

Electricity Trading Between Smart Nano-grids

Luke McGuinness

A thesis submitted to the University of Dublin, Trinity College

in fulfillment of the requirements for the degree of

Master of Computer Science

University of Dublin, Trinity College

Supervisor: Dr. Donal O'Mahony

Submitted to the University of Dublin, Trinity College

May 2014

Declaration

I, Luke McGuinness, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Luke McGuinness

Dated: May 22, 2014

Summary

Currently, the electricity grid is centrally controlled and has poor demand-response capabilities. This dissertation proposes the design of nano-grid systems that aim to change electricity grid intelligence to be more distributed in nature, as well as laying the groundwork for improving overall grid efficiency. By adding nano-grids, utility companies would have the ability to interact with users of the grid with greater demand-response balancing potential. Nano-grids are essentially autonomous entities at the bottom of the grid hierarchy, i.e. a heater and a battery that are both connected to some sort of programmable controller that has access to electricity would be considered a nano-grid.

This dissertation implements nano-grid systems and a game theoretic bargaining algorithm that seeks to reduce the peak-to-average power ratio of the electricity grid through balancing planned usage. Utility companies interact with the nano-grids to plan the best time for them to buy electricity.

A small nano-grid system was successfully established, showing the implementation of the proposed nano-grid software design hierarchy. The algorithm used is outlined and examined to evaluate its usefulness when used by different numbers of simulated users. It is seen to operate successfully but requires more advanced testing.

This research area is relatively new and much research still needs to be carried out.

Acknowledgements

I would like to thank my supervisor Dr. Donal O'Mahony, for his assistance and guidance throughout this dissertation.

I would like to thank my friends, in particular Kieran O'Brien and Tom Curran, who kept me going through the toughest times.

Finally, I would like to thank my family without whom I could never have gotten this far, with special thanks to my brother John for support throughout my dissertation.

Luke McGuinness

University of Dublin, Trinity College

May 2014

Contents

Summary	iii
Acknowledgements	iv
List of Tables	ix
List of Figures	x
Chapter 1 Introduction	1
1.1 Background and Context	1
1.2 Scope and Objectives	1
1.3 Methodologies	2
1.4 Overview of Dissertation	2
Chapter 2 State of the Art	3
2.1 The Centralised Electricity Grid	3
2.1.1 Traditional Grid Paradigm	3
2.1.2 Adoption of Renewable Energy	3
2.2 Nano-Grids	5
2.2.1 Existing Nano-Grid research	5
2.2.2 A Smarter Grid	9
2.2.3 Grid Infrastructure Development	10

2.2.4	Supply-Demand Balancing Capabilities	10
2.3	Prosumers	11
2.3.1	Hawaii	11
2.3.2	The Rest of the USA	12
2.3.3	Germany	13
2.4	The Solar Panel Market	14
2.5	Security Concerns	15
2.5.1	Security	15
2.5.2	Potential uses for exposed data	15
2.5.3	Intruders	16
2.5.4	Privacy	16
2.6	Game Theory	19
2.6.1	Nash Equilibrium	20
2.6.2	Pareto Optimal Solution	20
2.7	Trading Algorithms	20
Chapter 3 Methodology		22
3.1	Architecture	22
3.1.1	Hardware	22
3.1.2	Software	24
3.2	Testing Framework	25
3.3	Limitations of Methodology	25
3.3.1	Information	25
3.3.2	Hardware	26
3.3.3	Simulation	26
Chapter 4 Technical Work		27
4.1	Design	27
4.1.1	Planned Trading Algorithm	27

4.1.2	Base Design	33
4.1.3	Nano-grid Logic Design	34
4.2	Implementation	36
4.2.1	Networking	36
4.2.2	Abstract Classes	38
4.2.3	User Interface	39
4.2.4	Light Interface	40
4.2.5	Battery Interface	40
4.2.6	Feature Breakdown	41
4.2.7	Optimising Planned Usage	41
4.2.8	Finding the Cheapest Source	43
Chapter 5 Results		44
5.1	Balancing 10,000 Users	45
5.1.1	Load Profile at Iteration 0	45
5.1.2	Load Profile at Iteration 1	45
5.1.3	Load Profile at Iteration 2	46
5.1.4	Peak-to-Average Ratio over Iterations	46
5.2	Balancing 50,000 Users	47
5.2.1	Load Profile at Iteration 0	47
5.2.2	Load Profile at Iteration 1	47
5.2.3	Load Profile at Iteration 2	48
5.2.4	Load Profile at Iteration 3	48
5.2.5	Peak-to-Average Ratio over Iterations	49
5.3	Balancing 100,000 Users	50
5.3.1	Load Profile at Iteration 0	50
5.3.2	Load Profile at Iteration 1	50
5.3.3	Load Profile at Iteration 2	51

5.3.4	Load Profile at Iteration 3	51
5.3.5	Peak-to-Average Ratio over Iterations	52
5.4	Balancing 200,000 Users	53
5.4.1	Load Profile at Iteration 0	53
5.4.2	Load Profile at Iteration 1	53
5.4.3	Load Profile at Iteration 2	54
5.4.4	Load Profile at Iteration 3	54
5.4.5	Peak-to-Average Ratio over Iterations	55
Chapter 6 Conclusion		56
6.1	Evaluation	56
6.1.1	Nano-grid Class Hierarchy	56
6.1.2	Planned Usage Algorithm	56
6.2	Future Work	57
6.2.1	Scaled Testing	57
6.2.2	Value Testing	57
6.2.3	Interesting Use Cases	57
6.2.4	Higher Level of Automation	58
Bibliography		59
Appendix A Code		64
Appendix B Code		67

List of Tables

4.1	Game Theory Term Reference	29
-----	--------------------------------------	----

List of Figures

2.1	LoCal System Architecture ©[2008] IEEE	5
2.2	Nano-Grid Conceptual Design ©[2012] IEEE [1]	6
2.3	German Electricity Production by Source for a Given Week [2]	13
3.1	Logical Representation of the Hardware Configuration Used	23
4.1	Planned Trading Algorithm Overview - (Transmitted Data Shown)	31
4.2	Inertia Scenario	32
4.3	Bounded Step Scenario	32
4.4	Class Diagram	35
4.5	Solar Nano-grid Graphical User Interface	39
4.6	Light Nano-grid Graphical User Interface (Idling)	40
4.7	Light Nano-grid Graphical User Interface (Running)	40
4.8	Battery Nano-grid Graphical User Interface	41
4.9	Planned Trading Algorithm with Example Transmission Data	42
5.1	10K Users balancing their loads equally among their given time-slots	45
5.2	10K Users after iteration 1 of the planning algorithm	45
5.3	10K Users after iteration 2 of the planning algorithm	46
5.4	The peak-to-average power ratio of total load throughout balancing	46
5.5	50K Users balancing their loads equally among their given time-slots	47

5.6	50K Users after iteration 1 of the planning algorithm	47
5.7	50K Users after iteration 2 of the planning algorithm	48
5.8	50K Users after iteration 3 of the planning algorithm	48
5.9	The peak-to-average power ratio of total load throughout balancing . . .	49
5.10	100K Users balancing their loads equally among their given time-slots . .	50
5.11	100K Users after iteration 1 of the planning algorithm	50
5.12	100K Users after iteration 2 of the planning algorithm	51
5.13	100K Users after iteration 3 of the planning algorithm	51
5.14	The peak-to-average power ratio of total load throughout balancing . . .	52
5.15	200K Users balancing their loads equally among their given time-slots . .	53
5.16	200K Users after iteration 1 of the planning algorithm	53
5.17	200K Users after iteration 2 of the planning algorithm	54
5.18	200K Users after iteration 3 of the planning algorithm	54
5.19	The peak-to-average power ratio of total load throughout balancing . . .	55

Chapter 1

Introduction

1.1 Background and Context

While electrical grids have been updated and expanded over the years, it has always been a case of meeting varying demand with large consistent centralised supplies that can be adjusted as needed. This has been possible due to energy supply being primarily provided by non-renewable but consistent resources like fossil fuels. The relatively recent adoption of renewable energy sources has changed this. There are now distributed supplies (solar, wind, etc.) joining the grid which can be erratic depending on different environmental conditions.

As the cost of renewable power generation continues to decrease and the adoption rate of these technologies increases, the grid will become more complex. As more medium and large facilities are built, and home and business markets emerge, the complexity of the grid will require its balancing to become more distributed in nature.

1.2 Scope and Objectives

The goal of this dissertation is to examine the implications of setting up a real world nano-grid trading system and observe its operations. The nano-grid system will attempt

to use trading strategies to match electrical supply at off-peak times with loads that can be scheduled, thus reducing energy costs.

1.3 Methodologies

The system being implemented for this research makes use of game theoretic trading strategies that aim to reduce costs for providers and customers. Using these strategies all users receive the best outcome by trading *selfishly* and honestly.

1.4 Overview of Dissertation

The dissertation consists of six chapters:

Chapter 2 Summarises background research undertaken for this dissertation relating to the electricity grid, smart grids and trading algorithms.

Chapter 3 Covers the experimental methods utilised and the reasoning behind those choices.

Chapter 4 Details in-depth the research carried out in this dissertation.

Chapter 5 Reports the results found via testing.

Chapter 6 Concludes with analysis and discussion of the results garnered.

Chapter 2

State of the Art

2.1 The Centralised Electricity Grid

2.1.1 Traditional Grid Paradigm

The electrical grid requires a large amount of management to handle generation, transmission, distribution and control of the networks to schedule electricity supply safely, reliably and economically. Grid management essentially requires regulation of the grid's power, voltage and frequency. This is made possible by the reliable nature of generation capabilities allowing for the ramping up of energy production at short notice. This task is simplified by the centralised nature of the grid.

2.1.2 Adoption of Renewable Energy

Currently renewable resources are being introduced at large scale to electrical grids and being used in conjunction with conventional means of supply. These integrate relatively easily because the grid still remains largely centralised. More rigorous planning to account for the crests and troughs of inconsistent large scale renewable supplies is required. Even more difficult to manage will be the growing number of households and businesses adding small scale renewable energy to the grid.

A relatively new source contributing to the electrical grid are “prosumers”. “Prosumers” are consumers of the grid’s electricity that now also produce electricity and can store and/or sell their produced electricity back to the grid. More information on this is provided in Chapter 2.3 on page 11.

2.2 Nano-Grids

Origin There is as of yet no formal definition of a nano-grid that is universally agreed upon. One of the first practical instances of a nano-grid, although it was presented under a different name, was the LoCal System introduced by M.M. He et al.(2008) [3]. The LoCal System considered nano-grids to consist of some sort of IPS (Intelligent Power Switch), one or more loads, some capacity to store electricity and optionally a supply (as seen in Fig. 2.1). Connected nano-grids can be represented as loads or supplies, this would be determined via communication.

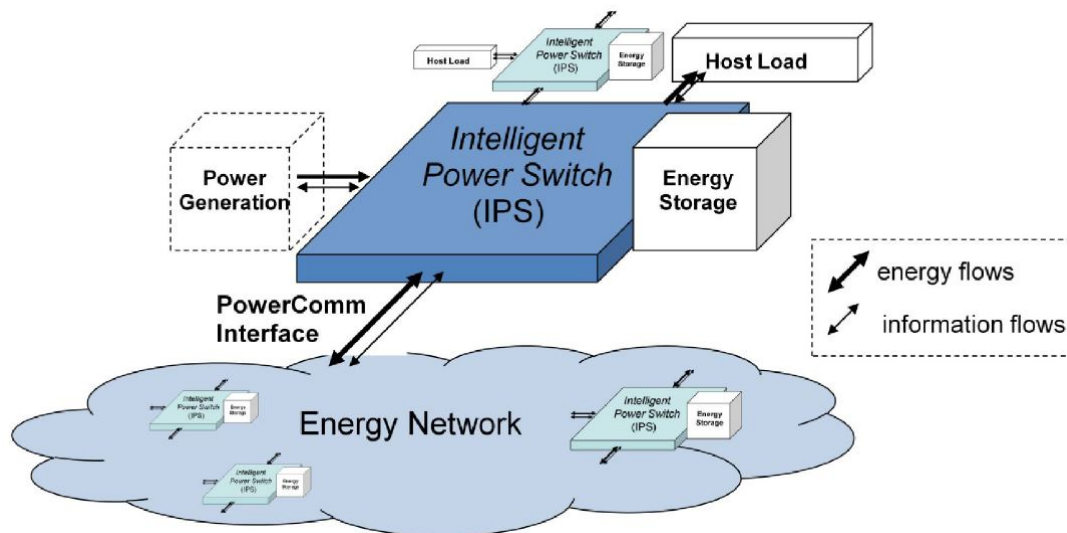


Fig. 2.1: LoCal System Architecture ©[2008] IEEE

2.2.1 Existing Nano-Grid research

Currently the leading concept of a nano-grid is outlined by Bruce Nordman (2011) [4], "A nanogrid is a single domain for voltage, reliability, and administration. It must have at least one load (sink of power, which could be storage) and at least one gateway to the outside" (as seen in Fig. 2.2).

In this design, the controller is a simple computer that contains all the logic of

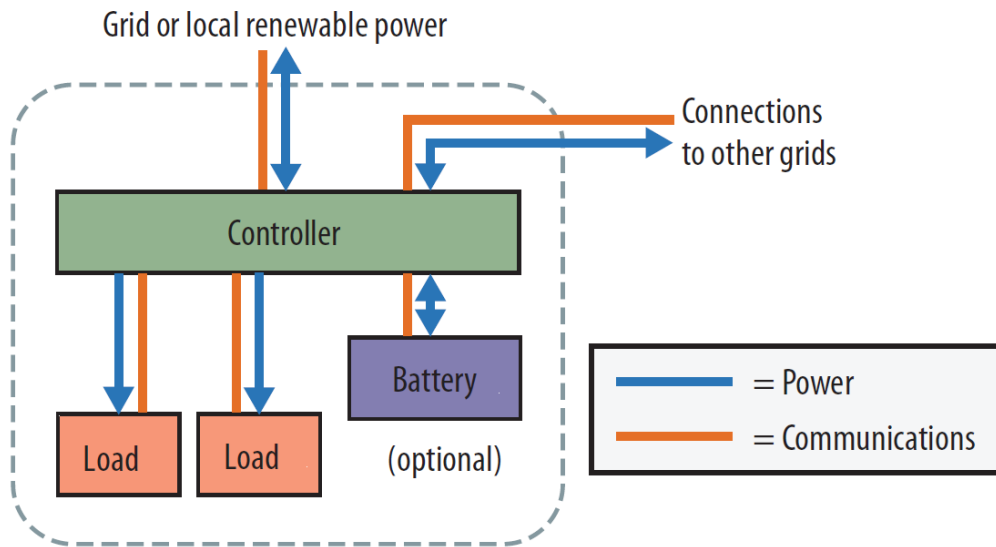


Fig. 2.2: Nano-Grid Conceptual Design ©[2012] IEEE [1]

the nano-grid. It gathers information on its attached loads, sources and storage by monitoring current and voltage of its inputs and outputs.

I. Cvetkovic et al. [5] raises several scenarios of unique potential for nano-grids. Scenarios that require high-reliability are highlighted as areas that would benefit considerably from nano-grid systems:

1. Emergency facilities like hospitals currently use backup generators and battery arrays as they strive to maintain power during grid disturbances. Nano-grids could increase the intelligence of such systems with potential performance gains.
2. Data centers already often contain large arrays of batteries and/ or server mounted batteries, used in conjunction with local backup generators. Standardisation of power management techniques would likely come as a relief to these facilities, as opposed to their current marketplace of incompatible proprietary solutions.
3. Military usage could see many applications, perhaps most interestingly as a robust mobile power management system.

Lower reliability-centric scenarios include electric vehicle power management. A car could use an embedded controller to become a nano-grid, its battery becoming a storage device that uses and is used by the grid intelligently. Allowing not only for easy management of the vehicle's battery but also intelligent interaction with the wider grid with potential for demand-response balancing mirroring that of nano-grid controlled stationary batteries (more information on this available in 'Chapter 2.2.4 Demand-Response Capabilities' on page 10).

In developing countries nano-grids offer great potential for setting up ad-hoc style grids that could be easily connected to any traditional infrastructure that was subsequently added to the area. At an even more basic level, 12V car batteries are often used to power several small loads in areas without mains power, a single nano-grid could more intelligently manage such usage to provide a safer system.

DC (Direct Current) in the Context of Nano-grids Traditionally electricity has been AC (Alternating Current) throughout the grid's distribution network and in homes and businesses. This is the case because AC allows electricity to be transformed to higher voltages to decrease the losses incurred in moving electricity over long distances. Progress in high voltage DC transmission has leveled the playing field with regards to this advantage of AC.

There is now a new push towards DC microgrids for the following reasons:

1. The majority of home and office devices now run using DC power but must use AC to DC converters (incurring losses).
2. Solar energy is produced in DC and must be changed to AC for the grid's sake. It would be better for prosumers if this energy could be used with as few conversions as possible.
3. It would be easier to develop products without worrying about international frequency standards of AC varying.

The EMerge Alliance are an organisation that is attempting to tackle legal requirements and regulations in this area and are eager to see local DC systems operational and efficient. “The EMerge Alliance is an open, membership-based, not-for-profit industry association formed in 2008 to create and promote the adoption of new standards for direct current “DC” power distribution within commercial buildings to improve their flexibility and sustainability” [6].

Research Pertaining to the Physical Aspects of Nano-grids Modelling of power converters for a DC-based nano-grid system has been undertaken to better understand how nano-grids will interact through power conversion [7]. This research aims to allow acquisition of useful converter information without being given access to the controls, dynamics or internal structure of the converter. This research will be useful for nano-grids in particular in the case of a nano-grid first being attached to other devices; the concept proposed by this paper would allow nano-grids to identify the voltages they would be required to supply to/ request from other nano-grids without user input.

Work has also been done by R. Adda et al. [8] to produce novel ways of converting power for standalone DC nano-grid applications. This research shows how to effectively power local AC loads from DC sources (e.g. solar panels and storage). It also boasts the ability of supplying both DC and AC loads simultaneously from a single DC input. This research would be particularly applicable to mobile nano-grids that would be required to connect to other devices without knowing their physical specifications in advance.

A process for handling multiple supplies all attempting to charge a battery using a single controller has been implemented by S.U. Arun et al. [9]. It operates by accepting multiple inputs, it then varies the duty cycle based on optimum operation of the largest supply with the other supplies acting as slaves and boosting their voltage to match the largest supply and then charge a battery. This has potential to allow multiple nano-grids to all source power to a single large load.

Nano-grid like systems are required now more than ever as it becomes more and more economically feasible to supply and store energy locally. A house with solar panels can provide for itself, but in the case where it has more than enough electricity, that excess has the potential to be sold back to the grid. A neighbour who likely gets their electricity from non-renewable resources can now avail of the locally sourced renewable energy. This also circumvents the losses incurred when power is sourced from a distant power station.

There are several intentions behind the design of nano-grid systems:

- To make the grid smarter (i.e. easier and better reporting of usage, more intelligent decisions being made autonomously).
- Nano-grids should be able to cloak their complexity and appear simply as a load or supply to the external grid. This allows nano-grids to be added on the fly to an existing centralized grid, an existing decentralized grid (of nano-grids) or operate independently.
- Availability of storage and access to market signals gives demand-response balancing capabilities (i.e. Market signals can be observed to buy electricity when demand is low and then sell it when demand is high, resulting in profit for the nano-grid and reduced pressure on the utility provider).

2.2.2 A Smarter Grid

Nano-grids contain intelligence not currently seen in traditional grid design. They offer real-time reporting of grid activity in a similar way to smart meters (smart meters are the digital counterpart of analog meters, they can offer additional features such as near real-time pricing of electricity). They offer even more potential however as they can report statistics, such as current storage capacity and maximum capacity, and contain the potential to act on this information.

Nano-grids, once implemented, could be updated in the future to play a role in an as of yet unthought of solution to the grid's ongoing issues.

2.2.3 Grid Infrastructure Development

Electricity grids the world over are outdated to varying degrees. Systems contain parts that were installed as long as forty years ago. These parts are now at the end of their life-span and performing much less efficiently because of this.

A report on the need for a smarter grid in America sheds light on the issue of reliability: "There have been five massive blackouts over the past 40 years, three of which have occurred in the past nine years. More blackouts and brownouts are occurring due to the slow response times of mechanical switches, a lack of automated analytics, and "poor visibility" a "lack of situational awareness" on the part of grid operators. This issue of blackouts has far broader implications than simply waiting for the lights to come on. Imagine plant production stopped, perishable food spoiling, traffic lights dark, and credit card transactions rendered inoperable. Such are the effects of even a short regional blackout". [10]

While grids will require continual maintenance going forward, nano-grids can be used to supplement what is already in place. Nano-grids can be installed in conjunction with the current grid infrastructure without the need for any changes to it.

2.2.4 Supply-Demand Balancing Capabilities

Nano-grids allow management of distributed supplies, loads and storage. Through interaction with other suppliers and consumers of electricity there exists everything needed to balance supply and demand more accurately.

This idea is fleshed out in Chapter 4.1.1 on p.27.

2.3 Prosumers

Currently solar panels are the most common source of production of electricity for prosumers. In areas with very high adoption rates it is being found that the introduction of so much solar power is stressing the local electricity distribution systems. The original system design would not have foreseen this kind of behavior.

2.3.1 Hawaii

Hawaii is one such place that has had a solar energy boom. In September of 2013 a utility company in Hawaii, Hawaiian Electric Co. (HECO), began requiring contractors and residents to request permission before adding more solar panels to the grid. HECO says that the amount of solar energy in the system poses a threat to the system itself and raises issues of safety. HECO wants studies undertaken on the system to discover if upgrades are necessary and solar contributor taxation to fund this. The public in turn is skeptical of the company and suspects that they simply wish to retain consumers.

Issues that arise from too much solar energy entering the grid are caused by the grid's innate structure. The grid traditionally is built to disperse energy from large power stations to sub-stations. The sub-stations then power individual premises. When an area controlled by a sub-station begins to generate too much power, it is unable to redistribute this power elsewhere effectively. This creates a situation where there is "over-voltage" and potential for power surges. "Overvoltages are extremely high voltages that damage or even completely destroy insulation and hence impair or completely disrupt the function of electrical and electronic components of all kinds" [11]. Crews working in such areas will also face new challenges as power can come from non-traditional sources (e.g. even if the local sub-station is dormant, prosumers may still be generating power).

One way to circumvent many of these issues is to install batteries in conjunction with solar panels, and remove interaction with the main grid. This would require the system to be disconnected from the main grid and be entirely self-sufficient. This is usually not

cost effective with current battery technology as batteries would need to be replaced many times during the lifetime of the solar panels.

Hawaiian Electric is currently in the process of piloting schemes to handle the requirements of this evolving grid. The scope of these schemes is known to contain how “To reliably transform and increase renewable energy penetration and facilitate adoption of promising distributed resource technologies for the grid, an integrated approach to see and manage change impacts from the distribution up to the transmission system will be needed by all utilities and electric grid operators” [12].

2.3.2 The Rest of the USA

The struggle in Hawaii is seen as predictive for other states in the USA (e.g. California, Arizona and Colorado) as they begin to deal with the same issues. While only around 1% of the market at this point is solar, the issues can arise from over-saturation of a small area.

Net metering is the system whereby electricity meters already in place on a prosumer’s premises are run backwards when they supply energy to the grid. This leaves the prosumer paying for the amount electricity they bought from the grid minus the amount they supplied.

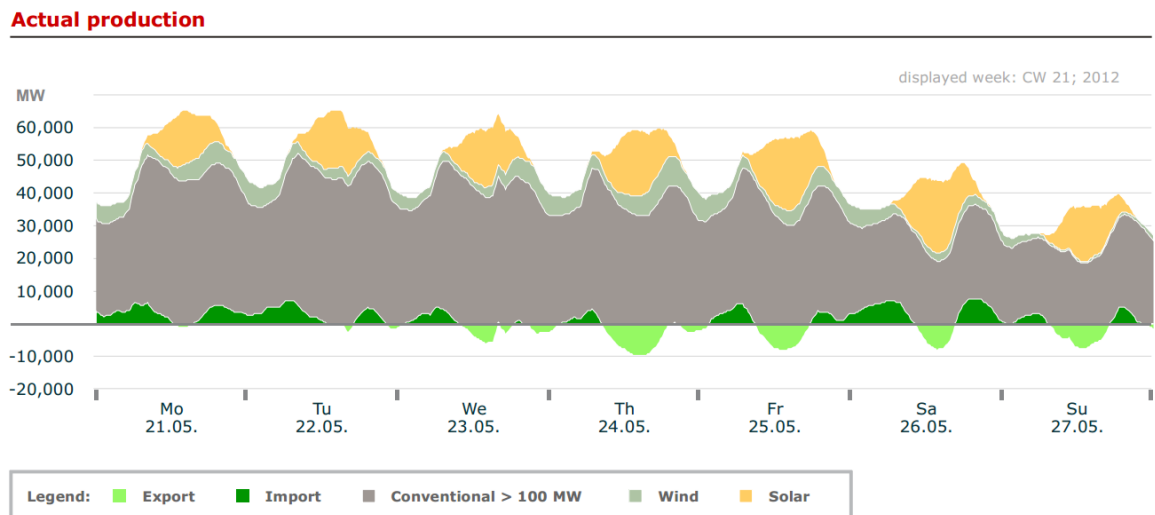
Net metering is a hot topic in America over fears that the people with the least money could end up paying for any necessary upgrades to the grid. When solar powered electricity is added to the grid by a prosumer, the prosumer pays less but the grid requires more funding. In California it was found that the average household income of prosumers was \$91,000, making it \$37,000 more than that of the average household’s income of \$54,000. Current net metering practices favour those who can afford installation of solar panels. New laws are already being passed to address this issue. Such laws are spearheading the initiative and will likely need to be revised as the situation continues to develop.

2.3.3 Germany

Germany has a massive collection of solar panels achieved through forward thinking legislation and the guarantee that renewable energy will be bought from green suppliers at above market prices.

This has now created a huge change to the norm in how mid-day usage of the grid occurs. Where once utility companies received large amounts of money for powering up stations to meet peak demand, now solar power is at its most productive around the same time. This can go as far as to result in traditional powering methods needing to be

Electricity Production in Germany: Calendar Week 21



Graph: Bruno Burger, Fraunhofer ISE; Data: EEX Transparency Platform

170

© Fraunhofer ISE

Fraunhofer
ISE

Fig. 2.3: German Electricity Production by Source for a Given Week [2]

dialled back (as seen in Fig. 2.3, with conventional energy reduction most clearly visible on the Friday).

This same power profile raises the issue of solar power's inconsistency. Large amounts of unexpected cloud cover creates huge supply/ demand imbalance as the expected power from solar sources becomes unavailable. Existing infrastructure has not been designed to supply for this unexpected demand at such short notice.

As the issues of maintaining and upgrading the grid continue the cost falls to the general consumers. These increased bills can end up pushing people towards buying solar panels, for the saving they could make if they were generating their own solar power. In this way, the utility companies are inadvertently exacerbating the issue.

2.4 The Solar Panel Market

Renewable energy has been a hot topic in recent years. There exists a desire to lessen a country's dependency on fossil fuels for both economic as well as environmental reasons. This has led to a market that is heavily influenced by global governments.

This has led to Chinese manufacturers flooding the solar panels market. A solar panel boom in China has been created by government tax reductions in foreign countries (e.g. Germany, the USA) in conjunction with Chinese government funding. The market grew too fast, creating huge amounts of stock. Lately, tax incentives in foreign countries (e.g. US and Germany) were reduced as the solar panel market matured. This resulted in stock going unsold and one of China's biggest solar panel producers declaring bankruptcy.

China's recent adoption of solar panels has been monumental, having added 12GW (GigaWatts) of solar generation capability in 2013 alone according to Bloomberg news [13], although Chinese media quoting industry sources reports at least 9.5 GW [14]. Regardless, this is the most any country has ever added in one year.

China is largely avoiding the issues faced by other countries as the vast majority of their solar energy comes from large scale providers rather than "prosumers".

2.5 Security Concerns

In nano-grid systems there is the potential for security and privacy risks.

2.5.1 Security

A distributed power system that uses Wi-Fi to communicate will inherit many of the security concerns that Wi-Fi itself faces (e.g. masquerade attacks, message replay, message modification, etc.). Due to the sensitive nature of electricity trading precautions should be carefully considered should this be implemented at scale. Proper encryption and authentication of data must be ensured at all times.

Encryption will also be necessary to implement a system that restricts monitoring of individual's power usage habits. If an individual or household's data can be monitored it can be invasively revealing.

2.5.2 Potential uses for exposed data

There are many possible uses for power usage profiling of users through either legal acquisition or data leakage considered by E.L. Quinn [15]:

- Car insurance providers could observe if you get enough sleep at night or if the time you leave for work is too late to get there on time without rushing.
- House insurance providers could observe if you often forget to turn off devices when you are not home.
- Advertisers could learn if you like gadgets and how much time you spend using them. They could observe if you often eat in and how you prepare your dinner.
- Uses in law enforcement, such as observation of suspected or known criminals.
- Medical uses could include monitoring various users with mental issues, for instance a bi-polar user may have a distinctive change in power usage depending on if they are taking their medication or not.

2.5.3 Intruders

An intruder could attempt to use the communication from nano-grids to assist them in several ways:

- Passive monitoring could be done on data transmitted in an attempt to profile a user's behaviour.
- Data analysis could be used to check when data is being sent which may be enough to know that someone is on holidays as there is no data being sent.
- There is potential to see if security systems are installed and their on/ off status.

While these concerns are avoidable it highlights some of the more unique security concerns such a system entails.

2.5.4 Privacy

With all of this information about user's electricity information there is a concern that unwarranted observation could take place. Laws regarding privacy of electricity usage is struggling to keep up with their necessity.

2.5.4.1 International Smart Grid Law

Netherlands A paper by F. D. Garcia et al. [16] proposes a security solution for smart grid systems after outlining the need for such a solution. "In April 2009 the Senate in the Netherlands has refused to pass a bill that made it compulsory for consumers to accept E-meters in their homes, precisely because of privacy concerns".

Europe In 2009 a smart grid task force was set up by the European commission. This task force has since issued key recommendations for standardisation, consumer data privacy and security.

In 2013 a proposal of security measures for smart grids was released [17]. The aim of this report was to help smart grid asset owners to define what is good practice, not to provide them with a set of minimum or appropriate security measures. While stricter measures that providers can be held to will be needed, this document does thoroughly catalog threats to a smart grid system and suggests related security controls to mitigate them.

New Jersey Plans were proposed in New Jersey for an upgrade to the grid infrastructure in the wake of hurricane Sandy. “7. (c) The installation and use of smart grid infrastructure and smart meters that would have the ability to alert the electric utility when and where electric utility service is lost, find service disruption locations quickly, and notify a customer when to expect restoration of service. The plan shall allow a customer to opt out of receiving a smart meter. Where the electric utility reduces meter reader personnel as a result of the operation of smart grid infrastructure, the electric utility shall recognize any existing employee bargaining unit and shall continue to honor and abide by any existing collective bargaining agreement for the duration of the agreement.” [18].

This legislation failed to make it through committee but shows what is being suggested in parts of America; the use of non-mandatory smart meters to increase grid intelligence.

California In 2011 California Public Utilities Commission released a robust article outlining many points of interest in recent privacy legislation being enforced on utility companies in relation to smart grid technology [19]:

- Regulation applies to both utility companies and any contractors that receive work from them.
- While rules protecting privacy and security are of great importance in relation to utility companies, rules are vitally also being put in place to govern how consumers

and third parties can access this information.

- Customers are entitled to information on electricity pricing and their usage costs including “bill-to-date, bill forecast data, projected month-end tiered rate, and notifications as the customers cross rate tiers”.
- Attempts are being made to give consumers real-time or near real-time pricing of electricity.

California is seen by many in America as paving the way in many of these issues as they are among the first to deal with them at scale.

America at Large It is highlighted by T. Kostyk et al. [20] that the smart grid is more difficult to regulate even than the internet as electricity inherently requires more regulation. Where the internet grew somewhat organically into what it is today, the smart grid is being encouraged to grow on top of an already established system. This has the smart grid being influenced by a large number of bodies, including state and federal government, before even its basic principles are established.

Electricity provision is a service that requires such a high level of reliability and must be treated with such a large degree of safety that any planned systems require an extremely well-thought through basis before they can even be considered for implementation.

Canada Canada has had large roll-outs of smart meters into businesses and homes, and has been the source of some very good suggestions for developing privacy and security practices [21]:

- Encouraging provincial regulators and utilities to consider open standards rather than proprietary solutions. Closed systems can claim to be safe but this is usually not believed by the security community. Open access to software must be provided to thoroughly test a given system and provide an objective grading of it.

- Establishment of a sub-committee to keep standards up to date as this technology develops.
- Establishment of a “Smart Grid Steering Committee” responsible for continued development of plans such as this one, and to “champion and promote key standards activities - filling identified gaps, reporting on progress, or suggesting steps to address delays or conflicts”.

2.6 Game Theory

This dissertation applies game theoretic concepts to algorithms in an attempt to improve load balancing in the electricity grid.

“Game theory is the formal study of conflict and cooperation. Game theoretic concepts apply whenever the actions of several agents are interdependent. These agents may be individuals, groups, firms, or any combination of these. The concepts of game theory provide a language to formulate, structure, analyze, and understand strategic scenarios.” [22].

The algorithm in Chapter 4.1.1 (p.27) makes use of game theoretic concepts; the basis of these will be outlined below.

A “game” in game theory must contain the following elements in order to be modelled:

- The players of the game
- The information and actions available to each player at a decision point (any point at which a player may perform an action)
- The payoffs (positive or negative) for each outcome of a decision point

In this dissertation there are two types of “agents” represented:

1. Electricity suppliers who are the primary providers of electricity in the grid.

2. Electricity users who buy fractions of electricity from their supplier.

For the purposes of this dissertation there will be only one supplier but many users.

2.6.1 Nash Equilibrium

A Nash equilibrium is an optimal solution to a game where no player stands to benefit by changing their current strategy if all other agents remain with their current strategies.

Games can potentially contain no Nash equilibrium or many Nash equilibria.

2.6.2 Pareto Optimal Solution

Pareto optimality is a measure of efficiency. “An outcome of a game is Pareto optimal if there is no other outcome that makes every player at least as well off and at least one player strictly better off” [23].

This means that a player in a Pareto optimal solution cannot improve their payoff without at least one other player reducing their payoff.

Nash equilibria need not be Pareto optimal solutions and this is often the case. A Nash equilibrium that is not Pareto optimal means that the players’ payoffs have the potential to increase.

2.7 Trading Algorithms

The goal of the trading algorithms considered in this dissertation is to perform load balancing in the electrical grid. The specific aim is to balance loads in such a fashion that it reduces the overall peak-to-average power ratio.

$$\text{Peak-to-Average Power Ratio} = \frac{\text{Peak Load}}{\text{Average Load}}$$

These algorithms require interaction between the established grid providers and intelligent machines inside the grid. These entities plan and communicate in an attempt increase their usage efficiency, effectively reducing their costs.

The algorithms outlined below represent research carried out in recent years.

P. Vytelingum et al. [24] propose a novel electricity marketplace in which bids to buy and sell electricity are matched together. The solution uses an online balancing method suited to this marketplace. It aims to prevent users from gaming the system (by providing information suggesting they would under-produce/under-supply to then be able to achieve higher profits with intra-day trading). In this way users in this system can be assumed to be acting “selfishly” without risk of harming the system. This solution also accounts for the risk of overloading transmission lines by design. While this work is very interesting it was deemed out of scope for this dissertation due to time constraints.

S. Y. Al-Agtash et al. [25] propose an evolutionary negotiation algorithm for power generating and power consuming companies in a distributed electricity market environment. While the use of an evolutionary algorithm (an algorithm that learns as it processes more information) is an interesting direction for this research, the process was found to need further testing considering real-time limitations, power availability, fault tolerance, and operational safety.

Chapter 3

Methodology

This dissertation aims to implement and analyse a load balancing algorithm for use between established grid operators and intelligent nano-grid systems. To do this it is necessary to create nano-grid systems capable of taking part in this algorithm and simulating more of these systems in an attempt to glean insight into their effectiveness at load balancing.

3.1 Architecture

3.1.1 Hardware

The popular credit-card sized computer ‘Raspberry Pi’ will be used to represent and manage each nano-grid.

The Raspberry Pi’s CPU is a 700 MHz ARM1176JZF-S core (from the ARM11 family using ARMv6 instruction set [26]). The ‘Model B’ version being used has 512 MB of RAM, 3 USB ports (one from the Broadcom BCM2835 system-on-a-chip, the other two available from a built-in USB hub). Its networking capabilities are provided by an Ethernet port, provided by the same USB hub [27]. It has the capacity to output video through both composite RCA and HDMI. In regards to audio output, it can use

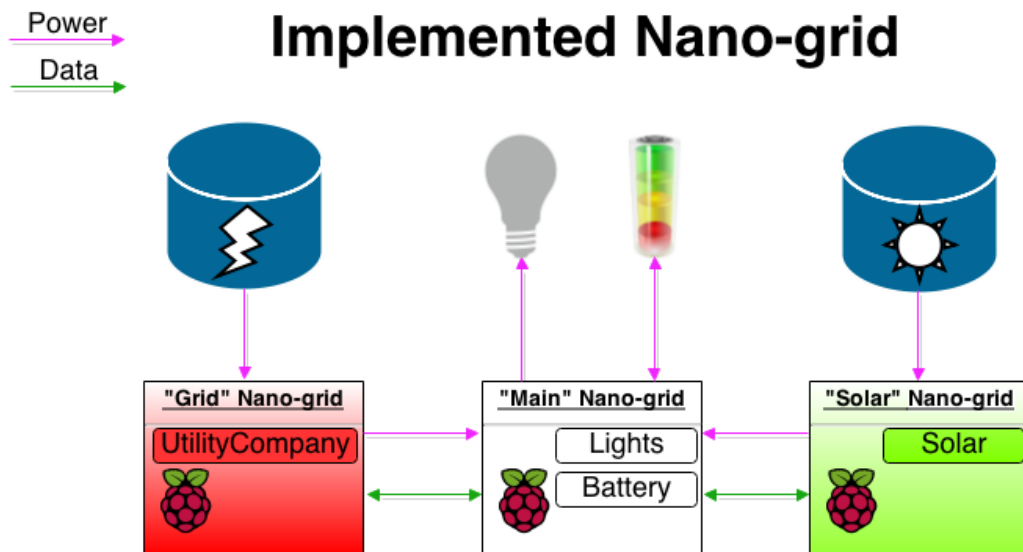


Fig. 3.1: Logical Representation of the Hardware Configuration Used

its 3.5 mm jack, HDMI, or I²S audio. It receives its power of 5 volts via MicroUSB. A USB Wi-Fi adapter is used in conjunction with the Raspberry Pis.

It contains all the aspects required of a nano-grid; capacity to trade electricity, potential to process the trading in an intelligent manner and the ability to interact with other grids.

A battery array will be used for power storage. Solar panels will be used as a supply as well as access to mains power. The load available to the nano-grids is a series of lights.

The hardware configuration was setup by Joseph Doyle, a post-graduate working on similar research in Trinity College Dublin.

In Fig. 3.1 the power and data connections present in the hardware configuration are clearly displayed. On the top left is a representation of mains electricity access and below it is the nano-grid that represents its intelligence. The central nano-grid is seen to control the lights and has access to a small battery. On the top right is the representation of the battery that holds power collected from the solar panels, below it is the nano-grid

that represents its interests.

3.1.2 Software

Several programming languages were considered before Python was ultimately deemed the most suitable for this dissertation.

C/C++

Due to the high efficiency of C and C++ they would be especially beneficial for use on the low-powered CPUs that were likely to be running this code.

Java

Java contains extremely robust libraries as well as increased portability through the use of the JVM (Java Virtual Machine). This portability does come at the cost of some efficiency however.

Python

When compiled Python runs natively on the Raspberry Pi (the hardware of choice for this dissertation). It is also recognised as a good language for prototypes due to the speed in which it can be developed.

While I am comfortable using any of the languages listed here, Python allows for the quickest adoption of any areas I may still be unfamiliar with because of its extremely simple API (Application Programming Interface). This reduces the risk of delays due to unforeseen coding issues.

3.1.2.1 External Libraries Used

quick2wire Provides utilities to enable I²C (Inter-Integrated Circuit), a multi-master serial single-ended computer bus, which enables devices with different clock speeds to

communicate. This library is used to take voltage readings from the ADC (Analog to Digital Converter).

RPi A Python GPIO (General-Purpose Input/Output) interface for the Raspberry Pi.

3.2 Testing Framework

Sourcing Information: SEMO (Single Electricity Market Operator)

The single electricity market is the wholesale electricity market operating in the Republic of Ireland and Northern Ireland. Information was used from SEMO to find the base load profile for a weekday in Ireland.

“The Single Electricity Market provides for a competitive, sustainable and reliable wholesale market in electricity aimed to deliver long-term economic and social benefits that are mutually advantageous to Northern Ireland and the Republic of Ireland. The market encompasses approximately 2.5 million electricity consumers, 1.8 million in the Republic of Ireland and 0.7 million in Northern Ireland.” [28]

3.3 Limitations of Methodology

3.3.1 Information

While the load profile taken from SEMO is exemplary real world data, it was necessary to use a proprietary function for costing electricity so that it could be used to update the total cost for each hour as users updated their planned usage of electricity. The cost function used is a strictly convex increasing function. This was chosen as it most suitably simulates how utility companies price their electricity.

3.3.2 Hardware

There was insufficient time in which to set up a larger quantity or variation of hardware on which to test the content of this dissertation. This means that only one user of the system existed with the rest being simulated.

Possible uses for further hardware is outlined in ‘Chapter 6.2 Future Work’ on p. 57.

3.3.3 Simulation

While users’ planned usage is added to the load profile for a day in simulations, this stored energy is not accessed by users during simulation. This would be beneficial in reducing peak demand by having simulated users “use” their stored energy during the peaks, effectively flattening the peak-to-average ratio even further. This level of simulation had to be descoped due to time restraints.

Chapter 4

Technical Work

4.1 Design

The design of this dissertation is split into two parts:

1. **Planned Trading Algorithm:** The logic needed to implement day-ahead planning of flexible electricity loads in order to reduce the peak-to-average ratio.
2. **Base Design/ Nano-grid Logic Design:** The logic needed for nano-grids to interact, basic operations (user interface, networking, hardware signaling, etc.) and how to abstract this to allow for different nano-grid implementations as required by hardware variation.

The design section begins by outlining the trading algorithm implemented in this dissertation, and then proceeds to explain the underlying nano-grid systems that were created to make use of the algorithm.

4.1.1 Planned Trading Algorithm

Nano-grids will be most useful when they are intelligently using storage. Storage will allow nano-grids to store cheap electricity at off-peak times and use this reserve when

the grid is stressed. This will serve the two-fold purpose of reducing cost for consumers and lowering peak usage, resulting in reduced cost for suppliers.

The algorithm used to attempt the intelligent use of storage technology and planning is proposed by Narahari et al [29]; a constrained tâtonnement (bargaining) strategy based on work done for server load balancing.

This algorithm is based on the idea that there is one supplier communicating with multiple consumers. One goal of the of the algorithm is to converge on a solution in a constant number of iterations regardless of the number of users. For this to be achievable the algorithm has to have very basic iteration design. An iteration of the algorithm is simple; (1) the utility company broadcasts the current plan of its production for the next day considering all the information it currently has and (2) users respond all at once with the times they plan to buy electricity based on this information.

These iterations are repeated until a solution is converged upon (until users stop updating their planned usage because they believe there would be no better plan than their current one).

Narahari et al. establish how to converge on a solution that is both a Nash equilibrium and Pareto optimal (game theoretic concepts as explained in Chapter 2.6 on p 19).

4.1.1.1 Game Theoretic Theorems

Below is a reference table for terms used in the following theorems. A load Profile

Table 4.1: Game Theory Term Reference

<i>Term</i>	<i>Definition</i>
I	All users where each user is i
T	All time-slots where each time-slot is t (There are 24 time-slots, one per hour)
l_i	$\{l_i^1, l_i^2, \dots, l_i^T\}$ The load vector of a user i over T time-slots
l_{-i}	The load vectors for all user other than i
L_t	The total amount of load in a time-slot t
C	The costing function of the electricity company
c_t	$C(L_t)$ The cost of electricity at a time-slot t
$\sum_t c_t$	The total cost of electricity generated for the given day
$\frac{\xi_i}{\sum_j \xi_j}$	The fraction of load i is using
$u_i(l_i, l_{-i})$	$-\frac{\xi_i}{\sum_j \xi_j} \sum_t c_t$ (The utility/payoff to a player i)

$(l_1^*, l_2^*, \dots, l_n^*)$ of all users' usage in a given time-slot is a Nash Equilibrium if:

$$u_i(l_i^*, l_{-i}^*) \geq u_i(l_i, l_{-i}) \forall l_i \in i$$

This condition would mean the current utility of the user is at least as good as any other options currently available to user.

A load Profile $(l_1^*, l_2^*, \dots, l_n^*)$ is Pareto optimal if $\nexists (l_1, l_2, \dots, l_n)$ such that:

$$u_i(l_i^*, l_{-i}^*) \leq u_i(l_i, l_{-i}) \forall i \text{ and}$$

$$u_i(l_j^*, l_{-j}^*) < u_i(l_j, l_{-j}) \text{ for some } i$$

With this condition there exists no load profile for this user whereby this user i can improve their payoff without a user j receiving a worse payoff (which is not something they will be willing to do).

Satisfying these conditions is important to ensure convergence of a solution. Users know from satisfying these conditions they have reached a solution they cannot improve upon without someone else choosing to worsen their own payoff (which would be illogical).

From this the following theorems are found:

1. A load profile $(l_1^*, l_2^*, \dots, l_n^*)$ is a Nash equilibrium if it is the solution of optimization problem $\min \sum_h C(L_h)$
2. If load profile $(l_1^*, l_2^*, \dots, l_n^*)$ is a solution of optimization problem $\min \sum_h C(L_h)$, then it is a Pareto optimal profile.

The following distributed algorithm aims to minimise $\sum_t C(L_t)$ to get a solution that is both a Nash equilibrium and Pareto optimal. This is sought by each user to reduce the cost of their load. The constraints in place on the algorithm ensure that when a user reduces their load/ cost, it reduces the total load/cost.

4.1.1.2 Algorithm Design

This algorithm is designed to balance all planned usage within a set number of iterations regardless of the the number of users that must be balanced. This section aims to show the logical aspects of the algorithm. Finer detail of the implementation (data types, etc.) is shown in Chapter 4.2.7.

This optimization of planned electricity use is carried out in the following steps:

1. The utility company begins by broadcasting its load and the associated cost for each time-slot (an hour) to all users.
2. Each user updates their planned use based on factors, including pricing and preference. Constraints* are in place on the amount a user may vary their usage from iteration to iteration.
3. Users send their updated usage plans to their electricity provider.

Planned Usage Trading Algorithm

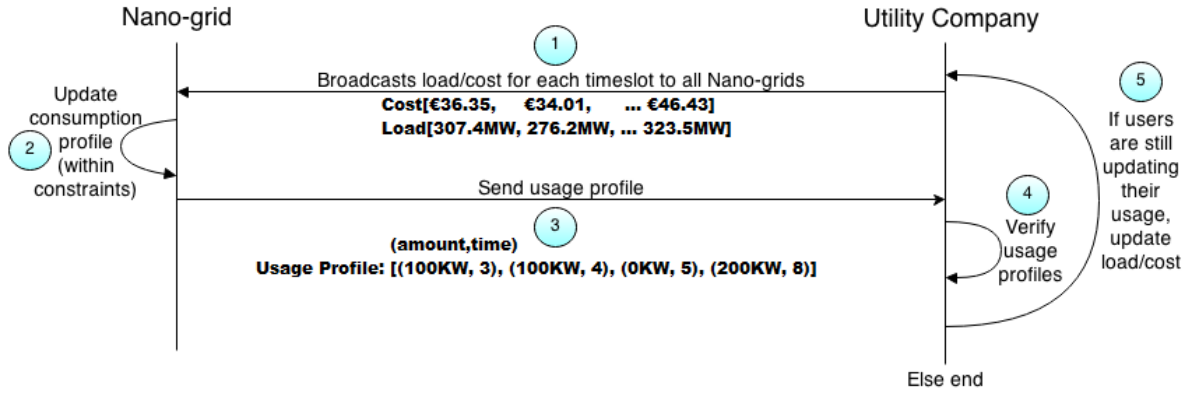


Fig. 4.1: Planned Trading Algorithm Overview - (Transmitted Data Shown)

- The utility company verifies that each customer's update is compliant with the constraints*. If loads vary so little that prices remain the same, then the algorithm has converged on optimum pricing and may end.
- While prices are still changing the utility company begins the next iteration of the algorithm. The algorithm can use an iteration limit to end the algorithm, providing a solution that has converged to a suitable degree.

*There are two constraints in place on users as they update their usage profiles:

The Inertia Constraint A portion of load may only be moved from one time-slot (j) to another (r) if j is sufficiently larger than r . This constraint helps ensure convergence as it stops small amounts of load being moved back and forth for trivial benefit as seen in the "Inertia Scenario" figure. Following is the calculation that must be carried out:

$$L_j \geq (1 + \epsilon)^3 L_r$$

where ϵ is a constant that when varied affects the speed/ accuracy of convergence. The constant ϵ is also defined by a theorem provided by Narahari et al.: "The

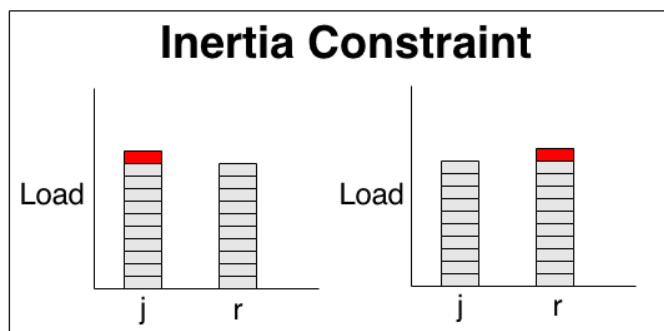


Fig. 4.2: Inertia Scenario

constrained tâtonnement protocol converges to a $(1 + \delta)$ approximation in $O(\frac{1}{\delta^6} \log \frac{H}{\delta})$ iterations with $\epsilon = O(\frac{\delta^2}{\log H})$ [29]. The value of δ is a constant set to affect the rate of convergence, H is the number of time-slots.

The Bounded Step Constraint This constraint imposes a maximum and minimum size on the percentage (p) of a load that can be moved from one time-slot (j) to another (r). This avoids a non-convergent scenario whereby large amounts are moved back and forth infinitely as users “herd” onto the cheapest time-slots as represented in the “Bounded Step Scenario” figure.

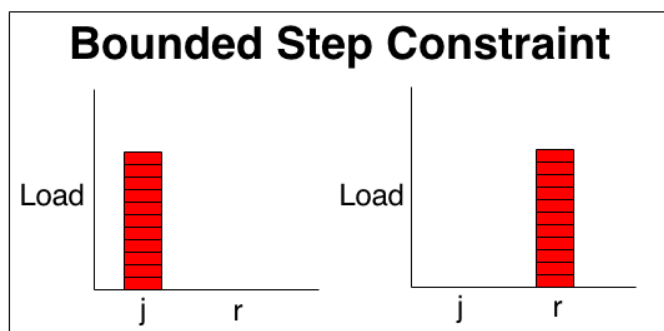


Fig. 4.3: Bounded Step Scenario

This constraint is implemented via two formulae: For $d =$ the iteration of the

algorithm and $\eta = \frac{\epsilon}{H^2}$

$$\eta \leq p_{ih} \leq 1 \text{ and}$$

$$p_{ih}^d / (1 + \epsilon) \leq p_{ih}^{d+1} \leq (1 + \epsilon) p_{ih}^d$$

Where p_{ih} is the percentage of load i wishes to move for a time-slot h .

From this it can be deduced that the minimum percentage of load movable for a given time-slot is $\min(\eta, p_{ih}^d / (1 + \epsilon))$ and the maximum is $\min(1, (1 + \epsilon) p_{ih}^d)$.

4.1.2 Base Design

The design of the real-world nano-grid requires the ability to autonomously and intelligently trade electricity. The concept of “intelligence” in the nano-grids refers to their ability to make the best available choices. They should aim to buy electricity at its cheapest price and sell it for maximum profit.

This section outlines the component parts that are needed to assemble a nano-grid’s logic.

4.1.2.1 Networking

Nano-grids communicate with one another by invoking each other’s remote functions. This is handled using Python’s XML-RPC library.

XML-RPC is a Remote Procedure Call method that uses XML passed via HTTP as a transport. With it, a user can call methods with parameters on another user that has been established as a server (the server is named by a URI (Uniform Resource Identifier)) and get back structured data. This library handles all the details of translating between conformable Python objects and XML on the wire.

4.1.2.2 User Interface

A simple user interface is created for each nano-grid class/ process. The graphical user interfaces display the options afforded to users for a given nano-grid and supply the

user with relevant information. The intention of these user interfaces is to be accessible either through an embedded display or via VNC (Virtual Network Computing). VNC transmits the keyboard and mouse events from one computer to another, relaying the graphical screen updates back in the other direction, over a network [30].

4.1.2.3 Hardware Signalling

As mentioned in Chapter 3.1.1, the hardware configuration was set up by Joseph Doyle. In conjunction with setting up the hardware physically, code to operate the hardware was provided. This code was subsequently refactored into a library that is utilised by all the nano-grids.

This library includes functions for reading voltage through the ADC (Analog-to-Digital Converter) and for granting access to electricity. A “connection” class is used to store information on the physical connections between devices, and control access between them.

4.1.3 Nano-grid Logic Design

A nano-grid system needs to be able to handle varying levels of operation depending on what it is controlling (e.g. loads, supplies, batteries). Even these individual categories provide a large amount of variance; a light will require different implementation to a heater. To handle this, a class hierarchy is established as seen in Fig. 4.4.

In this diagram the yellow classes are super-classes. They provide abstract functions that must be implemented. The bottom row of the diagram shows sub-classes derived from these super-classes and examples of additional functions they may require that are unique to their nano-grid configuration.

This approach has many benefits:

- Use of abstract functions allows for nano-grid specific configuration
- The standardised XML-RPC functions allow for interactions to be planned; a

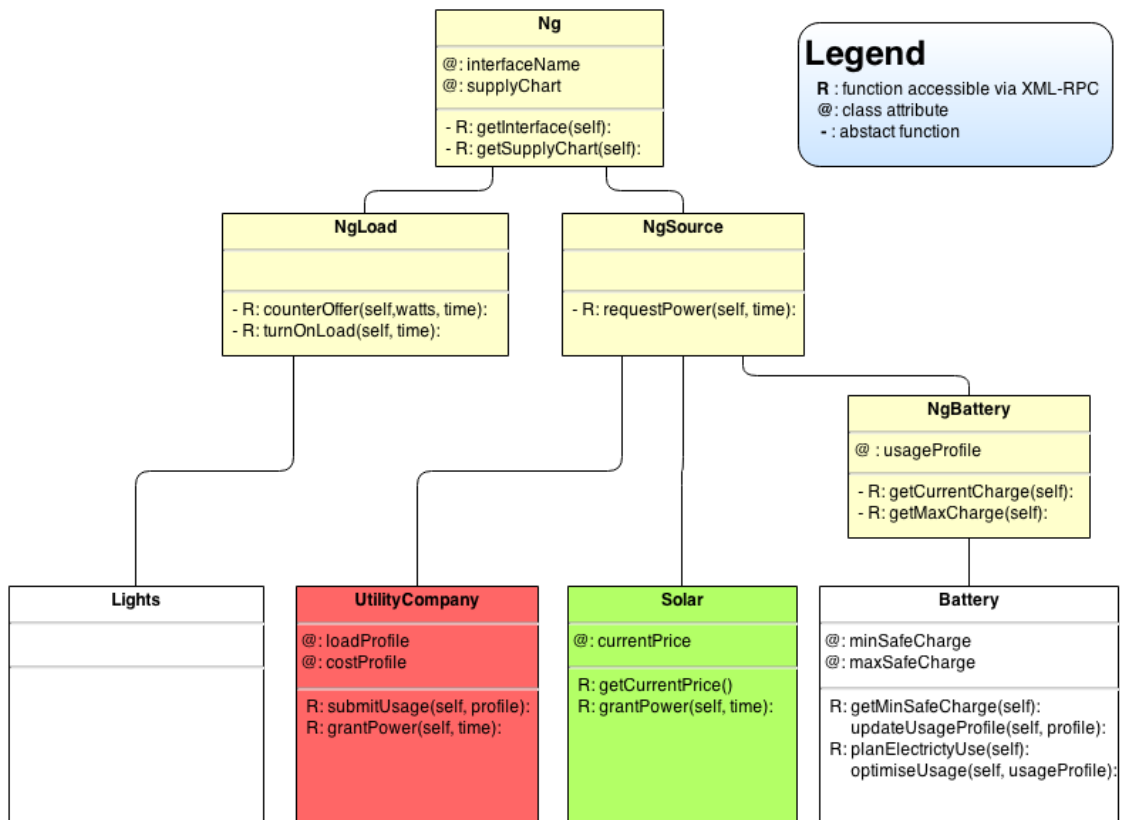


Fig. 4.4: Class Diagram

nano-grid a can inquire as to what a nano-grid's b 's interface is, e.g. "battery", and subsequently a will know that b will have a "getMinSafeCharge()" function

- Multiple instances of nano-grid classes can be run on a single controller (e.g. a nano-grid that controls both a light and battery), this avoids the need for variation super-classes like "BatteryAndLoad"
- Configuration of sub-classes outside of required super-class functions is left open to allow for whatever level of complexity is required for the device

4.2 Implementation

Operation of the Raspberry Pi requires Python 3 and as such it was the Python version of choice for this dissertation.

4.2.1 Networking

Nano-grids store information on other known connected nano-grids, this information is stored in a local XML (Extensible Markup Language) file. The XML file contains different information depending on the corresponding nano-grid.

A supply nano-grid will have data stored about it in the following format:

- name - a unique identifier for the supply (e.g. LabMains1)
- ipAddress - the I.P. address used to communicate with the nano-grid
- port - the port number the nano-grid communicates on
- price - the price of buying electricity from this supply
- lastUpdated - the UTC of the date and time that this information was last updated on
- green - whether this energy supply is sourced from a renewable source, a non-renewable source or mixed sources
- efficiency - the percent of electricity that is transferred from the source to this nano-grid after transmission losses are incurred
- gateway - the local physical gateway through which electricity from this nano-grid will be received

Currently the 'green' and 'efficiency' fields are unused as the size of the system is too small for such data to be relevant in decision making.

A load will have data stored on it in a similar way but does not require the price field.

The use of different port numbers allow multiple nano-grid classes to be run on the same physical nano-grid.

4.2.1.1 Initial Connection Handling

Initial connection handling is out of the scope but the code in place is intended to support it. Nano-grids would search the network for available devices when required and attempt to identify other nano-grids using the “getInterface” function. This would return an identifier that would notify the user as to what functionality the new nano-grid would afford (e.g. if it’s a battery it affords both the possibility of buying and selling electricity).

4.2.1.2 Utility Company: Submitting Usage Profiles

For the scope of this dissertation there is only one real nano-grid that submits a usage profile and balances its usage accordingly (the main grid’s local battery), the rest of the usage profiles are simulated inside the utility company.

Utility companies have an RPC function, “submitUsage”, that when called begins the algorithm seen in Chapter 4.1.1.2, Fig. 4.1 on p31. The “submitUsage” function takes two parameters: the user’s new usage profile and the user’s old usage profile.

A user’s usage profile is submitted as a list of tuples (paired values). The first item in the tuple is the amount of load desired for a given time-slot, the second item is said time-slot. This gives the utility company various pieces of information:

- the load desired per time-slot
- the time-slots the user is willing to distribute its loads over
- the changes the user has made since the last iteration of the algorithm

If this is the first iteration of the algorithm the user sends an empty list as the old usage profile, the utility company recognises this and knows there is no need to check the constraints for the first stage of the algorithm. This algorithm implementation is covered in Chapter 4.2.7.

4.2.2 Abstract Classes

As seen in Fig. 4.4 on p.35, the nano-grid processes are all created as sub-classes of appropriate super-classes. Below points of interest in these super-classes are outlined.

Super-class: Ng The “Ng” super-class is at the top of the class hierarchy. It provides two basic functions; “getInterfaceName” and “getSupplyChart”. The “getInterfaceName” function has previously been outlined in Chapter 4.2.1.2.

The “getSupplyChart” function is used to acquire a nano-grid’s list of attached supplies, the “supply chart” data is outlined in Chapter 4.2.1.

Super-class: NgLoad The “NgLoad” super-class inherits from the “Ng” super-class. It is designed as a super-class for loads to inherit. It contains two functions; “counterOffer”, and “turnOnLoad”.

The “counterOffer” function is designed for suppliers to use in response to this load making an offer for electricity at a given price. Its parameters are “watts”, the quantity of electricity being proposed, and “time”, the amount of time this quantity of electricity will be supplied for.

The “turnOnLoad” function is not a remotely accessible function, it is simply a required function for all loads to ensure they can be used. More complex ways of utilising a given load can be specified in the implemented sub-class but this function is the base requirement. This function simply requires the desired amount of time to turn on the load as its only parameter.

Super-class: NgSource The “NgSource” super-class inherits from the “Ng” super-class. It is designed as a super-class for sources to inherit. It contains a single function; “requestPower”.

The “requestPower” function has two parameters, “watts” and “time”, and is an RPC function that allows loads to request power from this source. Requests can be negotiated using the load’s “counterOffer” function. This functionality is planned as possible future work but is out of scope for this dissertation.

Super-class: NgBattery The “NgBattery” super-class inherits from the “Ng” and “NgSource” super-classes. It is designed as a super-class for batteries to inherit. It contains two functions; “getCurrentCharge”, and “getMaxCharge”.

These functions take no parameters and work as one would expect. They are used to provide information relating to how much electricity could be bought or sold to a given battery.

4.2.3 User Interface

Simple graphical interfaces are provided for each nano-grid class but the amount of content required for each varies. Each class gets its own window with a title.

The title displays the class type and informs the user of relevant information, example shown in Fig. 4.5. The title in Fig. 4.5 informs the user that their solar supply is currently “Idling” (not generating or supplying electricity).

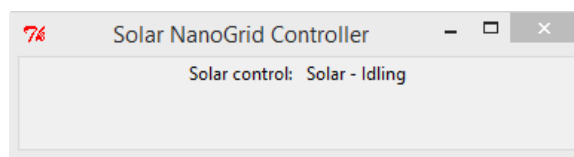


Fig. 4.5: Solar Nano-grid Graphical User Interface

Both the “utility company” and “solar” nano-grid classes have minimal user interface, they simply display information. The “light” and “battery” user interfaces afford the

users more options.

4.2.4 Light Interface

To control the lights the interface allows users to request the lights be turned on for a set amount of time and visual feedback of this is given to the user, as shown in Fig. 4.6 and Fig. 4.7.

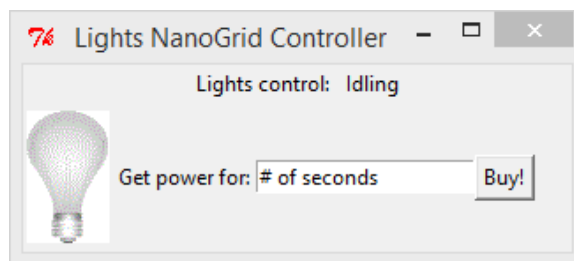


Fig. 4.6: Light Nano-grid Graphical User Interface (Idling)

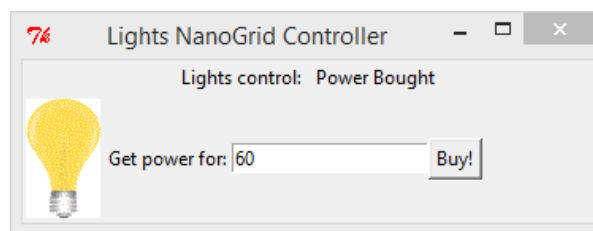


Fig. 4.7: Light Nano-grid Graphical User Interface (Running)

4.2.5 Battery Interface

The “battery” interface works in conjunction with the planned usage algorithm, see Fig. 4.8. The title shows the current charge of the battery and this figure can be dynamically updated as the charge changes. The interface allows users to specify how much supply they would like, this is given as an integer that represents kilowatt-hours. Time-slots are represented in the user interface as a tick-boxes labeled from 0 (midnight)

to 23 (11pm), ticked boxes represent the hours you are willing to accept planned load during.

A preferred time-slot can optionally be set also, this serves only to show a user how much they save by planning their usage versus simply using a given time-slot.

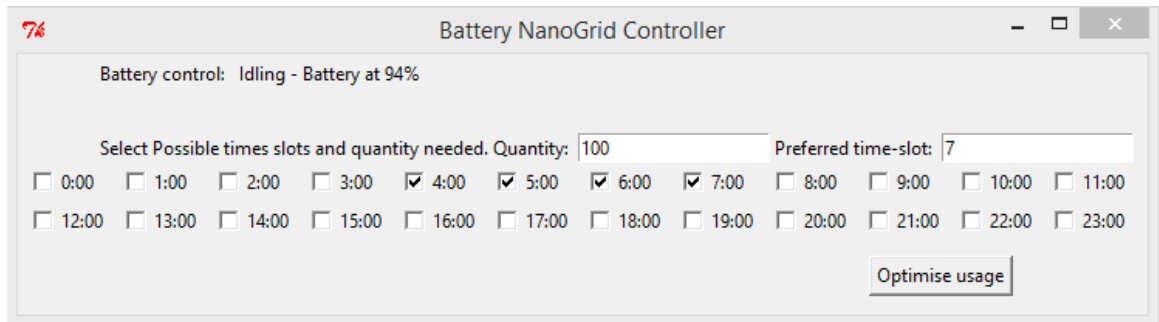


Fig. 4.8: Battery Nano-grid Graphical User Interface

4.2.6 Feature Breakdown

Here the code implementation of the previously outlined design is broken down into individual features.

4.2.7 Optimising Planned Usage

This is the implementation of the algorithm outlined in Chapter 4.1.1 on p27 with technical details explained.

1. The utility company broadcasts its load and associated cost for each time-slot (an hour). This is sent in the form of two lists of length twenty-four (the number of hours in a day) as shown in Fig. 4.9 below point 1. The first list is the load profile, the amount of electricity being requested of the supplier per hour in MWhrs (MegaWatt Hours). The second list is the cost profile, the cost of electricity for a MWhr in Euro for each hour.

Planned Usage Trading Algorithm

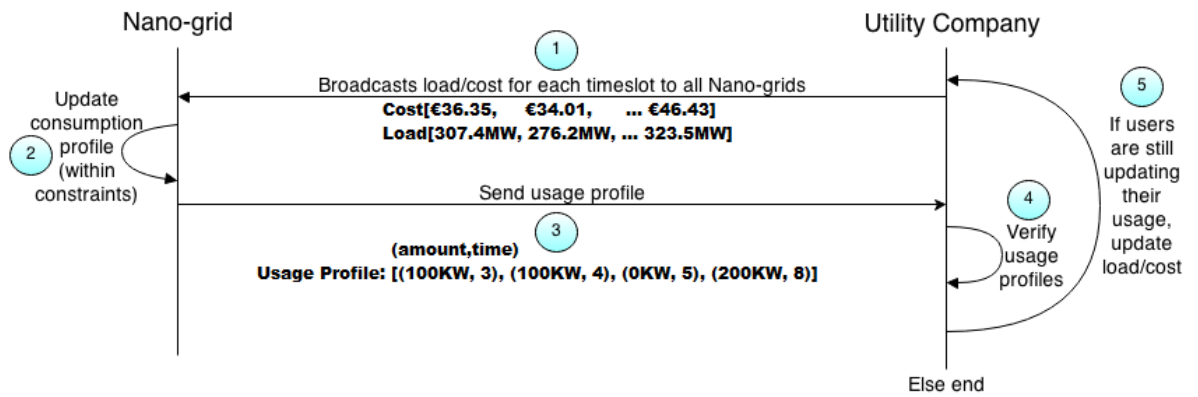


Fig. 4.9: Planned Trading Algorithm with Example Transmission Data

- Each user updates their planned usage within the constraints*. This usage is stored and transmitted as a list of tuples, as described in Chapter 4.2.1.2. The code used to implement this is provided in Appendix A.
- Users send their updated usage plans (data as outlined below point 3 in Fig. 4.9) to their electricity provider.
- The utility company verifies that each customer's update is compliant with the constraints*. If loads vary so little that prices remain the same, then the algorithm has converged on optimum pricing and may end.
- While prices are still changing the utility company begins the next iteration of the algorithm. The algorithm iteration limit can be set to any desired number to end the algorithm, providing a solution that has converged to a defined degree.

*The usage profile must be updated within the constraints provided here:

The Inertia Constraint The code used to implement this is provided in Appendix B on p.68. This condition is absolute, it doesn't matter how much load you plan to

move only how much load already exists in the given time-slots.

The Bounded Step Constraint This condition is relative, it depends on the percentage of load a user already has in two given time-slots.

The code used to implement this is provided in Appendix A built into the *optimiseUsage* function. The code relating to the bounded step constraint is primarily based around the final *if* statement.

4.2.8 Finding the Cheapest Source

When a user makes an unplanned purchase of electricity, as seen in Fig. 4.7(p.40), an intelligent choice as to where to buy electricity must be made quickly.

Two systems were considered for this operation:

1. Begin initially using local battery supplies while negotiations take place between nano-grids
2. Update available sources at set intervals and simply choose the cheapest when electricity is required

The second option was chosen as it can be used with nano-grids regardless of their storage potential.

Utility electricity prices can be sourced from SEMO as needed (with prices set for 30 minute time intervals).

For the scope of this dissertation renewable electricity is priced based on the capacity of the battery it is stored in. It is made exceedingly cheap when the battery is full to ensure it is bought. When the battery goes below 85% the battery suffers decay to its total life-span, to ensure battery life is maintained the electricity becomes very expensive at this point (relative to mains electricity). This system works because electricity traded behind a single user's electricity meter is essentially free (the user already owns it), behind the meter prices serve only to optimise usage to reduce costs.

Chapter 5

Results

The primary use case for the given system consists of household users planning to charge batteries with a small to moderate amount of electricity (approximately 10-100KWhrs). The testing undertaken for this use case seeks to find the effectiveness of the planning algorithm for different quantities of users. With this in mind there are several variables to be considered throughout testing:

1. The number of users
2. The constant δ that affects the speed/accuracy of convergence
3. The number of time-slots available to users

To evaluate the effectiveness of these tests the constant δ is kept constant at the value '0.5'. The number of time-slots available to users is randomly assigned per user. The quantity of load each user requests is also randomly chosen. It is then equally divided among the users' available time-slots before the algorithm begins, this is represented as iteration 0 in the following graphs.

5.1 Balancing 10,000 Users

5.1.1 Load Profile at Iteration 0

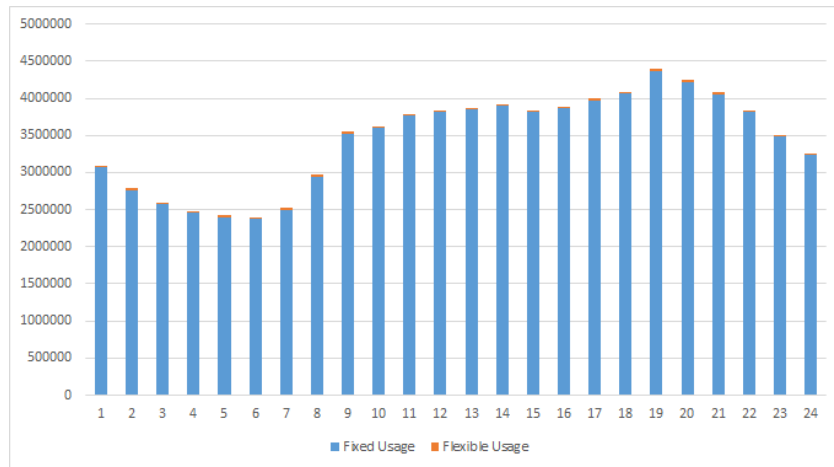


Fig. 5.1: 10K Users balancing their loads equally among their given time-slots

5.1.2 Load Profile at Iteration 1

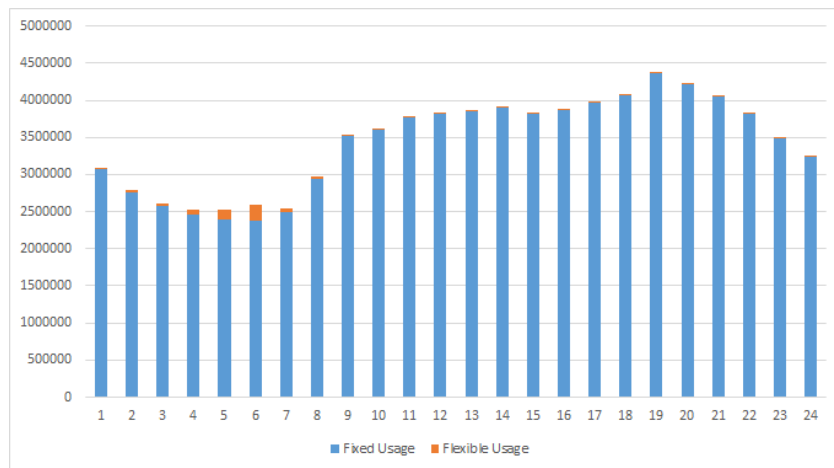


Fig. 5.2: 10K Users after iteration 1 of the planning algorithm

5.1.3 Load Profile at Iteration 2

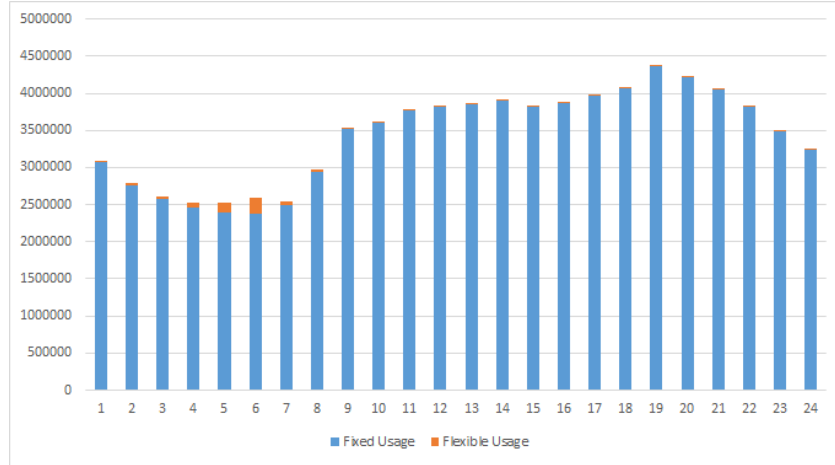


Fig. 5.3: 10K Users after iteration 2 of the planning algorithm

5.1.4 Peak-to-Average Ratio over Iterations

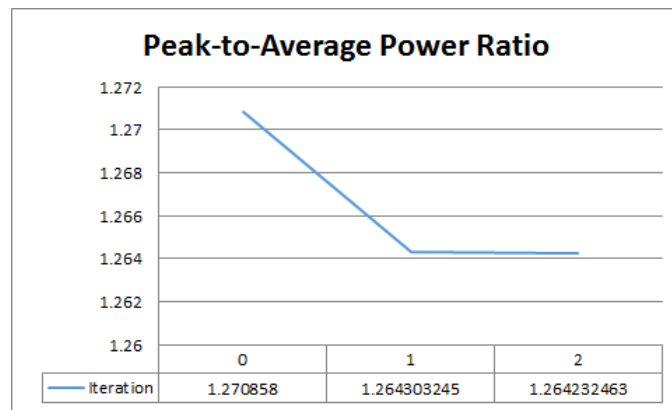


Fig. 5.4: The peak-to-average power ratio of total load throughout balancing

Observations 10,000 users represent a negligible proportion of the overall usage for a given day as seen in the given graphs. While this is true there is still a quantifiable reduction in the peak-to-average ratio as seen in Fig. 5.4.

5.2 Balancing 50,000 Users

5.2.1 Load Profile at Iteration 0

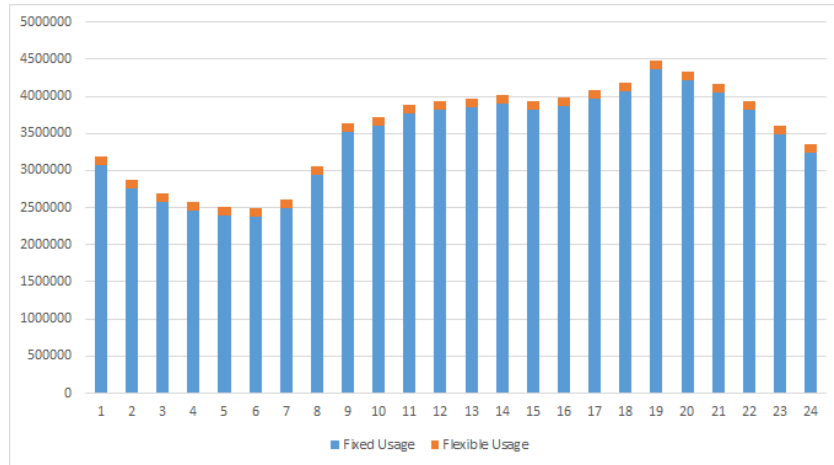


Fig. 5.5: 50K Users balancing their loads equally among their given time-slots

5.2.2 Load Profile at Iteration 1

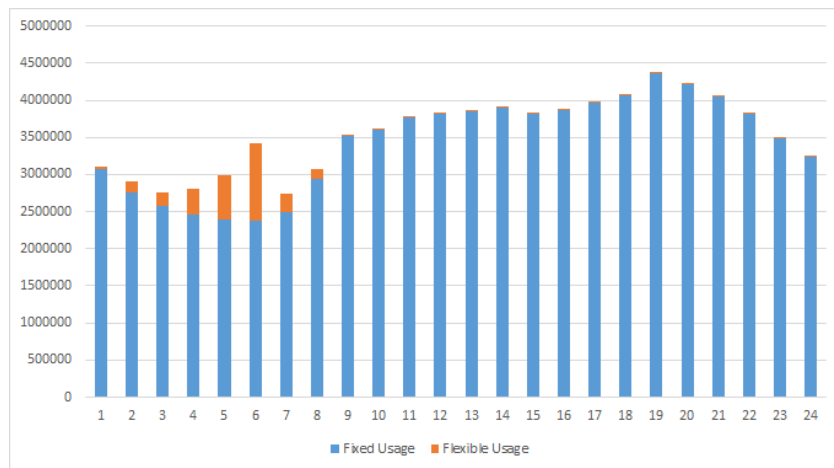


Fig. 5.6: 50K Users after iteration 1 of the planning algorithm

5.2.3 Load Profile at Iteration 2

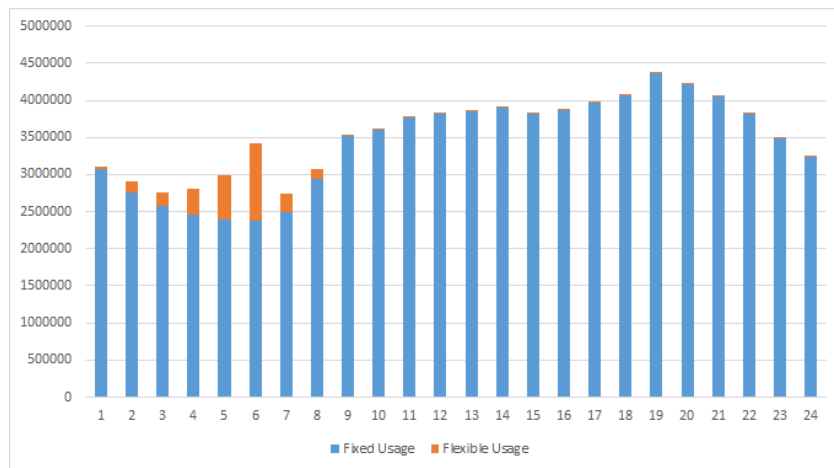


Fig. 5.7: 50K Users after iteration 2 of the planning algorithm

5.2.4 Load Profile at Iteration 3

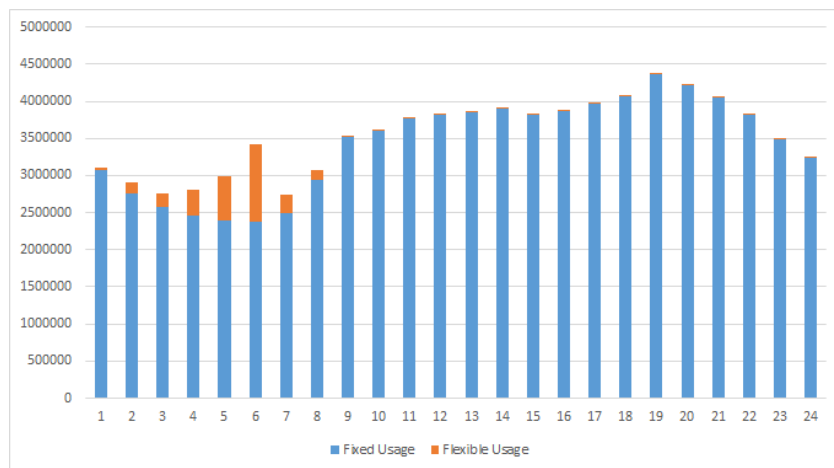


Fig. 5.8: 50K Users after iteration 3 of the planning algorithm

5.2.5 Peak-to-Average Ratio over Iterations

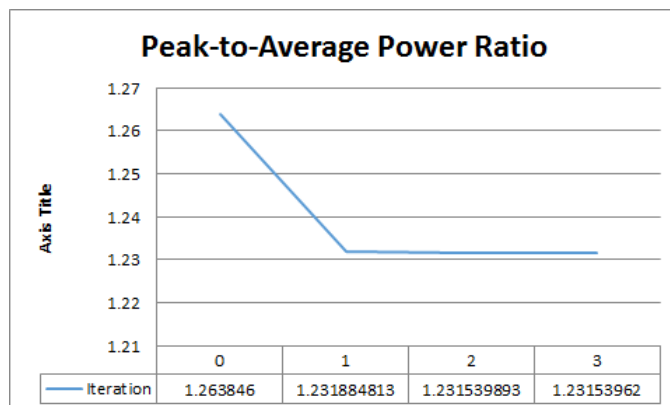


Fig. 5.9: The peak-to-average power ratio of total load throughout balancing

The balancing of 50,000 users' planned usage highlights the intended nature of the algorithm. Users seek the cheapest time-slots, and as time-slots become more balanced, trading ends.

5.3 Balancing 100,000 Users

5.3.1 Load Profile at Iteration 0

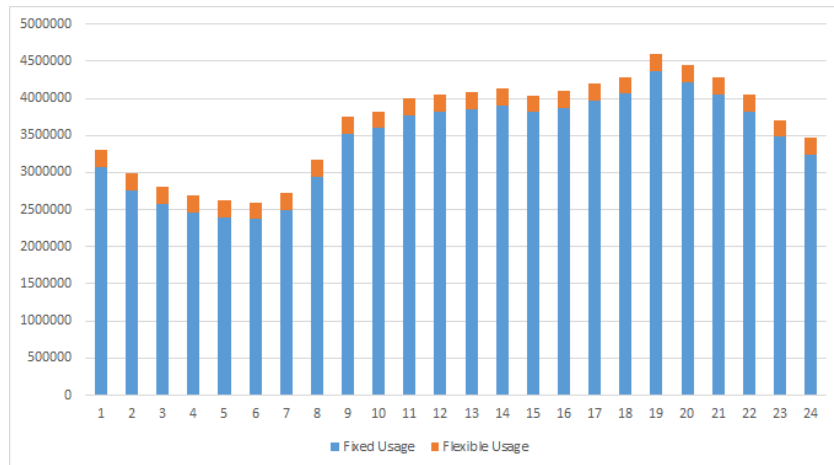


Fig. 5.10: 100K Users balancing their loads equally among their given time-slots

5.3.2 Load Profile at Iteration 1

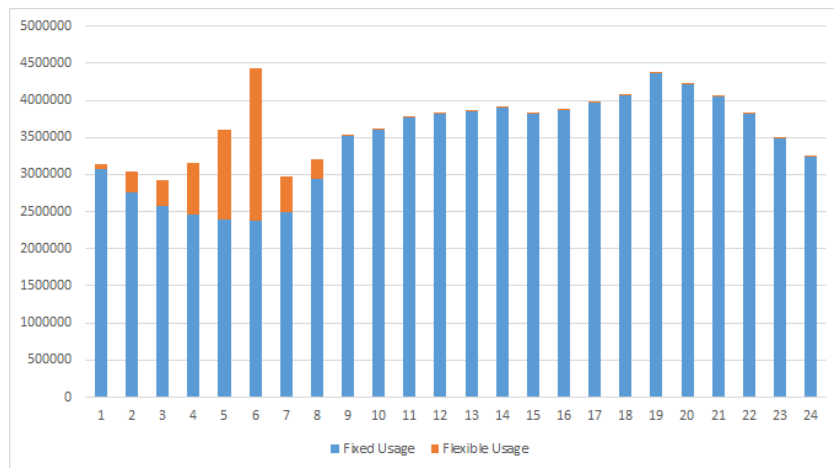


Fig. 5.11: 100K Users after iteration 1 of the planning algorithm

5.3.3 Load Profile at Iteration 2

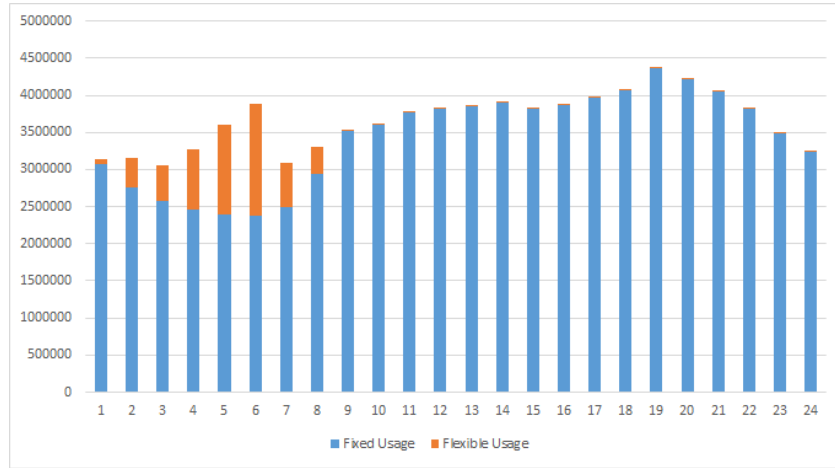


Fig. 5.12: 100K Users after iteration 2 of the planning algorithm

5.3.4 Load Profile at Iteration 3

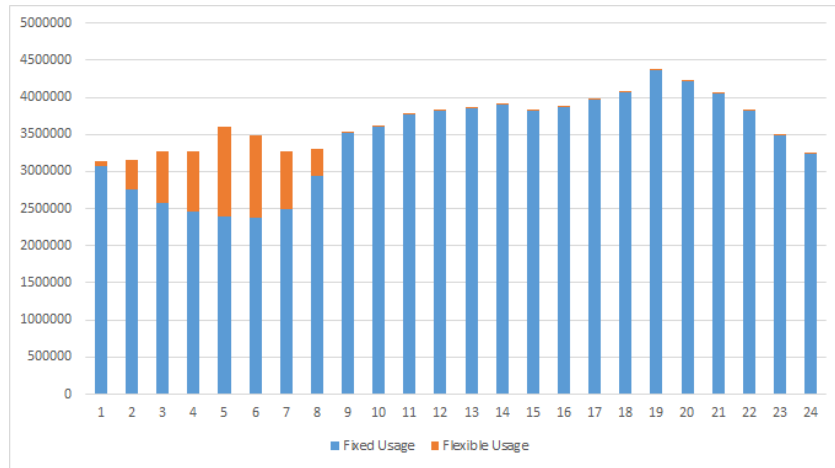


Fig. 5.13: 100K Users after iteration 3 of the planning algorithm

5.3.5 Peak-to-Average Ratio over Iterations

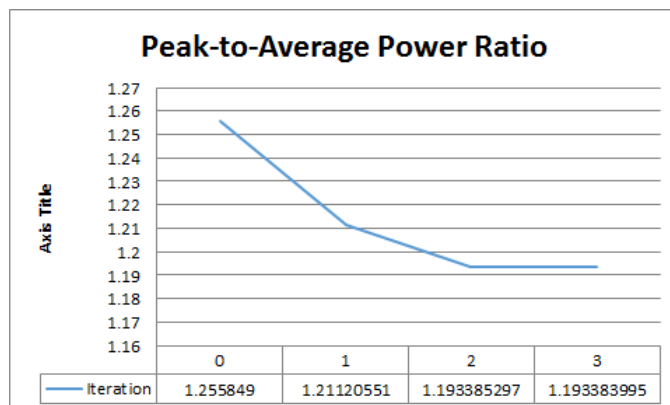


Fig. 5.14: The peak-to-average power ratio of total load throughout balancing

100,000 users show clear improvement in the peak-to-average ratio. There is a reduction of 4.973% at iteration 3 versus the trivial case at iteration 0.

5.4 Balancing 200,000 Users

5.4.1 Load Profile at Iteration 0

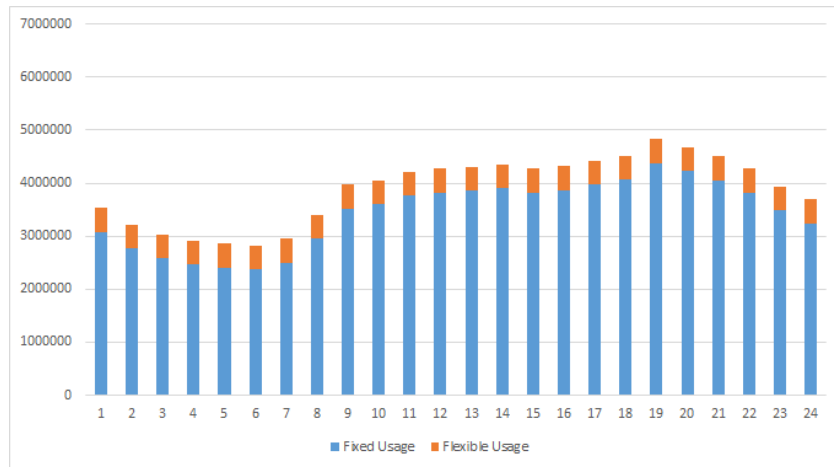


Fig. 5.15: 200K Users balancing their loads equally among their given time-slots

5.4.2 Load Profile at Iteration 1

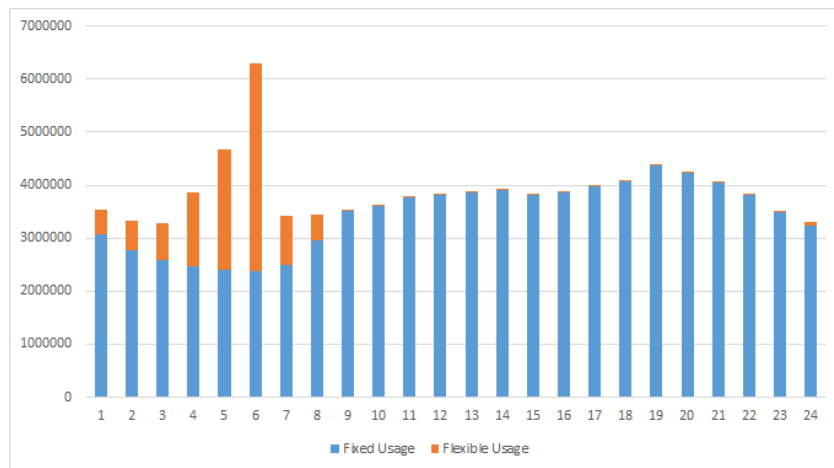


Fig. 5.16: 200K Users after iteration 1 of the planning algorithm

5.4.3 Load Profile at Iteration 2

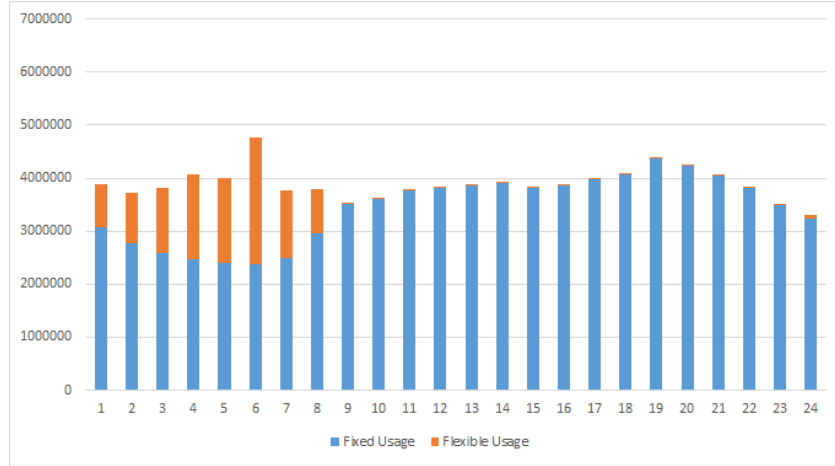


Fig. 5.17: 200K Users after iteration 2 of the planning algorithm

5.4.4 Load Profile at Iteration 3

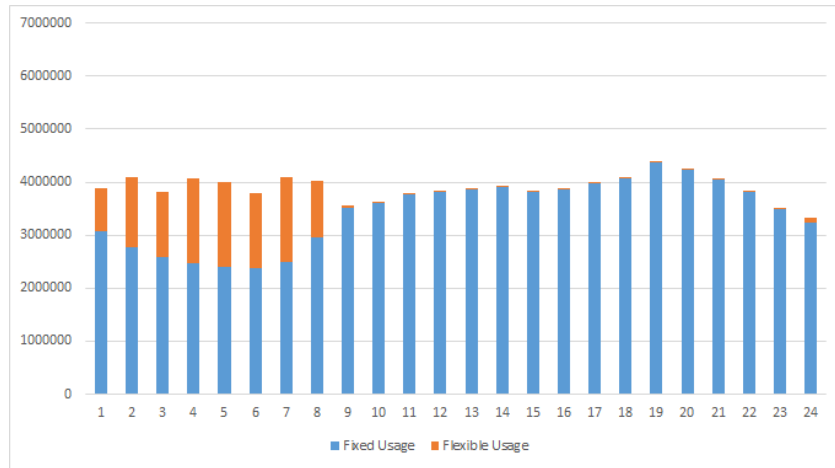


Fig. 5.18: 200K Users after iteration 3 of the planning algorithm

5.4.5 Peak-to-Average Ratio over Iterations

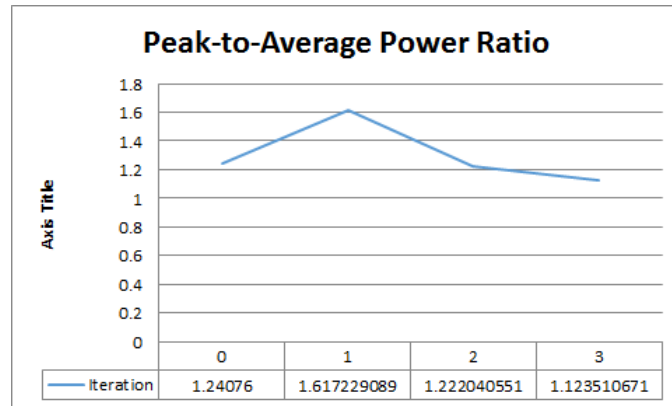


Fig. 5.19: The peak-to-average power ratio of total load throughout balancing

The use of 200,000 simulated users initially makes it look like it would be less effective to use the proposed bargaining algorithm (iteration 1, Fig. 5.16) rather than just trivially balancing loads (iteration 0, Fig. 5.15). The first iteration of the algorithm produces a peak during the early morning much larger than any other throughout the day.

Even with this somewhat extreme example the algorithm shows its constraints are effective with the final peak-to-average ratio at iteration 3 (Fig. 5.19) coming in lower than if the algorithm had not been used.

Chapter 6

Conclusion

6.1 Evaluation

6.1.1 Nano-grid Class Hierarchy

The class hierarchy proposed in this paper offers a simple framework for developers to begin creating nano-grid software without a steep learning curve.

Its use of Python and XML-RPC would limit it to a testing environment but any code designed could be refactored into another mainstream language with relative ease should the need arise.

6.1.2 Planned Usage Algorithm

The implementation of the tâtonnement bargaining strategy for reducing the peak-to-average power ratio displays clearly how the algorithm can be logically implemented in a object-oriented/imperative programming style.

It also suggests possible data-flows between users of the algorithm, and outlines how it performs in given contexts.

The results of the simulations does lead to the question of if it is cost effective to balance loads in this way. The final set of simulation results for 200,000 users has quite

a small return but the limitations of the testing, as outlined in Chapter 3.3.3, would imply that more advanced simulation is necessary.

6.2 Future Work

The amount achievable in this dissertation was quite limited in comparison to the vast amount of research possible in the developing nano-grid field.

There are any number of ways in which to approach future research, and in choosing any of these paths even more topics would likely become apparent. Outlined below are some areas most readily apparent for future work.

6.2.1 Scaled Testing

Larger groups of inter-connected nano-grids would require testing and possibly new algorithms depending on how they perform at scale. It would be interesting to test these nano-grids both attached and removed from the grid.

6.2.2 Value Testing

If realistic hardware (most likely an embedded solution) were sufficiently developed it would become feasible to calculate how long it would take to recuperate the cost of buying and running the hardware. Mass production would reduce costs, market saturation would increase usefulness, but in order to be practical a nano-grid system would have to be value-adding for users from a very early stage.

6.2.3 Interesting Use Cases

Nano-grids have many applications and scenarios in which they have the potential to operate. Each of these will likely require individual testing:

- Nano-grids used in data centers to increase redundancy versus current solutions.

- Portability contexts whereby it would be disconnected from the grid and re-connected in a new location.
- Emergency situations whereby electricity becomes short in supply.
- After a black-out, how would a battery saturated market avoid creating massive demand on the grid as it becomes available once more.

6.2.4 Higher Level of Automation

Nano-grids that analysed and profiled a user's usage could automatically create its own usage profile and balance these with the utility company without the need for user interaction with the nano-grid.

Bibliography

- [1] B. Nordman, K. Christensen & A. Meier. Think globally, distribute power locally: The promise of nanogrids. In *Computer*, pages 89–91. ©IEEE, 2012.
- [2] Fraunhofer Institute for Solar Energy Systems ISE. Electricity production from solar and wind in germany in 2012. Online, February 2013. ‘<http://www.ise.fraunhofer.de/en/downloads-englisch/pdf-files-englisch/news/electricity-production-from-solar-and-wind-in-germany-in-2012.pdf>’.
- [3] M.M. He, E.M. Reutzler, Jiang Xiaofan, R.H. Katz, S.R. Sanders, D.E. Culler & K. Lutz. An architecture for local energy generation, distribution, and sharing. In *Proceedings of the 2008 Energy 2030 Conference. IEEE*, pages 1–6. ©IEEE, 2008.
- [4] B. Nordman. Nanogrids - evolving our electricity systems from the bottom up, May 2011. ‘<http://nordman.lbl.gov/docs/nano.pdf>’.
- [5] I. Cvetkovic, D. Boroyevich, P. Mattavelli, F.C. Lee, D. Dong, L. Jiang, and Y. Chang. Future home dc-based renewable energy nanogrid system. In *Proceedings of the 2010 Conference on Grand Challenges in Modeling & Simulation, GCMS '10*, pages 337–343, Vista, CA, 2010. Society for Modeling & Simulation International.
- [6] EMerge Alliance. Alliance FAQs. Online, 2014. ‘<http://www.emergealliance.org/About/AllianceFAQs.aspx>’.

- [7] I. Cvetkovic, D. Boroyevich, P. Mattavelli, F.C. Lee and D. Dong. Non-linear, hybrid terminal behavioral modeling of a dc-based nanogrid system. In *Proceedings of the 2011 Twenty-Sixth Annual IEEE Conference and Exposition (APEC)*, pages 1251–1258. ©IEEE, 2011.
- [8] R. Adda, O. Ray, S.K. Mishra and A. Joshi. Synchronous-reference-frame-based control of switched boost inverter for standalone dc nanogrid applications. In *Power Electronics, IEEE Transactions on*, pages 1219–1233. ©IEEE, 2013.
- [9] S.U. Arun, S.S. Nag and S.K. Mishra. A Multi-Input Single-Control (MISC) battery charger for DC nanogrids. In *ECCE Asia Downunder (ECCE Asia), 2013 IEEE*, pages 304–310. ©IEEE, 2013.
- [10] Litos Strategic Communication. The SMART GRID: An Introduction. *The U.S. Department of Energy*, pages 1–48, 2008.
- [11] Weidmuller. The basics of overvoltage protection. Online, September 2007. ‘http://www.elektrospoji.si/uploads/menu_download/file_276_vse_o_prenapetostni_zasciti.pdf’.
- [12] Hawaiian Electric. Hawaiian electric leads national project to reliably integrate solar energy. Online, April 2014. ‘http://www.hawaiianelectric.com/heco/_hidden/Hidden/CorpComm/Hawaiian-Electric-leads-national-project-to-reliably-integrate-solar-energy’.
- [13] Christopher Martin. China added 12 gigawatts as top solar market in 2013, bnef says. Online, January 2014. ‘<http://www.businessweek.com/news/2014-01-23/china-installed-12-gigawatts-as-top-2013-solar-market-bnef-says>’.
- [14] NE21. ”2013 china’s pv installed capacity 9.5gw, accumulated over 18gw (*translated)”. Online, January 2014. ‘<http://www.ne21.com/news/show-51365.html>’.

- [15] Elias Leake Quinn. Privacy and the new energy infrastructure. Online, February 2009. ‘<http://dx.doi.org/10.2139/ssrn.1370731>’.
- [16] Flavio D. Garcia and Bart Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In *Proceedings of the 6th International Conference on Security and Trust Management, STM’10*, pages 226–238, Berlin, Heidelberg, 2011. Springer-Verlag.
- [17] Smart Grid Task Force. Proposal for a list of security measures for smart grids. In *Proposed security measures for smart grids*, pages 1–54. European Commission, 2013.
- [18] Sponsored by Rep Angel Fuentes and Rep Amy H. Handlin. Public Utility Reliability Investment Act. Online, February 2013. ‘<http://legiscan.com/NJ/text/A3816/id/734348>’.
- [19] California Public Utilities Commission. Decision Adopting Rules To Protect The Privacy And Security Of The Electricity Usage Data Of The Customers Of Pacific Gas And Electric Company, Southern California Edison Company, And San Diego Gas & Electric. Decision 11-07-056. Online, July 2011. ‘<https://www.cdt.org/files/pdfs/PUC%20Smart%20Grid%20Final.pdf>’.
- [20] Timothy Kostyk and Joseph Herkert. Societal implications of the emerging smart grid. *Commun. ACM*, 55(11):34–36, November 2012.
- [21] CNC/IEC Task Force on Smart Grid Technology and Standards. The Canadian Smart Grid Standards Roadmap: A strategic planning document. Online, October 2012. ‘http://www.scc.ca/sites/default/files/publications/Smart-Grid-Report_FINAL_OCT2_EN.pdf’.
- [22] Theodore L. Turocy and Bernhard von Stengel. Game theory. In *(Draft*

for) *Encyclopedia of Information Systems*, Academic Press, CDAM Research Report LSE-CDAM-2001-09, pages 1–39, 2001.

- [23] Mikhael Shor. ‘Pareto Optimal’. Dictionary of Game Theory Terms, Game Theory .net. ‘<http://www.gametheory.net/dictionary/CitationInformation.html>’. Online: Accessed 13/05/14.
- [24] Perukrishnen Vytelingum, Sarvapali D. Ramchurn, Thomas D. Voice, Alex Rogers, and Nicholas R. Jennings. Trading agents for the smart electricity grid. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS ’10, pages 897–904, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.
- [25] Salem Y. Al-Agtash and Amjed A. Al-Fahoum. An evolutionary computation approach to electricity trade negotiation. *Adv. Eng. Softw.*, 36(3):173–179, March 2005.
- [26] Broadcom. High definition 1080p embedded multimedia applications processor. Online, 2013. ‘<http://www.broadcom.com/products/BCM2835>’.
- [27] Standard Microsystems Corp. Industry’s first single-chip, hi-speed usb 2.0 hub and high-performance 10/100 ethernet controllers. Online, 2013. ‘<http://www.smsc.com/index.php?tid=300&pid=135>’.
- [28] SEMO (Single Electricity Market Operator). The single electricity market. Online. Accessed on: April 2014. ‘<http://www.semo.com/AboutSEMO/Pages/default.aspx>’.
- [29] Narahari, Jain, Shweta, Balakrishnan and Narayanaswamy, Yadati and Hussain, Saiful A. and Voo, Nyuk Yoong. Constrained tâtonnement for fast and incentive compatible distributed demand management in smart grids. In *Proceedings of the*

Fourth International Conference on Future Energy Systems, e-Energy '13, pages 125–136, New York, NY, USA, 2013. ACM.

- [30] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1):33–38, January 1998.

Appendix A

Code

The Function used by Nano-grids to Optimise their Usage
Within the Restraints

```
def OptimiseUsage(self, aUsageProf, costProfile):
    totalUsage=0
    pAmount=[]
    p=[]      # the percentage
    minAmountInP=[]
    maxAmountInP=[]

    for (amount, ts) in aUsageProf:
        totalUsage += amount

    for (amount, ts) in aUsageProf:
        pAmount.append(amount)
        # % amount of the total load for a given Time-slot
        p.append(float(amount)/totalUsage)
```

```

# e.g. totalUsage*0.13 (a.k.a. 13%)
minAmountInP.append((self.minPercentMovable(p[-1])
                    * totalUsage)/100)
# e.g. totalUsage*0.70 (a.k.a. 70%)
maxAmountInP.append((self.maxPercentMovable(p[-1])
                    * totalUsage)/100)

orderHigh = orderByPriceHighLow(aUsageProf, costProfile)
orderLow = [x for x in orderHigh]
orderLow.reverse()

for a in range(len(orderHigh)-1):
    # try to move your load to the cheapest ts
    dearLocalPos = orderHigh[a][1]
    dearTS = orderHigh[a][0]
    amountMoveablefromDear = pAmount[dearLocalPos] -
                            minAmountInP[dearLocalPos]

    for b in range(len(orderLow)-(a+1)):
        cheapLocalPos = orderLow[b][1]
        cheapTS = orderLow[b][0]
        if( not self.checkInertia(dearTS, cheapTS) ):
            break # failed inertia constraint

    # see how much load can be given,
    # compared to how much load which can be taken
    amountMoveabletoCheap = pAmount[cheapLocalPos] -
                            maxAmountInP[cheapLocalPos]

```

```

amountToMove = min(amountMoveablefromDear ,
                    amountMoveabletoCheap)
percentToMove = amountToMove/totalUsage

if(self.n<=percentToMove and percentToMove <= p[dearLocalPos]):
    # if Amount is big enough to bother move
    # local and global representations of load usage are moved

    # Reduce amount available form expensive TS
    pAmount[dearLocalPos] -= amountToMove
    aUsageProf[dearLocalPos] = (aUsageProf[dearLocalPos][0] -
                                amountToMove,
                                aUsageProf[dearLocalPos][1])
    p[dearLocalPos] = (pAmount[dearLocalPos]/
                       totalUsage)

    # Add amount available to cheap TS
    pAmount[cheapLocalPos] += amountToMove
    aUsageProf[cheapLocalPos] = (aUsageProf[cheapLocalPos][0] +
                                amountToMove,
                                aUsageProf[cheapLocalPos][1])
    p[cheapLocalPos] = (pAmount[cheapLocalPos]/totalUsage)
return aUsageProf

```

Appendix B

Code

“OptimiseUsage” Dependencies

```
def minPercentMovable(self, p):
    minPercentMovable1 = self.n
    minPercentMovable2 = p/(1+self.e)
    if(minPercentMovable1 > minPercentMovable2):
        return minPercentMovable1
    else:
        return minPercentMovable2

def maxPercentMovable(self, p):
    maxPercentMovable1 = 1
    maxPercentMovable2 = (1+self.e)*p
    if(maxPercentMovable1 < maxPercentMovable2):
        return maxPercentMovable1
    else:
        return maxPercentMovable2
```

```
def checkInertia(self, oldTS, newTS):  
    return ( self.loadProfile[oldTS] >=  
             ( self.loadProfile[newTS] * self.ePlus1Cubed) )
```