                    unique = false;
                    break;
                }
            }
            if (unique && newChromosome.Fitness >= 0)
            {
                // chromosome is unique with +ve fitness so add it to the
population
                chromPop.Add(newChromosome.ChromosomeId, newChromosome);
            }
        }

        // ** Step 4: Determine reference chromosome **
        // order the population by fitness
        orderedList = new Guid[_popSize];
        orderPop();

        // calculate the relative fitness values
        calcRelFitness();

        // calculate the cumulative fitness values
        calcCumFitness();

        // reference chromosome is at the head of the ordered list
        refChromosome = chromPop[orderedList[0]];
       }
```

This second constructor builds a new generation based on the previous one by applying mutation and crossover. It does not guarantee that all chromosomes within the population are unique.

```
    // This method creates a new generation by applying mutation and
crossover
    // It is based on the constructor above
    public Population(OfferPool _pool, double _tgtPwr, int _popSize,
Population _prevPop, double _px, double _pm, double _smax, double _smin)
    {
        // if the offer pool is empty then nothing to be done
        if (_pool.PoolSize == 0) { return; }

        // initialise a random number generator
        Random rnd = new Random();

        // increment generation number
        g = _prevPop.Gen + 1;

        // save a copy of the reference chromosome from the previous
generation
        GaChromosome oldRefChromosome = new GaChromosome(_pool, _tgtPwr);
        oldRefChromosome.GaChromosomeReplace(_prevPop.RefChromosome);

        // ** Step 5: Reproduction **
        // select chromosomes that will propagate into this generation
        // generate random selector variable between smin and smax
        // Any chromosome with a cumulative fitness (0...1) less than or
equal to this value will be propagated.
        // If smin is too low then the chances for a low value of p are
greatly increased which will lead to very
        // few being selected for propagation and a consequent drop in
diversity. The population will rapidly fill
        // with multiples copies of the same few chromosomes. smax can be
used to actively prevent propagation of
        // the worst chromosomes. smin and smax therefore control the
diversity.
        // If smin is too low the algorithm will rely on crossover and
mutation for diversity which are both
        // probabilistic events.
        // Need to make sure that smin is not < the cumulative fitness
value of the current best
        // otherwise nothing will be selected, setting it to the previous
best fitness guarantees a value
        // greater than or equal to it is used as the selector.
        if (_smin < oldRefChromosome.CumFitness)
        {
            _smin = oldRefChromosome.CumFitness;
        }
        // calculate a random increment between smax and smin that is
```

```
added to smin
        double p = rnd.NextDouble();
        double increment = (_smax - _smin) * p;
        p = _smin + increment;

        // copy from the existing population all chromosomes with a
cumulative fitness <= p
        chromPop = new Dictionary<Guid, GaChromosome>();
        foreach (Guid k in _prevPop.chromPop.Keys)
        {
            if (_prevPop.chromPop[k].CumFitness <= p && chromPop.Count <
_popSize)
            {
                chromPop.Add(k, _prevPop.chromPop[k]);
            }
        }

        // fill any gaps in the population with duplicates
        while (chromPop.Count < _popSize)
        {
            foreach (Guid k in _prevPop.chromPop.Keys)
            {
                if (_prevPop.chromPop[k].CumFitness <= p && chromPop.Count
< _popSize)
                {
                    GaChromosome newChromosome = new GaChromosome(_pool,
_tgtPwr);

newChromosome.GaChromosomeReplace(_prevPop.chromPop[k]);
                    newChromosome.GaChromosomeChangeId();
                    chromPop.Add(newChromosome.ChromosomeId,
newChromosome);
                }
            }
        }

        // iterate through chromosomes randomly selecting them for
crossover
        List<Guid> crossoverList = new List<Guid>();
        foreach (Guid k in chromPop.Keys)
        {
            p = rnd.NextDouble();
            if (p <= _px)
            {
                crossoverList.Add(k);
            }
        }

        // perform crossovers
        for (int i = 0; i < crossoverList.Count; i = i + 2)
        {
            // trap an odd number of entries in the list
```

```
            if (i + 1 == crossoverList.Count) { break; }

            // select a zero-based crossover position (everything from x
onwards)
            int x = rnd.Next(_pool.PoolSize - 2) + 1;

            // make a copy of the first
            GaChromosome c1 = new GaChromosome(_pool, _tgtPwr);
            c1.GaChromosomeReplace(chromPop[crossoverList[i]]);
            // perform the crossover with the second
            c1.GaChromosomeXover(chromPop[crossoverList[i + 1]], x);

            // make a copy of the second
            GaChromosome c2 = new GaChromosome(_pool, _tgtPwr);
            c2.GaChromosomeReplace(chromPop[crossoverList[i + 1]]);
            // perform the crossover with the first
            c2.GaChromosomeXover(chromPop[crossoverList[i]], x);

            // only replace the chromsomes with the modified versions if
fitness >= 0
            if (c1.Fitness >= 0)
            {
                chromPop[crossoverList[i]].GaChromosomeReplace(c1);
            }
            if (c2.Fitness >= 0)
            {
                chromPop[crossoverList[i + 1]].GaChromosomeReplace(c2);
            }
        }

        // iterate through chromosomes performing mutations
        foreach (Guid k in chromPop.Keys)
        {
            p = rnd.NextDouble();
            if (p <= _pm)         // mutate if true
            {
                // make a copy of the chromosome
                GaChromosome c1 = new GaChromosome(_pool, _tgtPwr);
                c1.GaChromosomeReplace(chromPop[k]);
                // perform the mutation
                c1.GaChromosomeMutate();
                // only replace the chromsome with the modified version if
fitness >= 0
                if (c1.Fitness >= 0)
                {
                    chromPop[k].GaChromosomeReplace(c1);
                }
            }
        }

        // order the population by fitness
        orderedList = new Guid[_popSize];
```

```
        orderPop();

        // calculate the relative fitness values
        calcRelFitness();

        // calculate the cumulative fitness values
        calcCumFitness();

        // reference chromosome is at the head of the ordered list
        refChromosome = chromPop[orderedList[0]];

        // check to see if the previous generation's reference is better
than this one
        // replace this generation's reference with the previous
generation's if it is
        if (oldRefChromosome.Fitness < refChromosome.Fitness)
        {
            refChromosome.GaChromosomeReplace(oldRefChromosome);
        }
    }
```

The following private method is used to re-order by fitness the chromosomes within the population.

```
    // Method to re-order the population by increasing fitness value
    private void orderPop()
    {
        // build a list of IDs and fitness values
        Dictionary<Guid, double> unsorted = new Dictionary<Guid,
double>(chromPop.Count);
        foreach (Guid k in chromPop.Keys)
        {
            unsorted.Add(k, chromPop[k].Fitness);
        }

        // build the sorted list
        double minVal;
        Guid g;
        int index = 0;
        while (unsorted.Count > 0)
        {
            // assume that the first element is the best
            minVal = unsorted.First().Value;
            g = unsorted.First().Key;
            // find the lowest value
            foreach (Guid k in unsorted.Keys)
            {
                if (unsorted[k] < minVal)
                {
                    minVal = unsorted[k];
```

```
                    g = k;
                }
            }
            // add the selected values to the sorted list
            orderedList[index] = g;
            index++;
            // remove the just added ID from the unsorted list
            unsorted.Remove(g);
        }
    }
```

The following two methods are used to calculate the relative and cumulative fitness values.

```
    // Method to calculate relative fitness values
    private void calcRelFitness()
    {
        double fitnessTotal = 0;

        // calculate fitness total
        foreach (Guid key in chromPop.Keys)
        {
            fitnessTotal += chromPop[key].Fitness;
        }

        // calculate relative fitness
        for (int i = 0; i < chromPop.Keys.Count; i++)
        {
            chromPop.ElementAt(i).Value.RelFitness =
chromPop.ElementAt(i).Value.Fitness / fitnessTotal;
        }
    }

    // Method to calculate cumulative fitness values
    private void calcCumFitness()
    {
        double relFitnessTotal = 0;

        // iterate through the chromosomes in fitness order
        for (int i = 0; i < orderedList.Length; i++)
        {
            relFitnessTotal += chromPop[orderedList[i]].RelFitness;
            chromPop[orderedList[i]].CumFitness = relFitnessTotal;
        }
    }
```

This final method is implementation specific and writes the new population to the SQL database.

```
    // Method to write this population to database
    // Writes generation number, reference chromosome ID, target power and
the entire chromosome population to the database
    public void WritePopToDb(double _tgtPwr, string _table)
    {
        // record the time now
        DateTime now = DateTime.UtcNow;

        // write to the correct database table
        switch (_table)
        {
            case "RuntimeData":
                // initialise the interface to the database
                OfferDataTableAdapters.RuntimeDataTableAdapter
RuntimeDataTableAdapter = new
OfferDataTableAdapters.RuntimeDataTableAdapter();
                // add each chromosome to the db in order from best to
worst
                GaChromosome current;
                for (int i = 0; i < orderedList.Length; i++)
                {
                    current = chromPop[orderedList[i]];
                    RuntimeDataTableAdapter.Insert(now, g, _tgtPwr,
refChromosome.ChromosomeId.ToString(), current.ChromosomeId.ToString(),
current.AggPwr, current.Fitness, current.RelFitness, current.CumFitness,
current.A, current.B, current.C, current.ChromosomeString);
                }
                break;
            default:
                break;
        }
    }
}
```