

UNIVERSITY OF AUCKLAND
DEPARTMENT OF ENGINEERING SCIENCE

Global Optimisation Carbon Pricing Initiative (GOCPI)

Author: Connor McDowall
Supervisor: Rosalind Archer

October 13, 2020

Completed Declaration of Contribution Form

This form is to accompany

Abstract

Acknowledgements

Contents

1	Introduction	9
2	Literature Review	9
2.1	An Introduction to Energy	9
2.2	Energy, Emissions, and the Economy	9
2.2.1	Energy	9
2.2.2	Emissions	10
2.2.3	Economy	10
2.3	Current Political Action	11
2.3.1	Paris Agreement	11
2.3.2	Carbon Pricing Initiatives	12
2.4	Opportunities and Challenges	13
2.4.1	Market-Based Approaches	13
2.4.2	Benefits and Costs to a Carbon Tax	14
2.4.3	Decarbonization, Investment, and Technology	14
2.5	Energy Modeling	15
2.5.1	Global and Domestic Energy Scenarios	15
2.5.2	The Integrated MARKEL-EFORM System (TIMES) Model	16
2.5.3	General Algebraic Modeling System (GAMS)	16
2.5.4	Open Source Energy Modeling System (OseMOSYS)	16
3	Project Scope and Research Objectives	17
4	Methodology and Implementation	18
4.1	Project organisation	18
4.1.1	Version Control	18
4.1.2	Folder Structure	18
4.1.3	Python	19
4.1.4	Package Management	19
4.1.5	Excel	19
4.1.6	IBM ILOG CPLEX Optimization	20
4.1.7	IBM Watson Machine Learning Service	20
4.1.8	PyPI	21
4.1.9	Code Style	21
4.1.10	Infrastructure	21
4.2	Documentation	21
4.2.1	Project updates	21
4.2.2	Meeting minutes	22
4.3	Geographical Partitioning	22
4.4	Energy Modelling: TIMES	22
4.4.1	TIMES: The Integrated Markel-Eform System	22
4.5	Energy Modelling: OseMOSYS	25
4.5.1	Overview	25
4.5.2	Implementation	25
4.5.3	Integration	27
4.5.4	Storage	27
4.6	Building Energy Systems	28

4.6.1	Utopia Example	28
4.6.2	NZ Example	28
4.6.3	Energy System Functions	28
4.7	Standardised Forecasting Methodology: GOCPI Package Development . .	29
4.7.1	PyPI and Directory Structure	29
4.7.2	Navigation Module	30
4.7.3	EnergySystems Module	30
4.7.4	CreateCases Module	32
4.7.5	Forecasting Module	32
4.7.6	Optimisation Module	32
4.8	Interface	32
5	Results	33
6	Discussion	34
7	Future Work	35
7.1	Python Development	35
7.2	NZ Example Completion	35
7.3	Develop Simulation Processes	35
7.4	Develop Investment Strategies	35
8	Summary and Conclusion	36
9	Reflection	37
10	Appendices	38
10.1	Project Modules	38
10.2	OseMOSYS Model	38
11	References	39

Notion List

List of Figures

1	TIMES Implementation Process	23
2	VEDA Energy System Consistency Key	23
3	VEDA Energy System Base Year and Scenarios	24
4	VEDA Technology, Demand and Trade Scenarios	24
5	OseMOSYS Objective Function	27
6	GOCPI File Structure	29

List of Tables

1	OseMOSYS Components	26
---	-------------------------------	----

1 Introduction

2 Literature Review

2.1 An Introduction to Energy

Energy powers the world we live in. Industrial sectors use energy to power machining and production processes. Trade uses oil-based products to fuel shipping and logistics networks. The technology sector uses energy to supply power to data centres to facilitate internet services. The manufacturing sector use the chemical properties of oil and natural gas to produce glass, synthetic, and explosive products. In summary, energy has a role in every industry. Energy comes in many forms. The International Energy Agency (IEA) aggregates energy products into six key fuels: Coal, Gas, Oil, Renewables, Electricity, and Nuclear [1]. Coal, Gas, and Oil are non-renewable fossil fuels with high energy intensities and carbon emissions. Nuclear energy is a non-renewable resource as the uranium used in production is finite. Renewables utilise technologies to produce energy from infinitely available wind, solar, geothermal and hydro resources. These technologies emit low to no carbon emissions. Regardless, energy will power our way of life.

2.2 Energy, Emissions, and the Economy

Academics spent decades hypothesizing causal relationships between economic growth, energy consumption, and carbon dioxide (CO_2) emissions. One study explored this hypothesis by applying panel unit root tests, panel co-integration methods, panel causality tests, Fully Modified Ordinary Least Squares (FMOLS), and Dynamic Ordinary Least Squares (DOLS) estimation methods [20]. Another study explored both linear and non-linear causality using Granger Causality and Vector Autoregressive models [11]. Although these modeling techniques are not explored further in this literature review or project, there results show a relationship between economic growth, energy consumption, and carbon dioxide (CO_2) emissions. Subsequently, we must improve our understanding of the three variables and how their relationship informs energy strategy.

2.2.1 Energy

Global energy consumption increased annually over the last decade. As outlined in BP's Statistical Review of World Energy, world consumption grew from 11705.1 Million Tonnes Oil Equivalent (Mtoe) in 2008 to 13864.9 Mtoe in 2018. World consumption grew 2.9% per annum in 2018 and 1.5% per annum for the period 2007-2017. In 2018, 85% of global energy consumption came from Oil, Natural Gas, and Coal. Global oil consumption increased by 1.2%, global coal by 1.4%, global gas by 5.3%, global nuclear by 2.4%, global hydro-electricity by 3.1%, and global renewables by 14.5% [39]. The 14.5% growth in renewables from 2017 to 2018 highlights how renewables are increasingly competitive as emphasised in the 2020 Deloitte US Renewable Energy Outlook. Flat electricity load growth, declining costs, and the maturation of energy storage all contribute to decreasing costs and increasing competitiveness. Decentralised energy networks with remote renewable micro-grids bolster resilience from increasingly frequent adverse weather events. Collaborative efforts lead to newly created efficiencies and the nurturing of new ideas, investment, and leadership. The combination of stakeholders' demand for increased resiliency in power networks, and collaborative efforts to foster innovation in renewables, will

continue to drive renewable uptake [47].

The fallout from the 2008 financial crisis led to a decrease in global consumption. The COVID-19 pandemic is ravaging global oil markets with an approximate 1550 Mtoe (33% [46]) decrease in consumption. The fall in demand from COVID-19, and the inability to store more oil, will catalyse significant change. Governments are distributing stimulus packages in response to the pandemic. Consumption is likely to bounce back after the fallout from COVID-19. Stimulus and the disruption to the energy industry may change the way energy is produced and consumed moving forward.

2.2.2 Emissions

Greenhouse gas emissions (GHG) are adverse by-products from the production or consumption of fossil fuels. Carbon dioxide (CO_2) and Nitrous Oxide (N_2O) originate from combusting fossil fuels. Methane (CH_4) is emitted from fossil fuel production processes. GHGs act as insulators, slowing the rate energy leaves the earth's atmosphere by absorbing energy. Global Warming Potential (GWP) measures how much energy one tonne of gas will absorb relative to one tonne of CO_2 . The GWP for CO_2 is 1 and remains in the atmosphere for thousands of years. The GWP for N_2O is 265-298 over 100 years. Lastly, the GWP for CH_4 is 28-36 over 100 years [3]. Carbon dioxide emissions from the consumption of oil, gas, and coal for combustion increased annually over the last decade. This subset of global emissions rose from 30336.7 Mtoe in 2008 to 33890.8 Mtoe in 2018, growing 11.72% over this period [39].

It is hard to argue anthropomorphic contributions to global emissions are not playing a role in climate change. The Intergovernment Panel on Climate Change (IPCC) are an intergovernment body of the United Nations (UN). The IPCC inform relevant parties on the scientific basis of risk-induced climate change. The IPCC also convey it's natural, economic, and political implications. The IPCC prepared a special report on Global Warming of 1.5 °C, based on an assessment of around 6,000 peer-reviewed publications. This report confirms climate change is affecting livelihoods and ecosystems worldwide. Approximately between 0.8 °C and 1.2 °C of global warming above pre-industrial levels is attributable to human activities. It is likely to reach 1.5°C between 2030 and 2052 if the current rate of increase continues. The warming from the pre-industrial era to the present will persist for centuries to millennia. Climate models are predicting increased mean temperatures in land and sea regions, hot extremes, heavy precipitation, drought, and precipitation deficits. Biodiversity and ecosystem impacts, such as species loss and extinction, will increase. Ocean temperatures, acidity, and oxygen scarcity will increase. Global warming will impose risks to health, livelihoods, food security, water supply, human security, and economic growth. All expected to increase unless severe action is taken [15]. Subsequently, emissions will continue to play an important role in our future.

2.2.3 Economy

The global economy grew substantially over the last century. Worldwide Gross Domestic Product (GDP) per capita (constant 2010 USD) grew from \$3746 in 1960 to \$10891 in 2018 [9]. Gross Domestic Product increased from \$66.1 trillion (Current USD) in 2010 to \$85.9 trillion in 2018 (Current USD). The World Bank segments GDP into four key seg-

ments: Agriculture, Industry, Manufacturing, and Services. Agriculture corresponds to International Standard Industrial Classification (ISIC) divisions 1-5 which cover forestry, fishing, cultivating crops, and livestock production. Industry corresponds to ISIC divisions 15-37 covering mining, manufacturing, construction, electricity, water, and gas. Manufacturing corresponds to ISIC divisions 10-45 covering businesses which physically or chemically transform materials of components into new products. Services corresponds to ISIC divisions 50-99 covering wholesale trade, retail trade, transport, and government, financial, professional, and personal services. In 2018, Agriculture, Industry, Manufacturing, and Services made up 3%, 25%, 16%, and 65% of GDP respectively [10]. On inspection, energy related products are key inputs to producing segment outputs. Traditionally, production is a function of capital and labour. Energy should be included in this set with theoretical and time series analysis supporting this claim. Empirical and theoretical evidence suggests energy availability, coupled with energy use and output, plays a key role in enabling growth [43]. Unfortunately, the COVID-19 pandemic drew to a close this period of unprecedented growth as the economy is forecast to plunge into a deep recession.

It is clear action must be taken to reduce anthropomorphic emissions while meeting the needs the economy after reviewing energy, emissions, and the economy.

2.3 Current Political Action

2.3.1 Paris Agreement

Over the last few decades, there has been debate on anthropomorphic emissions causing climate change. During this period, the UN began the United Nation's Climate Change Regime to ratify global agreements to combat climate change. There have been three major phases to this regime. The first phase was negotiating, adopting, and ratifying the United Nations Framework Convention on Climate Change from 1990 to 1995. The second phase was the adoption of the Kyoto Protocol on the 11th of December 1997. Unfortunately, the protocol took until the 16th of February 2005 to be ratified due to a complex ratification process. The Kyoto Protocol operationalized the United Nation's Framework Convention on Climate Change. Industrialised countries were committed to reduce and limit GHG emissions in accordance with targets agreed on an individual basis. The protocol only bound developed countries as they were deemed mostly responsible for the state of emissions. The protocol was split into two commitment periods. The first set binding emission targets for 36 industrialised countries, and the European Union, to reduce 5% of emissions compared to 1990 levels. The second commitment was set by the adoption of the Doha Amendment on the 8th of December 2012. This commitment was set to start 2013, end in 2020, and reduce emissions by 18% from 1990 levels. Unfortunately, the second commitment period was not ratified. The protocol established three flexible market mechanisms; International Emissions Trading, Clean Development Mechanisms, and Joint Implementation. The protocol held parties accountable by establishing rigorous monitoring, review, verification, and compliance systems [16].

This protocol was a step in the right direction but there were some inherent issues. A complex ratification process limited uptake. Participating countries had disproportionate obligations to reduce emissions. The commitment periods did not have a long term outlook, contained different subsets of participants, and were not global in nature.

There was another long term co-operative action promoted under the UNFCCC, launched

in the Bali Action Plan. Both the second commitment and the co-operative action were to conclude at the 2009 Copenhagen Conference. However, this conference ended in disappointment as there wasn't enough time before the conference to resolve the issues in the regime.

At the conference, the Copenhagen Accord, political in nature, was agreed upon on the final night by a group of states including most major economies. This accord established a bottom up architecture where both targets and actions were set individually then reported internationally. In 2010, the Cancun Agreements incorporated the Copenhagen Accord into the UNFCCC regime. However, all progress to date did not look pass 2020.

This issue was resolved at the 2011 Durban Conference when the Durban Platform for Enhanced Action launched the negotiations which lead to the Paris Agreement. Negotiations continued for years, addressing how to develop an instrument suitable for all parties, and requested the submission of nationally determined contributions (NDCs)[13].

The 2015 United Nations' Climate Change Conference was held in Paris. After some exceptional negotiating from French Foreign Minister Laurant Fabius, the Paris Agreement was adopted on the 12th of December 2015, and ratified on the 4th of November 2016.

The Paris Agreement's main objective is to strengthen the global response to climate change by keeping the temperature rise this century below 2° above pre-industrial levels. The agreement also pursues efforts to limit the increase to 1.5°. The agreement addresses the following across 29 articles: long term temperature goals, global peaking targets, mitigation efforts through nationally determined contributions (NDCs), sink and reservoir use, market/non-market approaches, adaptations, loss/damage mitigations from the adverse effects of climate change, financial support, technology support, capacity-building support, education, reporting systems, and global stocktake procedures [17], [36]. Article six of the Paris Agreement outlines mitigations and NDCs which include the following carbon pricing initiatives: Emissions Trading Schemes and Carbon Taxes.

2.3.2 Carbon Pricing Initiatives

Emissions Trading Systems (ETS) facilitate emission reductions where cheapest. Polluters who find it easier to reduce emissions can sell emission allowances to polluters who struggle to lower emissions. There are two types of ETS. Firstly, Cap and Trade. Secondly, Baseline and Credit Systems. Cap and Trade sets an upper limit on emissions with emission credits either grandfathered or auctioned. Most emission credits under this system are grandfathered with proceeds captured by existing polluters. Most cap and trade systems are localised regionally or nationally. However, the European Union (EU) implemented the EU ETS to trade across the EU. Baseline and credit systems don't have a fixed level of emissions. Polluters can reduce emissions to earn credits to sell to other polluters who need to meet regulations [8].

Carbon Taxes are either implicit or explicit. An implicit tax is incorporated into the price of a GHG intensive product e.g. retail fuels. An explicit tax is a price per quantity of emissions produced e.g. \$10 per tonne of CO_2 equivalent.

Carbon pricing initiatives are increasingly recognized as instrumental to cost-effectively deliver the transition to low-carbon societies. The Organisation for Economic Co-operation

and Development (OECD), International Monetary Fund (IMF) and the IPCC all recognise the need to strengthen these initiatives.

57 carbon pricing initiatives are either currently implemented, or scheduled for implementation, as at April 1st 2019. 28 ETFs are spread across national and subnational jurisdictions with 29 carbon taxes primarily implemented at a national level. These pricing initiatives only cover approximately 20% of Greenhouse Gas Emissions (11 GtCO₂e). There is variation in carbon prices from less than US\$1/tCO₂e to US\$127/tCO₂e. Unfortunately most carbon tax price increases are linked to inflation only. Governments raised more than US\$44 billion in carbon pricing revenue during 2018. There are determined price trajectories that can deliver on the Paris Agreement which increase each decade on a non-discounted basis. IPCC trajectories show the marginal cost of reducing GHG emissions. Other sources provide carbon price ranges which consider ambitious climate policy. When setting carbon prices, local, ethical and distribution factors must be considered.

Internal carbon pricing is increasing in popularity within the private sector. 1300 companies, including 100 Fortune 500 companies, disclosed the current use or intent to use internal carbon pricing. Traditionally, internal carbon pricing drove investment planning for mandatory carbon policies. Internal carbon pricing is transitioning towards informing long-term climate investment and risk strategies. Financial policy frameworks are reassessed to use carbon pricing to support sustainable growth. Internal carbon pricing is informing the construction of market indexes e.g. S&P Carbon Price Adjusted Index to inform market climate risk. Internal carbon pricing will continue to catalyse both the investment in sustainable technology and the divestment in fossil fuels. Although new carbon pricing initiatives are emerging with an increased emphasis on global collaboration, it is clear the global community is far from reaching the objectives set by the Paris Agreement. Only 5% of carbon pricing initiatives are currently priced to meet the Paris Agreement's temperature goals [27].

2.4 Opportunities and Challenges

2.4.1 Market-Based Approaches

Both carbon pricing initiatives are market-based and can be less obtrusive for industry than regulatory controls. The prevalence of emissions across global industries sees a market-based approach more equitable, evenly distributed across industries, and quicker to implement.

Arguably, a carbon tax is the most straightforward approach. A carbon tax could be imposed on all emitting inputs and outputs of production. The marginal cost of carbon would be covered by a tax rate. Tax credits from sequestration or other initiatives could be used to invest directly into alternative technologies. Tax rates could be adjusted to illicit the desired market response. A carbon tax would align with existing systems used to facilitate inland revenue functions in the desired jurisdiction.

The effectiveness of cap and trade systems is undermined by grandfathering credits, using offsets in lieu of meaningful emission reduction targets, and the challenges of setting baselines. If credits are grandfathered, there are no tax revenues to invest in alternative energy or technology. In a cap and trade system, there is uncertainty around the price of those reductions, and the subsequent effect these credits have on lowering emissions.

Cap and trade initiatives provide benefit certainty as you can ascertain the environmental benefits from an imposed ceiling on emissions. However, there is no cost certainty which is provided through a carbon tax.

2.4.2 Benefits and Costs to a Carbon Tax

A carbon tax is simpler as is set at a cost per tCO_2e produced. In contrast, a cap and trade is more complex due to negotiating baselines, grandfathering/auctioning processes, monitoring systems, international trading guidelines, and cost uncertainty prevention. A carbon tax more easily generates revenue to invest in reducing greenhouse gas emissions and supporting businesses adopting more sustainable processes. A cap and trade system generates revenues using credit auctioning but is less effective than a carbon tax if grandfathering credits exists. A carbon tax ensures cost certainty as the exact cost of emissions is quantified. In addition, there is a clear message to polluters with a carbon tax. Under a cap and trade system, there is more uncertainty around the impact polluters have as they can merely purchase more credits to increase emissions. In light of these positives, there are also negatives to a carbon tax. There is significant political opposition in proposing a new form of tax. The benefits of a carbon tax will need to be clearly communicated to all stakeholders. The benefits of a carbon tax are uncertain in how they enact reductions in emissions. Tax exemptions may reduce the effectiveness of the carbon tax if granted inequitably. For example, if a super major was given a significant tax exemption on the basis of an existing political relationship. A carbon tax is also difficult to co-ordinate with other participants [8]. In addition to these disadvantages, there are misconceptions leading to the opposition of carbon taxes. These misconceptions include, but are not limited to, the following: Taxes reducing welfare, and increasing unemployment, from lower levels of consumption and production. Taxes perceived to put incumbents' business models at risk. Subsequently, these incumbents lobby heavily prevent change. Taxes increasing the prices of goods and services consumers consume, creating opposing public opinions [44].

Addressing the disadvantages and misconceptions of a carbon tax would help implement carbon taxes across geographies to reduce greenhouse gas emissions.

2.4.3 Decarbonization, Investment, and Technology

The revenue generated from carbon taxes would need to be reinvested to benefit regional, national, and global communities. One opportunity is investing in sustainable technologies. Fortunately, this type of investment is increasing. Between 2010 and 2019, renewable technology drew \$2.6 trillion in investment. In 2019, \$282 billion of renewable capacity was financed worldwide. Wind technologies (onshore and offshore) and solar were financed \$138 billion and \$131 billion respectively. This financing success was attributable to falling costs and maturing technologies. Renewable technologies are now profitable. Costs associated with solar and wind technologies have fallen 85% and 49% respectively in the last 10 years [26]. In Australia, New Zealand, Canada, Europe, Japan, and the United States, sustainable investments reached assets of \$30.7 trillion in early 2018, one-third of total investment. By 2025, renewables will be competitive with natural gas. New technology is emerging in the Oil and Gas Industry to decarbonise the industry. These technologies include but are not limited to: renewable power sources, electrification, vapor recovery units, carbon capture, carbon storage, and green hydrogen

[12]. Carbon tax revenues could be used in the following ways: investment into sustainable technologies, replacing existing infrastructure to support sustainable technology growth, reinvestment into participating businesses, or addressing the adverse affects of climate change. It is important to communicate the benefit of sustainable investment to all relevant stakeholders.

2.5 Energy Modeling

It is important to understand how energy demand and supply is modeled before discussing emission levels, carbon taxes, and reinvestment opportunities. Unsurprisingly, energy modeling is complex.

2.5.1 Global and Domestic Energy Scenarios

Different scenarios define modeling processes. The World Energy Council devised three energy transition scenarios describing plausible pathways for the global energy transition to follow. The scenarios look forward to 2060. There is an inflection point in 2040 to assess the success of the strategies underpinning the scenarios. The council leveraged expert member communities and annual surveys to devise these strategies. The three global scenarios tell different narratives relating to the progression of global primary energy demand, electrification, mobility improvements, energy efficiency, infrastructure innovation, investment, new technologies, political action, and Paris Agreement alignment. There are three global scenarios: Modern Jazz, Unfinished Symphony, and Hard Rock. Modern Jazz is a market driven scenario. Unfinished Symphony is a highly collaborative, policy driven scenario. Hard Rock is a minimally collaborative, internal policy driven scenario [42].

Countries use the global scenarios to inform their regional and national scenarios. New Zealand has followed this methodology through devising the Tui and Kea scenarios. Tui follows the narrative of a global community effort. New Zealand does not generally have a common view on what is important. Subsequently, the country adopts a wait and see approach with some protection provided to local businesses. New Zealand will focus firstly on economic prosperity and individually wellbeing by leveraging off comparative advantages. This is purely a commercial response. Kea forecasts the New Zealand economy cannot remain internationally competitive under current emission intensity trends. The country will take leadership in lowering emissions, choosing to undergo an early and aggressive economic transformation. New Zealand will act before the global economy at the expense of its own.

Both scenarios are underpinned by 19 critical uncertainties. These vary from external sources such as global stability, international fuel markets, urban sustainability, energy affordability, and the allocation of natural resources. Both scenarios consider different service demands e.g. Number of km travelled, population, GDP (\$), forecast carbon prices (\$/tCO₂e), carbon emissions (Mt/p.a), required investment (\$), and commodity prices [18], [19]. These scenarios were feed into The Integrated MARKEL-EFORM System (TIMES) model to forecast their impact.

2.5.2 The Integrated MARKEL-EFORM System (TIMES) Model

The Integrated MARKEL-EFORM System (TIMES) model generator was developed as a component of the International Energy Authority's (IEA) Energy Technology Systems Analysis Program (ETSAP). The IEA-ETSAP uses long term energy scenarios to conduct comprehensive environment and energy analyses [33]. Energy is modeled through combining complementary technical engineering and economic approaches. TIMES uses a technology rich bottom up architecture, using linear programming to produce least cost energy systems for medium to long term time horizons [34]. The TIMES model encompasses each step in the value chain to produce and supply energy to meet the demand for energy services by consumers. These include: primary resources, transformation processes, transportation methods and conversion processes. The supply side considers production methods and net exports. Energy is carried through to residential, commercial, agricultural, transport, and industrial sectors. The relationships between producers and consumers underpin the TIMES model. The nature of these relationships are mathematical, economic, and technological. TIMES considers technologies, commodities, flows, and scenarios constrained by policy decisions. Services demanded by consumers are the main inputs for the model. The model will make investment, supply, trade, and operating decisions when considering the inputs, constraints, and scenarios in the model. Consumer and producer surpluses are maximized from a mathematical perspective. The main outputs are energy system configurations at the lowest cost that meet end users' service demands. These include energy prices, flows, emission quantities, capacities, and costs [34], [22].

The TIMES model is useful for using energy scenarios to devise strategies for implementing carbon taxes.

2.5.3 General Algebraic Modeling System (GAMS)

GAMS is a high-level modeling system for optimization and mathematical programming. The system is suitable for large scale, complex modeling applications. The system gives you access to a diverse portfolio of solvers to solve linear, non-linear, and mixed integer optimization problems. GAMS is the ideal tool for modeling energy and carbon tax scenarios using TIMES Models [23].

2.5.4 Open Source Energy Modeling System (OseMOSYS)

The OseMOSYS project is an open source project to improve the accessibility of modeling energy systems. The modeling methodology is scalable from city to continental granularities. The approach is designed to require no upfront investment, little time commitment and provides a fast learning curve. This methodology is suitable for developer, modelers, academics and policy makers. The model structure includes pre-defined sets, parameters, objective function and constraints relating to energy systems. Python, GNU MathProg and GAMS versions are available for download in remote repositories on Github [38].

3 Project Scope and Research Objectives

The literature led to some interesting outcomes. Carbon taxes are an effective way to help meet the goals set by the Paris Agreement, reduce global emissions, and reinvest in both sustainable and decarbonizing technologies. However, there are significant obstacles stopping the implementation of carbon taxes. These obstacles include political opposition, effectiveness in reducing emissions, co-ordination with other local, regional, national and international carbon tax efforts, and the distribution of tax exemptions. Currently, the distribution and implementation of carbon pricing initiatives is inequitable as major polluters aren't enforcing these initiatives e.g the majority of the US and India. Producers who operate globally may pivot to produce and sell GHG intensive products elsewhere in the world if taxes are not correctly set and spread amongst geographies. In this context, geographies are defined as geographical areas where energy consumption and emission production occurs. Earth is the entire set of geographies. This set can be partitioned into continental, national, provincial, or regional geographies (subsets).

Any carbon pricing initiative must consider the specific combination of energy factors for the associated geography. The aggregate outcome from all geographies should meet the objectives set by the Paris Agreement. The revenues generated should be distributed back to participants with commercial benefits communicated through key financial metrics relevant to that geography or business.

The overall research objective is:

Develop a Global Carbon Pricing Optimisation Model.

The model will be developed by the following process:

1. Understand the existing approach to forecasting services demand and develop a standardised forecasting process which aligns with emission targets
2. Partition the global set of geographies into subsets with varying levels of granularity e.g. a continental subset with seven geographies (continents), a national subset with 195 geographies (sovereign states according to the United Nations) etc
3. Forecast services demand for each geography in each subset
4. Adapt an existing TIMES/OseMOSYS model to input standardised energy system parameters
5. Run simulations using the adapted model to form distributions on model outputs. These distributions are to inform the negotiation of carbon pricing initiatives
6. Develop and model investment strategies from subsequent key outputs including carbon prices
7. Develop a model interface (UI and UX) to communicate optimal global carbon prices, performance metrics, and the carbon tax benefits to support co-ordinated efforts to negotiate and implement carbon pricing initiatives

In conclusion, this project will attempt to address issues surrounding climate change. GOCPI will enable any user to design and model their own energy system to inform investment and policy decisions. The intention is to empower users to influence energy investment and policy decisions made by public and private parties.

4 Methodology and Implementation

4.1 Project organisation

GOCPI adopted Data Science best practice, as described by Wilson et al [49]. Although these practices are mostly reserved for data science projects, their principles are suitable for product development and version control. All data and results are saved regularly and reproducibly. The retention of data in all forms received high levels of attention. Project files were synched continuously to Google Drive [24]. Git [32] was used to manage version control for GOCPI’s source code, data, documentation and results. Git stores a complete history of versions using Git hashes. These hashes are strings unique to each state of the publicly available GOCPI repository¹. Git hashes enable the discretisation of GOCPI’s development over time, enabling the accessibility and recollection of all previous states given a unique git hash. This functionality enables reproducibility, error correction and the ability to revert to previous models.

4.1.1 Version Control

Git, hosted by GitHub, provides a comprehensive set of version control technologies. These technologies provide a range of benefits. Firstly, Git is excellent at providing and supporting collaborative functionalities. The master version of a project is accessible for all who have access to the repository. Each contributor can create custom copies of branches through pull requests on the master branch. Contributors can commit changes to custom branches and push these changes to the master branch through push requests. The product manager can review these push requests, approving suitable requests to integrate changes to the master branch. Collaborative efforts were possible with commit messages describing the contributions from each contributor. This project had one contributor. Git ensures the histories of code, work and authors are stored. The descriptive nature of the commit log ensures an accurate journal is kept.

4.1.2 Folder Structure

GOCPI maintained the file folder structure recommended in Wilson et al [49]. Project organisation was paramount as the modelling of energy systems involves integrating a range of optimisation models, data files and documents. Wilson et al’s recommendations were appropriate as data science projects require similar organisational rigor. Subsequently, file management and structure was most efficient and comprehensive. **GOCPI** is the root directory of this project and contains several sub directories: bin, data, doc, src and results. The **bin** subdirectory contains external scripts and compiled programmes related to the GOCPI project. The **data** subdirectory contains all raw data associated with the project. This data includes energy statistics, energy balance datasets, partitioned geographies, standardised optimisation models and TIMES modelling frameworks. The **doc** subdirectory stores GOCPI’s user guides, academic resources, research reports and project deliverables. The **results** subdirectory contains the output from optimisation simulations and processed data to display on dashboards and websites to inform investment and policy decisions. The **src** subdirectory stores the source code for preparing raw data, partitioning sets of geographies with varying granularities and the GOCPI

¹<https://github.com/CMCD1996/GOCPI>

python package available to download using PyPI² and install using pip³. All files were continuously backed up using Google Drive.

4.1.3 Python

Python 3.7 was the primary coding language for the GOCPI project. GOCPI's objective is to enable any user to design and model their own energy system to inform investment and policy decisions. The intention is to empower users to discuss energy investment and policy decisions made by public and private parties. Additionally, GOCPI intends to reduce misinformation regarding energy policies and help assess the feasibility of meeting the International Energy Agency's Sustainable Development Scenario [2]. Python is omnipresent, widespread in software development. Python's language design makes the language highly productive and simple to use. Python can hand off computationally straining tasks to C/C++ and has first-class integration capabilities with these two languages. The language also has a very active and supportive community [35]. In addition, Python is the most popular coding language on the planet defined by the PYPL Popularity of Programming Language Index. As at August 2020, Python had 31.59% of all language tutorial search instances on Google [41]. Python has many useful packages for creating the GOCPI package such as NumPy, Scikit-learn, os, csv and Pandas. Programming is quick due to Python's dynamic nature. The language is also open-source with no cost. Subsequently, the GOCPI model should be accessible for many users to use and extend.

4.1.4 Package Management

The Anaconda package management platform for Python [5] was the chosen coding environment. Anaconda is a well defined, free platform with known versions of python packages such as matplotlib, numpy and pip. The use of this environment ensures both reproducibility and consistency across infrastructure. These factors enable collaborative efforts. Although this project required no collaboration, the use of Anaconda will inform future developers on how to manage collaborative processes, especially for packages which are less well-maintained. Anaconda allows you to create custom environments which was necessary for creating scalable linear optimization problems to express energy systems. Pip is Python's default package manager and included within the Anaconda package. Pip was used to install and update packages for python not available on Anaconda such as twine and the custom GOCPI package developed for this project.

4.1.5 Excel

It is important users are comfortable with using the GOCPI model. Energy modelling can be quite complex. The modelling process must be transparent to inform users how to build their own models. Excel is ubiquitous across academic and professional communities. Excel's omnipotence makes the software well-suited for describing the components of the GNU Mathprog energy system model. The **GOCPI OseMOSYS Structure.xlsx** file describes the sets, parameters, constraints and objective function of a scalable energy system model. The User may toggle statements sets, parameters and constraints to change the complexity of the model. The model is imported to a text file using the

²<https://pypi.org/>

³<https://pypi.org/project/pip/>

CreateModelFile module in the **GOCPI** package. However, data related to these energy systems is stored using Python dictionaries, lists and NumPy arrays. Excel is best for two dimensional variables or data stored in Codd-Boyce relational databases [6]. Many of the parameters in energy systems modelling contain three or more dimensions. Therefore, Excel is not suitable to store these parameters. Python dictionaries, lists and NumPy arrays are preferred alternatives.

4.1.6 IBM ILOG CPLEX Optimization

The OseMOSYS methodology (see 4.5.1) translates energy systems into linear programming problems. A solver is required to optimise these user-defined energy systems. The IBM ILOG Optimization Studio [31], more commonly known as CPLEX, was chosen to be this solver. CPLEX solves very large linear programming problems using the Barrier Interior-point method [40] or primal/dual variants of the Simplex Method [14]. GOCPI's user-defined energy systems can be scaled up to model very large systems, creating large linear programming problems.

The IBM ILOG CPLEX Optimization Studio has an interface with the Python language based on a C programming interface. Subsequently, Python APIs are available to run the CPLEX solver when installed either locally or on a cloud service. The python packages are **cplex** and **docplex**. The **cplex** package contains classes for accessing CPLEX for the Python programming language. The **Cplex** class is the most important class in this package as provides methods for creating, modifying, querying, or solving optimisation problems. **Docplex** also enables the formulation of new linear programmes where one creates the model, defines the decision variables, sets the constraints and expresses the objective function. The user uses **docplex** to solve the linear programme on a local solver. Alternatively, the model can be solved on a private cloud using Decision Optimisation on Cloud service through the provision of a service url and personal API key. The CPLEX Python APIs are most attractive as provides the user with a powerful commercial solver in an accessible format.

There is a caveat to the use of the CPLEX solver. The IBM ILOG CPLEX Optimization Studio is commercial by nature and requires a license to use. Fortunately IBM have the IBM Academic Initiative [29], granting students access to commercial software for free. This commercial nature creates accessibility issues for users who are not enrolled at an academic institution or can afford to pay for the software. Accessibility issues caused by the need for commercial solvers must be addressed to enable the distribution of the GOCPI product.

4.1.7 IBM Watson Machine Learning Service

The IBM CPLEX Optimisation **Cplex** python API is suitable for smaller models that can be solved locally. As the model increases in complexity, the **docplex** Python API did enable the ability to solve larger linear programmes. Unfortunately, IBM have phased out the **docplex** Python API by incorporating Decision Optimisation on Cloud services into the IBM Watson Machine learning cloud services [30]. This change occurred very recently. This service uses IBM Cloud to access assets through credentials, create model deployments in IBM's servers and execute jobs to solve models. The model deployments must be Python-based models with jobs specifying a payloads containing input data.

4.1.8 PyPI

PyPI¹ is the Python Package Index, a repository of software for the python programming language. This repository helps you find and install software developed by the Python community who have decided to share their work. The GOCPI package is distributed from this platform to enable as many as possible the ability to model their own energy systems to inform and question energy policy and investment. Enter command: **pip install GOCPI** in the terminal to install the package using pip package management software.

4.1.9 Code Style

The GOCPI project is developed as the GOCPI package where all development code is organised. The PEP8 style for Python Code is the formatting style for development code [28]. All code is formatted with **yapf**, a formatter maintained by Google to format created Python files. Standardised formatting is important as makes the code easy to read, helps optimise the code and promotes consistency. Docstrings and commenting were most important in documentation. A docstring is a Python inline comment. Each class and function has an unique docstring, a one sentence description of the function, inputs with data types and types of outputs. The Google style docstring was most appropriate because of it's readability, ease to write and consistency with the Google Style Guide. Additionally, automated documentation generators (**pdoc3**, **Sphinx** etc.) can parse this format to create documentation. This self-consistent code style facilitated best practise maintenance and enables reproducibility.

4.1.10 Infrastructure

GOCPI creates scalable energy system optimisation models which change in complexity depending on there size. Computations either took place locally on a 128 GB, four core Apple MacBook Pro or remotely using a cloud service.

4.2 Documentation

The GOCPI project is well documented to keep an accurate record of key design decisions. The commit history described in 4.1.1 was the most important form of document. Other explicit documentation methods were applied to supplement this commit history. These methods, in addition to in-code documentation, include project updates and meeting minutes nested with a project logbook.

4.2.1 Project updates

Project updates were recorded as itemized lists. Each item is a brief description of the work completed during that day, week or month. Items include, but are not limited to, completing GOCPI submodules, researching energy system statistics, building websites or writing sections of this research report. These updates were pivotal to exploring new options, monitoring progress and making decisions to drive forward development. For example, the decision to adopt the OseMOSYS methodology in favour of the TIMES

¹<https://pypi.org/>

modelling methodology. Project updates were transcribed to the project logbook held in this project's research compendium.

4.2.2 Meeting minutes

Project meetings took place for half an hour once a week. These meetings included discussions on energy markets, modelling methodologies, project progress and key design decisions. The minutes from these meetings accompanied project updates in the project logbook nested within the research compendium.

4.3 Geographical Partitioning

The first stage of the GOCPI project was to come up with an exhaustive list of geographies. The user has the choice to either model the energy systems of these geographies individually or model a network of geographies connected by trade relationships. Geographies are grouped into three distinct sets. There is a continental set with six continents: Africa, North America, South America, Oceania, Asia and Africa. Antarctica was excluded from this set. The country set includes 194 United Nations member states. The elements in the city set number approximately 13800 unique cities from the 194 member states. The **GOCPI Geographies.gyp** script processes these unique sets to create a csv file detailing an aggregate geography set. This file contains 13800 geography combinations. Each combination includes a city's name, population, country and continent e.g. VANCOUVER,CANADA,2313328.0,NORTH AMERICA. A user may use this information to design energy systems for cities with similar populations, countries, continents. Cities may be combined to compare countries with similar populations or of residence within the same continent. The user may also build energy systems on a continent level. After the formation of an exhaustive list of geographies, the exploration and adaptation of energy system modelling methodologies could begin.

4.4 Energy Modelling: TIMES

The World Energy Council devised three energy scenarios to inform energy policy and investment. These are Hard Rock, Unfinished Symphony and Modern Jazz [42]. New Zealand devised two energy scenarios to inform national energy policy and investment. These are Tui and Kea [19]. The TIMES Modelling Methodology underpins the construction of these modelling techniques. Subsequently, I began the standardisation of a TIMES implementation to create a method to develop scalable energy systems.

4.4.1 TIMES: The Integrated Market-Eform System

An implementation of TIMES is divisible into a five stage process as displayed in Figure 1:



Figure 1: TIMES Implementation Process

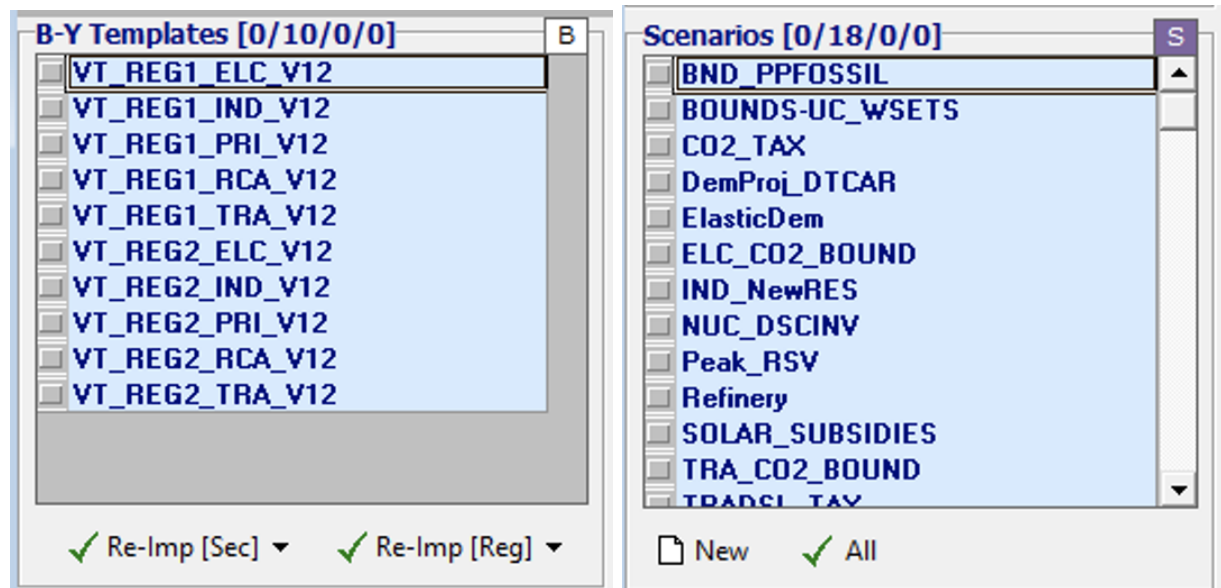
The VERSatile data analyst (VEDA) [21] is a software package required for TIMES modelling. VEDA is an interface to design energy systems from Excel-based inputs and create the TIMES data and model text files for optimisation using GAMS. The installation of VEDA FE and VEDA BE required a windows operating system. A virtual private network (VPN) and Microsoft Remote Desktop was used to access a windows operating system via a virtual machine. GAMS Studio was installed on both the local Macintosh operating system and remote Windows operating system to perform the optimisation formulated in GAMS. VEDA is proprietary and requires a commercial license to use. Fortunately, there is a 60 day trial version available for download and use. After installing VEDA, GAMS and Microsoft Excel, I began following a TIMES user manual written by the IEA to build a standardised TIMES model. The user manual started with a very basic energy system, incrementing on the energy model using twelve demo models. I downloaded these twelve demo models to edit their associated Excel files.

The first stage was to adapt the existing Excel files so the user can design their own reference energy systems and sets of energy scenarios. Data and assumptions for reference energy systems and scenarios are primarily recorded in several tables across separate excel sheets and files. I made copies of these excel files and began writing functions to change the data stored in these sheets for a user to create new energy systems.

The second stage was to design energy systems using the VEDA interface. Figures 3 and 4 show the VEDA interface. I used the VEDA interface to combine the Excel files I adapted from the VEDA demo models. Initially, I built the base year scenario as shown in figure 3a. The inclusion of different energy scenarios (figure 3b) led to a more robust energy system. Examples include introducing the price elasticity of demand, solar technology subsidies and carbon dioxide taxes. New resources, technologies, trade relationships and demand profiles were added as shown in subfigures 4c, 4b and 4a. VEDA informs the user if the energy system is consistent with all data and assumptions correctly defined by a key shown in figure 2.



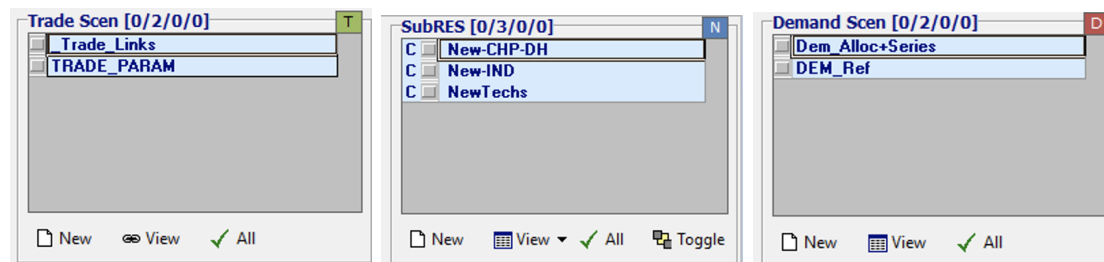
Figure 2: VEDA Energy System Consistency Key



(a) Energy System Base Year

(b) Energy Scenarios

Figure 3: VEDA Energy System Base Year and Scenarios



(a) Trade Relationships

(b) New Resources

(c) Demand Profiles

Figure 4: VEDA Technology, Demand and Trade Scenarios

After designing the energy system using the VEDA interface, a case manager is used to generate a TIMES text file and run the optimisation. The model file is written in GAMS and integrated into the VEDA interface. The case manager allows you to select different scenarios and produce new model files.

The third stage was to execute the GAMS optimisation and produce outputs to be fed into VEDA BE. The fourth stage is for VEDA BE to convert the GAMS output into Excel-based data files. In the fifth stage, data visualisation would enable the user to take the Excel outputs and make informed decisions relating to energy policy and investment.

Unfortunately, I faced serious limitations at stage 3 of the TIMES implementation. Several factors made the TIMES implementation difficult:

1. The remote nature of the project made the integration of software between Flex IT, local and remote difficult.
2. VEDA and GAMS Studio's commercial nature creates accessibility and equity issues for those who don't have the resources or connections to access the software. This

contradicts the project's purpose.

3. The complexity of TIMES inhibits reproducibility and creates barriers to entry for users not too familiar with energy modelling.

These reasons led to the abandonment of the TIMES implementation in favour of an alternative energy modelling methodology. An alternative methodology would trade the complexity and sophistication of TIMES modelling for usability and accessibility. After consultation with my project supervisor and one of New Zealand Energy Council modelers, the OseMOSYS methodology was recommended. Subsequently, I began developing an scalable open source energy system under this OseMOSYS methodology.

4.5 Energy Modelling: OseMOSYS

4.5.1 Overview

The OseMOSYS methodology was the next best alternative to TIMES for modelling large scale energy systems. TIMES can model useful complexities such as stochastic processes. However, OseMOSYS proved to be the better of the two methods as more closely aligned with the overall scope and purpose of the project. The aforementioned tradeoffs are justified given the usability and availability of OseMOSYS.

There is an online community who make contributions to the development of OseMOSYS. Their mission is to make energy modelling more transparent and accessible. This online community is developing three versions of OseMOSYS accompanied with exemplar energy systems. The resources created by this online community can be found on Github [48]. A version of OseMOSYS was available in GAMS, GNU Mathprog or Python (Pyomo). Each version was at different stages of development. The GNU Mathprog version was chosen as was one of the most complete of the three versions including two energy systems examples. The model and data files were downloaded as text files and stored in the project's working directory.

4.5.2 Implementation

It is not practical to explain the entire model given the scale of OseMOSYS. Therefore, the provision of a high level outline follows. Please refer to the research compendium for the complete model and comprehensive description. The OseMOSYS model contains 11 sets, 52 parameters, 67 variables, 1 objective function and 94 constraints. A summary of the model's components are as follows:

Component	Factors
Sets	Year, Technology, Timeslice, Fuel, Emission Mode of Operation, Region, Season, Day Type Daily Time Bracket, Storage
Parameters	Trade, Activities, Demand Profiles, Costs Storage Rates, Conversion Rates, Emission Limits etc
Variables	Demand, Storage Rates, Production Emissions Penalties, Operating Costs, Technology Use etc
Constraints	Energy Balances, Production, Charging Discharging, Storage Costs, Capacities and Investments Renewable Energy Production, Emissions Accounting etc

Table 1: OseMOSYS Components

The sets are the primary components to the model. Therefore, it is important they are explained to provide greater context.

1. **Year:** Represents the timeframe of the model containing all years to be considered.
2. **Technology:** Represent any element in the energy system that changes a commodity from one form to another, uses it or supplies it. The modeller is free to interpret the technology. It may be a single technology e.g. a power plant or an aggregation of many technologies e.g. millions of solar panels.
3. **Timeslice:** Represents the times resolution of the model through time splits.
4. **Fuel:** Represents any energy vector, proxies or energy service flowing through technologies. Depending on the required analysis, fuels may be individual groups, aggregate group or artificially separated.
5. **Emission:** Represents emissions potentially derived from operating technologies.
6. **Mode of Operation:** Represents the number of modes a technology may have. If a technology can have various inputs and output fuels, and can choose any linear combination of inputs and outputs, this may be expressed as a mode of operation. E.g. a CHP plant may produce heat in one mode and electricity in another.
7. **Region:** Represents the regions to modelled as explained in section 4.3. Supply-demand energy balances for all energy vectors, including trade with other regions, are ensured.
8. **Season:** Represents the number of seasons accounted for e.g. Summer, Autumn, Winter and Spring. These are expressed as successive integers e.g. Summer = 1, Autumn = 2.
9. **Day Type:** Represents the number of days types accounted for e.g. Weekend, Weekday. These are expressed as successive integers e.g. Weekend = 1, Weekday = 2.
10. **Daily Time Bracket:** Represents the number of splits for a day e.g. night, morning, afternoon, evening. These are expressed as successive integers e.g. night = 1, morning = 2, afternoon = 3, evening = 4.

11. **Storage:** Represents storage facilities/capabilities used in the model.

$$\text{Min: } \sum_r \sum_y \text{TotalDiscountedCost}_{r,y}$$

Figure 5: OseMOSYS Objective Function

cast horizon (Figure 5). Total discounted cost is the sum of the total discounted costs by technology over all the technologies.

Combinations of sets are used to define the parameters and variables. These parameters and variables are used to set constraints. The objective function uses the constraints to find the optimal variables. The objective function is the minimisation of discounted investment costs in each region across all years specified in the fore-

4.5.3 Integration

An lp-format file was required to solve OseMOSYS implementations with a solver. These files are written in GNU Mathprog, a language intended for describing linear mathematical programming models [45]. GNU Mathprog is a subset of the Algebraic Mathematical Programming Language (AMPL) [4] and consists of user-defined sets of statements and data blocks. The custom anaconda environment (**osemosys**) was created to facilitate this model formulation. This environment contains the libcx package which provides a standard library of C++ functionalities. The GNU Linear Programming Kit (GLPK) sits within this library. GLPK uses GNU Mathprog to solve large-scale linear programming (LP), mixed integer programming (MIP), and other related problems. GLPK is called using the **glpsol** command in the osemosys conda environment:

```
glpsol -m model.txt -d data.txt -wlp GOCPI.lp
```

The aforementioned glpsol command converts a data and model text file to an lp-file. The command also solves the problem if the energy system is not too complex. It is likely glpsol would not be able to solve larger scale energy system therefore a commercial solver would be needed. The installation of the GLPK was tested using the Utopian example data and model files using the aforementioned glpsol function. The necessary lp-format file was generated for the users chosen solver. This was paramount before disembarking on developing a standardised energy systems development method to generate both text and data files from user-defined energy systems.

4.5.4 Storage

Energy modelling and OseMOSYS can be quite complex. It is important this complexity is conveyed to the user so they understand how to build their own energy systems. Subsequently, an Excel-based template was created to describe and store all model components. The **GOCPI-OseMOSYS-Structure.xlsx** contains all the sets, parameters, variables and constraints needed for an OseMOSYS model. The spreadsheet also includes descriptions of all model components and the ability to toggle their inclusion in the user defined model. The user may customise the structure of their energy model by making adjustments to the model's components in the GOCPI-OseMOSYS-Structure Excel sheet. Constraints may be added or removed and the objective function changed to consider other factors e.g. the minimise emissions penalties or maximise the uptake of renewable technologies. The base model came from the model text file, written in GNU

Mathprog, stored in the OseMOSYS Github Repository. The template was adjusted to ensure constraint, variable and parameter indices were consistent with the demo data files.

4.6 Building Energy Systems

After the correct installation and initialisation of the OseMOSYS structure, it was appropriate to begin developing a methodology for building energy systems. The first step in this process was to understand the components and structure of the Utopia.txt example available for distribution from the OseMOSYS Github repository.

4.6.1 Utopia Example

4.6.2 NZ Example

After understanding the structure of the Utopian example

4.6.3 Energy System Functions

Many functions were to enable scalable energy system modelling. The most important functions are addressed in this section. The function **def set_discount_rate** calculates the discount rate for each region. This function is most important as the discount rate underpins the total value of discounted cash flows and the objective function. The discount rate is calculated using the weighted average cost of capital (WACC) method. The WACC, cost of equity, cost of preference equity and pre-tax cost of debt are calculated as follows:

$$WACC = \frac{D}{E + D + K} \times r_d \times (1 - t) + \frac{E}{D + E + K} \times r_e + \frac{K}{D + E + K} \times r_k \quad (1)$$

$$r_e = r_f + \beta \times (r_m - r_f) \quad (2)$$

$$r_d = \frac{i}{D} \quad (3)$$

$$r_k = \frac{Div}{MV_{ps}} \quad (4)$$

D = Book value of debt (\$m)

r_f = Risk free rates (%)

K = Market value of preferred equity (\$m)

r_d = Pre-tax cost of debt (%)

r_e = Cost of ordinary equity (%)

Div = Preference Dividends (\$)

MV_{ps} = Market value of preference shares (\$)

r_k = Cost of preferred equity (%)

t = Effective tax rate (%)

i = Cost of borrowings (%)

β = Market risk co-efficient (%)

r_m = Market return (%)

E = Market value of equity (\$m)

The regions used in the energy system example are New Zealand and Australia. Both are public entities in this case. The model may be used for both private and public entities. It is assumed the financial performance and position for each region, reported in the the New Zealand and Australian Government's financial reports, are suitable for providing

the financials necessary for calculating discount rates. Cost of borrowings (i), Equity (E), Debt (D) and Preference Equity (K) are reported in each region's annual financial reports (New Zealand [50] and Australia [25]). It is assumed both have no preference shareholders, do not distribute preference dividends (Div), list preference shares (MV_{ps}) or have preference equity (K). The effective tax rates (t) are each region's company tax rate. This example assumes New Zealand's market value of equity (E) is zero as not publically traded and Australia's negative net work implies market equity (E) is zero. Risk free rates (r_f) were derived from average 10 Year Government bond yields using data from each region's respective Reserve Bank (Reserve Bank of New Zealand [37] and Reserve Bank of Australia [7]). The monthly market returns for each region are annualised to calculate the market return (r_m). Each regions has a market index which serves as a proxy for the market. New Zealand has the NZX 50 and Australia has the ASX 200 [**Insert NZX50 and ASX200 references later**]. Monthly returns are annualised as follows:

$$\text{Annualised Monthly Return} = \left(\left(1 + \frac{\text{Index}_{\text{Ending}} - \text{Index}_{\text{Beginning}}}{\text{Index}_{\text{Beginning}}} \right)^{\frac{12}{\text{Number of Months}}} \right) - 1 \quad (5)$$

The cost of ordinary equity is calculated using the Capital Assets Pricing Model (CAPM). Both New Zealand and Australia are assumed to have market risk co-efficients (β) of zero as their fiscal policy is market independent. This methodology is appropriate for both public and private entities.

4.7 Standardised Forecasting Methodology: GOCPI Package Development

4.7.1 PyPI and Directory Structure

The aforementioned distribution method PyPI (section 4.1.8) was most appropriate. The GOCPI project is listed as an active project.

The package requires a specific directory structure to publish new distributions. The **GOCPI** stores several files required for the package: A sub directory of the same name containing an initialisation file (**init.py**) and package modules. A distributions sub directory (**dist**) for new package versions. A python file for setup information (**setup.py**) and a setup file for package configuration (**setup.cfg**). Lastly, there is a sub directory storing text files related to dependencies, package information and source links.

New distributions are stored in the **dist** displayed in figure 6. **Twine** is a package to upload new setup descriptions, configurations and code changes via

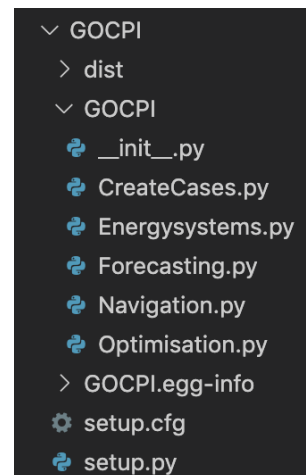


Figure 6: GOCPI File Structure

the local terminal. PyPI enables the immediate release and availability of the GOCPI package. Users can continue to use the product as new versions are released.

The development of the GOCPI package used the following process:

1. Make additions, deletions or increments to package modules.
2. Update the initialisation file to reflect additions, deletions or increments.
3. Update package information in the **setup.py** file and increment package version number.
4. Create a new distribution (tar.gz file) using the following command: **python setup.py sdist**.
5. Upload the new distribution using the following command: **twine upload dist/GOCPI-X.tar.gz** where X is the version number.
6. Download the new distribution using the following command: **pip install --upgrade GOCPI**.

Over 80 distributions were released during this prototype's development. Five models were developed to support the construction, forecasting and optimisation of user defined energy systems. These will be addressed in turn as contribute to the key functionalities of the package. The docstrings related to all written functions may be found in the appendix.

4.7.2 Navigation Module

The navigation module is a class to help the user navigate their local directory. The most important function in this class is the **Find_File** function. The purpose of Find_File() is to enable the user to seek and find designated files in their modelling directory from a user defined directory root. This root may be any node in the directory structure. The FindFile function is important for the other functions in this package.

4.7.3 EnergySystems Module

The EnergySystems module is a class to load, create model text files and create data text files from input user defined energy systems. There are three core modules to this class. **load_datacase()**, **create_model_file()** and **create_data_file()**. The EnergySystems class is initialized by taking the 11 sets, defined as lists, outlined in section 4.5.2 as inputs. The lengths of these sets are calculated using python's numpy module and used to initialise all 52 parameters related to an OseMOSYS energy model.

The load_datacase() function will load a user built energy system as outlined in section 4.6. The function simply replaces the EnergySystems initial sets and parameters with the ones defined by the user. The create_model_file() reads the GOCPI_OseMOSYS_Structure.xlsx explained in section 4.5.4 using the pandas module and writes the generated dataframe to a text file in the correct structure.

The create_data_file() function is the most extensive of the three functions in this class. Simplistically, the function write the python arrays and matrices for each set and parameter to one text file. Each set and parameter must have the correct syntax and structure

to create a compatible data file written as a text file. Effectively, user-defined python-based multi-dimensional lists/arrays had to be converted into GNU Mathprog-structure data blocks. This GNU Mathprog conversion had to occur for all 11 sets and 52 parameters. Sets and parameters varied in size, ranging from one dimensional arrays/lists to five dimensional matrices. The variability of size and need for correct naming conventions created a considerable design challenge. This solution had to change based on the users inputs while still generating the correct GNU Mathprog structure. The solution was to design a conversion block for each set and parameter. Examples of set and conversion block are shown in listings 1 and 2.

```

1  # year
2  set_string = ' '.join(self.year)
3  f.write('set YEAR\t:=\t{0};\n'.format(set_string))

```

Listing 1: One Dimensional Set Python to GNU Mathprog Conversion Block

```

1  # DaySplit = np.zeros((llh,ly))
2  param = 'DaySplit'
3  f.write('#\n')
4  columns = self.year
5  column_string = ' '.join(columns)
6  # Writes index specific parameter values to the text files
7  if toggle_defaults[param] == True:
8      f.write("param\t{0}\tdefault\t{1}:\t{2}:=\n".format(
9          param, defaults_dictionary[param], column_string))
10     # Converts maxtrix rows to list
11     array = np.array(self.dailytimebracket)
12     array = array.T
13     lt = array.tolist()
14     # Creates 2D matrix for this value
15     mat = self.DaySplit[:, :]
16     # Converts combined matrix to list and combines lists
17     matlist = mat.tolist()
18     #Combines the two lists
19     combined_list = list(zip(lt, matlist))
20     # Writes index specific parameter values to the text files
21     f.write("param\t{0}\t{1}:=\n".format(param, column_string))
22     for line in combined_list:
23         combinedflat = ''.join(str(line))
24         combinedflat = combinedflat.replace('[', '')
25         combinedflat = combinedflat.replace(']', '')
26         combinedflat = combinedflat.replace("'", '')
27         combinedflat = combinedflat.replace(",", '')
28         combinedflat = combinedflat.replace("(", '')
29         combinedflat = combinedflat.replace(")", '')
30         f.write("{0}\n".format(combinedflat))
31     else:
32         f.write("param\t{0}\tdefault\t{1}:=\n".format(
33             param, defaults_dictionary[param]))
34     f.write('; \n')
35

```

Listing 2: Two Dimensional Parameter Python to GNU Mathprog Conversion Block

Refer to the Research Compendium for the complete set of 63 Python to GNU Mathprog Conversion Blocks in the create_data_file() function.

4.7.4 CreateCases Module

4.7.5 Forecasting Module

4.7.6 Optimisation Module

4.8 Interface

Website available [here](#).

5 Results

6 Discussion

7 Future Work

7.1 Python Development

7.2 NZ Example Completion

7.3 Develop Simulation Processes

7.4 Develop Investment Strategies

8 Summary and Conclusion

9 Reflection

10 Appendices

10.1 Project Modules

10.2 OseMOSYS Model

The Open Source Energy Modelling System is a linear programme with sets, parameters, an objective function and constraints related to energy systems. The model is described below. Please refer to GOCPI OseMOSYS Structure.xlsx in the GitHub Repository for full descriptions of the sets, parameters and constraints.

11 References

References

- [1] International Electricity Agency. *Fuels and Technologies*. Available at <https://www.iea.org/fuels-and-technologies> (2020/04/21).
- [2] International Energy Agency. *World Energy Model Documentation*. Tech. rep. 2019.
- [3] United States Environmental Protection Agency. *Overview of Greenhouse Gas Emissions*. Available at <https://www.epa.gov/ghgemissions/overview-greenhouse-gases> (2020/04/21).
- [4] AMPL Optimization Inc. *AMPL*. Aug. 8, 2020. URL: <https://ampl.com/>.
- [5] Anaconda, Inc. *Anaconda*. Version 2020.02. Aug. 8, 2020. URL: <https://www.anaconda.com/>.
- [6] Marcelo Arenas. “Boyce-Codd Normal Form”. In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 264–265. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_1245. URL: https://doi.org/10.1007/978-0-387-39940-9_1245.
- [7] Reserve Bank of Australia. *Zero Coupon Interest Rates - Analytical Series - 2009 to Current*. Available at <https://www.rba.gov.au/statistics/tables/> (2020/08/15).
- [8] Reven S. Avi-Yonah and David M. Uhlmann. “Combating Global Climate Change: Why a Carbon Tax Is a Better Response to Global Warming Than Cap and Trade”. In: *Stanford Environmental Law Journal* 28.3 (2009).
- [9] The World Bank. *The World Bank Data*. Available at <https://data.worldbank.org/indicator/NY.GDP.PCAP.KD> (2020/04/21).
- [10] The World Bank. *World Bank Indicators: Structure of Output*. Available at <http://wdi.worldbank.org/table> (2020/04/23).
- [11] Marco R. Barassi and Nicola Spagnolo. “Linear and Non-linear Causality between CO2 Emissions and Economic Growth”. In: *The Energy Journal* 33.3 (2012), pp. 23–38.
- [12] Chantel Beck et al. “The Future is Now: How Oil and Gas Companies can Decarbonise”. McKinsey and Company: Oil and Gas Analysis. 2020.
- [13] Daniel Bodansky. “The Paris Climate Change Agreement: A New Hope?” In: *American Journal of International Law* 110.2 (2016), pp. 288–319.
- [14] Richard Bronson and Gabriel B. Costa. “4 - An Introduction to Optimization”. In: *Matrix Methods (Third Edition)*. Ed. by Richard Bronson and Gabriel B. Costa. Third Edition. Boston: Academic Press, 2009, pp. 127–148. ISBN: 978-0-12-374427-2. DOI: <https://doi.org/10.1016/B978-0-08-092225-6.50010-3>. URL: <http://www.sciencedirect.com/science/article/pii/B9780080922256500103>.
- [15] Intergovernmental Panel on Climate Change. *Global Warming of 1.5 C*. Tech. rep.
- [16] United Nations Framework Convention on Climate Change. *What is the Kyoto Protocol?* Available at https://unfccc.int/kyoto_protocol (2020/04/23).
- [17] United Nations Framework Convention on Climate Change. *What is the Paris Agreement?* Available at <https://unfccc.int/process-and-meetings/the-paris-agreement/what-is-the-paris-agreement> (2020/04/23).
- [18] BusinessNZ Energy Council. *New Zealand Energy Scenarios - Navigating Energy Futures to 2050*. Tech. rep. 2015.

- [19] BusinessNZ Energy Council. *New Zealand Energy Scenarios - Navigating Energy Futures to 2060*. Tech. rep. 2019.
- [20] Chaido Dritsaki and Melina Dritsaki. “Causal Relationship between Energy Consumption, Economic Growth and CO2 Emissions: A Dynamic Panel Data Approach”. In: *International Journal of Energy Economics and Policy* 4.2 (2014), pp. 125–136.
- [21] Kanors EMR. *VErsatile Data Analyst*. Available at <https://www.kanors-emr.org/> (2020/10/03).
- [22] ETSAP. *Overview of TIMES Modelling Tool*. Available at <https://iea-etsap.org/index.php/etsap-tools/model-generators/times> (2020/04/26).
- [23] GAMS. *An Introduction to GAMS*. Available at <https://www.gams.com/products/introduction/> (2020/04/23).
- [24] Google LLC. *Google Drive*. Aug. 5, 2020. URL: <https://www.google.com/drive/>.
- [25] The Australian Government. *Financial Statements of the Government of New Zealand for the year ended 30 June 2019*. <https://www.finance.gov.au/sites/default/files/2019-12/consolidated-financial-statements-201819.pdf>. Dec. 2019.
- [26] Bloomberg Green. *Renewable Investment*. Available at <https://www.bloomberg.com/graphics/climate-change-data-green/investment.html> (2020/04/23).
- [27] World Bank Group. *State and Trends of Carbon Pricing 2019*. Tech. rep. 2019.
- [28] Barry Warsaw Guido Van Rossum and Nick Coghlan. “PEP 8: Style Guide for Python Code”. Python. org, 1565. 2001.
- [29] IBM. *IBM Academic Initiative*). Available at <https://www.ibm.com/academic/home> (2020/08/09).
- [30] IBM. *IBM Watson Machine Learning*. Available at <https://www.ibm.com/cloud/machine-learning> (2020/09/23).
- [31] International Business Machines Corporation. *IBM ILOG CPLEX Optimization Studio*. Version 12.10. Aug. 9, 2020. URL: <https://www.ibm.com/nz-en/products/ilog-cplex-optimization-studio>.
- [32] Linus Torvalds. *Git*. Version 2.28.0. Aug. 5, 2020. URL: <https://git-scm.com/>.
- [33] Richard Loulou, Gary Goldstein, and Ken Noble. “Documentation for the MARKEL Family of Models”. In: (2004).
- [34] Richard Loulou et al. “Documentation for the TIMES Model - PART I 1–78”. In: (2005).
- [35] Medium. *Top 10 In-Demand programming languages to learn in 2020*. Available at <https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020-4462eb7d8d3e> (2020/08/06).
- [36] United Nations. *Paris Agreement*. Tech. rep. 2015.
- [37] Reserve Bank of New Zealand. *Wholesale Interest Rates -B2*. Available at <https://www.rbnz.govt.nz/statistics/b2> (2020/08/15).
- [38] OSeMOSYS. *Open Source Energy Modelling System*. Available at <http://www.osemosys.org/> (2020/08/02).
- [39] BP plc. *BP Statistical Review of World Energy*. Tech. rep. 2019.
- [40] Florian A. Potra and Stephen J. Wright. “Interior-point methods”. In: *Journal of Computational and Applied Mathematics* 124.1 (2000). Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations, pp. 281–302. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/S0377-0427\(00\)00433-7](https://doi.org/10.1016/S0377-0427(00)00433-7). URL: <http://www.sciencedirect.com/science/article/pii/S0377042700004337>.

- [41] PYPL. *PYPL PopularitY of Programming Language*. Available at <http://pypl.github.io/PYPL.html> (2020/08/06).
- [42] World Energy Scenarios. *World Energy Scenarios 2019*. Tech. rep. 2019.
- [43] David I. Stern. “The Role of Energy in Economic Growth”. In: *SSRN Electronic Journal* (2010).
- [44] T Sterner and G Kohlin. “Pricing Carbon: The Challenges”. PhD thesis. 2015.
- [45] Massachusetts Institute of Technology. *GNU Mathprog*. Available at http://web.mit.edu/lpsolve_v5525/doc/MathProg.htm (2020/08/08).
- [46] Financial Times. *US Oil Price Back Below Zero After Historic Plunge*. Available at <https://www.ft.com/content/26ea5ef9-0619-4e50-b605-58e36d3fc4d9> (2020/04/21).
- [47] Deloitte US. *2020 Renewable Energy Industry Outlook*. Tech. rep. 2019.
- [48] Various. *OseMOSYS Github Repository*. Available at <https://github.com/OSeMOSYS/OSeMOSYS> (2020/10/10).
- [49] Greg Wilson et al. “Good Enough Practices in Scientific Computing”. In: *PLOS Computational Biology* 13 (Aug. 2016). DOI: 10.1371/journal.pcbi.1005510.
- [50] Treasury New Zealand. *Financial Statements of the Government of New Zealand for the year ended 30 June 2019*. <https://treasury.govt.nz/sites/default/files/2019-10/fsgnz-2019.pdf>. June 2019.