Discrete Optimization of Upgrade Scheduling

by

Derick Smith

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Science in Modelling and Computational Science

Faculty of Science
University of Ontario Institute of Technology (Ontario Tech University)
Oshawa, Ontario, Canada
August 2022

Thesis Examination Information

Submitted by: **Derick Smith**

Master of Science in **Modelling and Computational Science**

Thesis title:
Discrete Optimization of Upgrade Scheduling

An oral defence of this thesis took place on August 19, 2022 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Professor Mehran Ebrahimi
Research Supervisor	Professor Lennaert van Veen
Research Co-supervisor	Professor Daniel Hoornweg
Examining Committee Member	Professor Greg Lewis
Examining Committee Member	Professor Nader Azad

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

A primary objective of the mission to meet climate change goals of reducing greenhouse gas (GHG) emissions is to transition from fossil fuels to zero-emission energy. Fossil fuel production and transportation account for approximately half of the GHG emissions in Canada, making transitioning to zero-emission vehicles (ZEV) a climate action cornerstone. However, a 100% ZEV transportation system is beyond the capacity of Canadian electrical infrastructure in some areas, and the cost to upgrade these systems will be significant. The null hypothesis of this study is that there does not exist an approach to upgrades that optimally reduces the financial burden; conversely, the alternative hypothesis is that such an approach does exist. Mathematical rigour confirms the alternative hypothesis (with assumptions). Computational simulations reject the null hypothesis. The overall average of cost savings within satiable constraints (a subset of generated constraints in proportion to upgrade costs) is 34.5% (μ) $\pm$ 18.0% (σ).

Keyterms: climate action; green transition; electrification; optimization; operational research

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

______________________________________ Derick Smith

Statement of Contributions

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Acknowledgements

"That others may live."

– Guardian Angels, 131st

I lack the ability to effectively describe the true value of being in a program with peers, leaders, and mentors that understand the burden of climate change and, more importantly, that are able to provide the support, guidance, and wisdom that a student, such as myself, needs in order to make a meaningful contribution, which would otherwise not be possible. Any successes in this thesis are thanks to the individuals in these Acknowledgements. Any shortcomings in this thesis are my sole responsibility and are a consequence of my inability to actualize the collective brilliance of those around me.

Professor Jane Breen, your insights regarding data aggregation through sub-graph nesting are this model (I only hope that I have done your ideas justice). Professor Lennaert van Veen, your subtle guidance and motivation gave me the encouragement to explore topics I considered beyond my abilities. Professor Daniel Hoornweg, your vision for the real-world applications of ideas is a quality I aspire to develop. Professor Greg Lewis, your course in Advanced Topics in Modelling is what I truly believe to be the future of learning.

My friends and family. I serve many masters, but it is you that I serve above all others. I love you with all my heart.

Contents

Preface

Part Contents:

1. Structure

2. Context

3. Critique

"By any means necessary."

\- Malcolm X[5]

Preface: Structure

1. The overall structure of this thesis is designed to get to the point. In the event that any of these ideas might be useful to the fight against climate change, the intention behind the thesis structure is to make the ideas as accessible as possible to those that need them.

2. The use of curly braces does not serve the more traditional usage, which is similar to parentheses, but typically used to list items. Instead, curly braces in this thesis are used to group a set of words together that would otherwise have ambiguous meaning or a meaning that is difficult to interpret.

   (a) For instance:

      i. A government has discovered significant flaws in its economy, system of logistics, and electrical equipment that have insufficient capacity for projected demands.

   (b) Does "insufficient capacity" refer to "economy, system of logistics, and electrical equipment" or "electrical infrastructure" alone?

(c) The meaning of the above could potentially be derived with the inclusion of additional context or a different arrangement of words. Instead, the following is used:

   i. A government has discovered significant flaws in its {economy, systems of logistics, and electrical equipment} that have insufficient capacity for projected demands.

   ii. A government has discovered significant flaws in its economy, systems of logistics, and {electrical equipment that have insufficient capacity for projected demands}.

(d) The above example is, by no means, a perfect or canonical example of the use of curly braces. It is simply meant to illustrate the potential benefit of borrowing an undisputably useful tool from mathematics (and also computer science) that is not used in dogmatic written languages.

   i. In mathematics, the use of parentheses is not always needed to clarify the orders of operations of an expression; however, it can make the expression much more readable and easier to work with.

   ii. In computer science, there are instances where nested parentheses are unavoidable as the meaning would otherwise be ambiguous.

(e) Perhaps a better example:

   i. “We will crack down on tax havens and illicit financing that contribute to income inequality, fund terrorism, and generate pernicious foreign influence” ("FACT SHEET: Establishing the fight against corruption as a core U.S. national security interest,” 2021)[16].

(f) Which interpretation is correct?

   i. We will {(i) crack down on tax havens and illicit financing that contribute to income inequality, (ii) fund terrorism, and (iii) generate pernicious foreign influence}.

   ii. We will crack down on tax havens and illicit financing that {(i) contribute to income inequality, (ii) fund terrorism, and (iii) generate pernicious foreign influence}.

(g) Cultural context suggests that the correct interpretation is (2). However, considering U.S. history, it may very well be (1).

(h) All this being said, curly braces are used sparingly in an effort to prevent the introduction of new notation from being too jarring, which would defeat the purpose of its use in the first place.

3. The use of enumerated point form is intended to provide layers of depth to the material. It also provides the ability to communicate ideas that would otherwise be difficult to express in a linear form effectively. The best instance of where this enumerated point form is considered standard is in legal material. It is not necessarily the easiest to read, but it is exceptionally functional.

   (a) The curly braces and enumerated points are also used to simplify statements that are more complex. This format might not be necessary for naturally strong readers, but could be helpful for individuals with conditions such as dyslexia. In neuroscience, working memory refers roughly to the ability of a brain to hold blocks (or chunks) of information that are computable. These blocks might be as simple as symbols (characters such as letters or numbers) or as complex as the Field Equations for General Relativity by Einstein (more specifically, a reference to the equations). The point of this thesis is not to blow the stack of working memory blocks of the reader; rather, the main point of this thesis is to efficiently and effectively communicate ideas that could potentially be useful in climate action, especially the optimization model.

      i. The main point might be difficult to believe, considering the format breaks scientific and academic traditions. However, the breaking of traditions is a smaller point of this thesis compared to the broader point of presenting new ideas regarding the model of theses in general. In other words, I am arguing that just because tradition is comfortable does not mean it is optimal.

   (b) Take the following excerpt from the work of Professor Noam Chomsky regarding the concept of three major factors in language design for example: "Many scientists agree with paleoanthropologist Ian Tattersall, who writes that he is 'almost sure that it was the invention of language' that was the 'sudden and emergent' event that was the 'releasing stimulus' for the appearance of the human capacity in the evolutionary record—the 'great leap forward' as Jared Diamond called it, the result of some genetic event that rewired the brain, allowing for the origin of modern language with the rich syntax that provides a multitude of modes of expression of thought, a prerequisite for social development and the sharp changes of behavior that are revealed in the archaeological record, also generally assumed to be the trigger for the rapid trek from Africa, where otherwise modern humans had apparently been present for hundreds of thousands of years (Tattersall 1998:24–25;see also Wells 2002)" (Chomsky, 2005, p. 3)[11].

(c) Compare the above to the following: "Many scientists agree with paleoanthropologist Ian Tattersall,

   i. who writes that he is 'almost sure that it was the invention of language' that was the 'sudden and emergent' event that was the 'releasing stimulus' for the appearance of the human capacity in the evolutionary record

      A. —the 'great leap forward' as Jared Diamond called it,

      B. the result of some genetic event that rewired the brain,

      C. allowing for the origin of modern language with the rich syntax that provides a multitude of modes of expression of thought,

      D. a prerequisite for social development and the sharp changes of behavior that are revealed in the archaeological record,

   ii. also generally assumed to be the trigger for the rapid trek from Africa,

      A. where otherwise modern humans had apparently been present for hundreds of thousands of years (Tattersall 1998:24–25;see also Wells 2002)}" (Chomsky, 2005)[].

(d) Prof. Chomsky is sometimes referred to as the father of modern linguistics (as well as a political activist and pioneer in exposing U.S. propaganda with notable work on the concept of manufacturing consent). His proficiency with language gives him the ability to express ideas through words that are substantially more complex than what might be found in average academic work; however, this does not guarantee that the material is particularly accessible.

   i. For example, the structure of 3b is a linear expression of a multidimensional statement, whereas 3c is a multidimensional expression of a multidimensional statement. The nesting of enumerated points of {3c or any other example in this thesis} is not necessarily perfect. In fact, there is any number of arrangements that are computationally equivalent. i.

      A. Computational equivalence example: Recipes for peanut butter and jam sandwiches may differ in how the instructions are presented, but if the recipes produce the same sandwich, they are computationally equivalent.

(e) The point I am making is not that I have discovered the most optimal form of communication (it is only the first attempt at something new that might be useful to the scientific community). Instead, I am arguing that the current traditions for language usage are suboptimal (particularly in a field such as climate action that has approximately two and a half years to peak GHG emissions), and the continuing scientific standards should not be maintained for the sake of tradition.

Preface: Context

1. Before reading this thesis, there are some important points to keep in mind.

2. If any reader is sensitive to the reality of climate change, this thesis (including this preface) is not for you. In this preface especially, I am intentionally honest about my feelings regarding the dynamics involved in climate change without any sugar-coating. One reason is to demonstrate the psychology of the person who produced this thesis so that the reader can mindfully watch for any biases that might result in unintentionally unscientific ideas. Another reason is to contest the tradition prevalent in academia, where the effects of conflicts of interest receive little attention (if any). I argue that it is intellectually dishonest and an active contributing factor in climate inaction. If it is unclear what conflicts of interests I am referring to, I am referring to the influence of money from the fossil fuel industry in climate change research. Would you trust research funded by the tobacco industry on whether or not cigarettes cause cancer? The correlation between research outcomes, media interpretation of those outcomes, and fossil fuel funding might be an interesting investigation.

   (a) It is quite possible that these corrupt actions of contributing to the destruction of the ecosphere could be considered crimes against humanity in the near future, and the perpetrators could be retroactively charged for their knowing participation. It is common for laws to be implemented reactively; however, in exceptional cases, there are international legal precedents of charges being brought against perpetrators for actions, which occurred before a newly created law.

   (b) A prime example of violations of an **x-law** (**x-law** being a variable, which can represent any law) occurring prior to the creation of the **x-law** is from the Justice Case of the Nuremberg Trials (NMT Case 3) of the International Military Tribunal.

i. From the trial transcripts: Judge Mallory B. Blair explains that "[the UN] General Assembly is not an international legislature, but it is [the] most authoritative organ in existence for the interpretation of world opinion. Its recognition of genocide as an international crime is persuasive evidence of the fact. We approve and adopt its conclusions. Whether the crime against humanity is the product of statute or of common international law, or, as we believe, of both, we find no injustice to persons tried for such crimes. They are chargeable with knowledge that such acts were wrong and were punishable when committed" ("Transcript for NMT 3: Justice Case." Harvard Law School Library., 2020, p. 10647)[56].

ii. Immediately following the above statement, Judge Blair explained that "enactment also provides 'the fact that any person acted pursuant to the order of his government or of a superior does not free him from responsibility for a crime, but may be considered in mitigation.' (C.C. Law 10, Article II, paragraph 4 (b))" ("Transcript for NMT 3: Justice Case." Harvard Law School Library., 2020, p. 10647) [56]. This statement refers to the legal defence well-known as the **Superior Orders** defence. Specifically, Judge Blair explains that "just following orders" does not guarantee innocence.

(c) The concept of ecocide has gained a significant amount of attention in recent decades. As of the 10th of June 2022, searching for "ecocide" (with quotes to require the exact term) on Google Scholar returns approximately 18,000 results (Google Scholar, 2004).

i. During the International Criminal Court (ICC) General Debate in December of 2021, a panel of international legal experts from Stop Ecocide International proposed amending the Rome Statute of the ICC to include ecocide as a fifth category. In the proposal, the term ecocide "means unlawful or wanton acts committed with knowledge that there is a substantial likelihood of severe and either widespread or long-term damage to the environment being caused by those acts" (Stop Ecocide International, 2020)[47].

(d) Although ecocide has yet to be adopted by the ICC, according to international precedents, it is possible for ecocide charges could be brought against the fossil fuel industry and any accomplices for violations from the past, present, and future. Further, it is possible that "just following orders" of the fossil fuel industry will not provide sufficient legal protection for any crimes that may have been committed.

3. Perhaps the most important point to keep in mind is that the following comments express my perspective. Although an effort is made to remove any and all apparent biases in this thesis, my perspective as a human being in the midst of a climate crisis is not reducible to zero.

   (a) In other words, the contents of this preface are not rigorously journalistic. Although there exist quoted facts and calculations, they are included to more accurately express my biases. Furthermore, any use of ethos is done purposefully (i) to convey the psychological depth of my perspective and (ii) to communicate the urgency of climate change in a way that academia - and the scientific community as a whole - has failed to do in the past half-century.

   (b) A good example of the failures of academia is in the mathematics community. Despite the arduous training, testing, and gate-keeping process, it is apparently not worth much, as fossil fuel companies have relied on mathematicians to accomplish their goals. It seems as though the mathematics community is in need of some new standards to hold its members accountable.

   (c) To put this into perspective, if a significant percentage of chemistry students were graduating from universities and going to work for companies to produce chemical weapons that were being used to kill hundreds of millions of people all around the world, I find it difficult to believe that society would not immediately institute policies to prevent the corruption of our scientific workforce. Yet, somehow STEM students going to work for the fossil fuel industry is not objectionable at all, despite the impending climate change apocalypse due largely to the actions of the fossil fuel industry.

   (d) To counter the corrupting influence of the fossil fuel industry, I propose the implementation of a green intellectual property commons (GIPC [gip-see]) that places creative works outside the legal reach of the fossil fuel industry and provides leverage to entities working in climate action.

      i. With the help of legal experts and software developers, an automated (or partially automated – such as checklists and qualifier questionnaires) legal assistant could be developed to streamline the IP rights process.

ii. For example, the following is a modification of the famous MIT license for software:

OTU GREEN LICENSE (MODIFIED MIT LICENSE) Version 1, 20 MAY 2022

Copyright [INSERT DATE] [INSERT ENTITY NAME]

Permission is hereby granted, free of charge, to any entity that is not in the FOSSIL FUEL INDUSTRY obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit any entity to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. Additionally, any copyright of any derivative of the Software may never grant permissions to any entity in the FOSSIL FUEL INDUSTRY to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of any derivative of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. THE "FOSSIL FUEL INDUSTRY" REFERS TO ANY ENTITY THAT HAS A LEGALLY RECOGNIZED ASSOCIATION WITH ANY ENTITY IN THE BUSINESS OF FOSSIL FUELS.

4. For instance, the thesis focuses on the transition to zero-emission vehicles and supporting infrastructure. The reason behind selecting this area of work is because it has compounding results in greenhouse gas emissions. The greenhouse gas emissions from transportation make up only one slice of the emissions pie, but the elimination of the demand for fossil fuels has a

direct effect on fossil fuel production. In other words, from my vantage point, the aggressive transition to zero-emission vehicles has the most devastating impact on the fossil fuel industry as a whole. This decision has more to do with preventing climate change catastrophe and less to do with a personal grudge against those involved in the industry (as understandable as it might otherwise be).

5. The *fossil fuel industry* is referenced frequently, especially in this preface section. The term refers to the collection of individuals and entities that act in the interests of the industry. This term might be interpreted as vague and monolithic, considering the accusations and evidence that are presented in the following sections. However, monolithic descriptions are common in legal and academic material.

    (a) Within the fossil fuel industry, there exist competing interests. However, with regard to the continuing existence of the fossil fuel industry beyond what is considered scientifically sustainable for the biosphere, there is no significant degree of dissent. This is factually the case since the current heading of the fossil fuel industry includes the influence of dissenters and non-dissenters.

6. (Please note that the following comparison to Nazis is made thoughtfully and without any form of exaggeration. There is a real possibility that the climate crisis will result in atrocities that are far worse than the combination of all atrocities in history. The "real possibility" is a virtual certainty on the current course of climate inaction around the world.) More specifically, consider the genocide of the Jewish people in Nazi Germany (an over-used example, perhaps, but undeniably useful). The term "Nazi" is not commonly disputed as an inaccurate description of the power structure that orchestrated one of the worst atrocities in human history, despite there being any number of counter-interests within the Nazi party and within the country of Germany. The concentration camps were in alignment with the Nazi rhetoric and agenda, although the planning and execution can be attributed to a small number of specific individuals. When I refer to the "fossil fuel industry," I am arguing that there exists a monolith that is more distinct than that of the Nazis. Regardless of what counter-argument there might be, this describes my views on the matter and is the basis of my analysis of historical and modern-day events.

7. The last point regarding the explanation for expressing my biases: In my opinion, it is crucial

that climate change research be as transparent and honest as possible, considering the dishonest and powerful actors that do not want effective solutions to proliferate. It could be argued that it is important in all areas of research; however, the consequences of allowing an industry to influence the research outcomes on climate change could mean the difference between the continuation of life on Earth and mass extinction. This being said, I have not received any funding from the fossil fuel industry. Any work that receives fossil fuel industry funding should receive severe levels of scrutiny - if not discarded entirely - due to the dangerous conflict of interests.

(a) I am not claiming that I am innocent of GHG emissions - we are all hypocrites -, but I do my best to: minimize my carbon footprint by eating plant-based when I can; drive as little as possible (with a plan to transition to EV as soon as possible); not flying; and, generally, live as a minimalist to reduce my contribution to the demand of products with significant GHG emissions. This being said, I still have not zeroed out my carbon footprint. Living off-the-grid might be a fair choice for any one individual; however, we still need people to work on climate action, which involves operating within a GHG emitting system.

Preface: Context: Means of Production

1. The {design, research, and development} of technologies such as electrical infrastructure requires teams of scientists, engineers, and years of experience; however, the automation of manufacturing certain equipment is already within reach. Civilization will be confronted with the reality that certain capitalistic solutions (that charge a substantial premium for basic infrastructure such as electrical equipment) are insufficient for achieving climate goals when the effects of climate change begin to ramp up (not if, when).

   (a) This statement regarding manufacturing and climate solutions are obvious truths and does not require a background in infrastructure logistics and electrical systems that I have.

      i. The production of electric motors is approaching the level of Lights-Out Manufacturing (LOM).

         A. LOM is a level of manufacturing where the production process does not include human intervention under normal operating conditions. The lights being out is not necessarily literal, as some machines may require vision systems.

    ii. The technology in electrical equipment, such as transformers, is arguably less complex than most electric motors. At scale, the cost to produce such electrical equipment approaches the cost of materials.

(b) The insufficiency of capitalistic climate solutions is not a political opinion. In the United States, for instance, governmental and non-governmental organizations that are responsible for providing the delivery of electrical energy are already struggling to keep up with the increasing number of service interruptions from the increases in demand due to cooling systems during increasingly hotter summers. Even worse, "[w]ith increasing temperatures, electricity demands for cooling increase, which can increase the risk of blackouts" (Hopkins, 2018, p. 51)[23].

    i. Let alone transitioning to EVs.

    ii. Let alone transitioning appliances to electric.

(c) There is no shortage of climate change problems that already exist or will exist. Even the wealthiest countries have a limited amount of resources, so resolving all of these problems while trying to manage an under-regulated market is not a plan for solving climate change.

    i. It is safe to assume that a vast majority of companies in charge of infrastructure are going to continue their pursuit to maximize profits. In fact, it is possible that they are planning to capitalize on this opportunity, despite their plans preventing effective climate action.

    ii. Under different circumstances, this kind of behaviour might be sustainable. However, the fact of the matter is that there are not enough resources to make these companies exceedingly rich and also solve the impending infrastructure crisis.

    iii. In contrast, automating the manufacturing of infrastructure equipment is significantly more likely to solve the energy grid crisis and spare a substantial amount of resources for use in other climate solutions.

2. The discrete optimization of upgrade scheduling might be an important aspect of retrofitting existing infrastructure. However, putting an end to the installation of non-electric appliances in new infrastructure is a far greater priority. Whatever the projected costs of retrofitting are, those numbers will more than double on a long enough time scale with the increase in

population. From my perspective, this point has been overlooked by the vast majority of the population, but, realistically, it will be seen - sooner or later - as atrocious mismanagement of resources.

Preface: Context: Climate Corps

1. Perhaps the most important point: the problems of climate change (now and in the future) are beyond the capability of uncoordinated efforts. The destruction of the ecosphere (with the effects of climate change being a major contributing factor along with the direct destruction of the biosphere) is a larger problem than any enemy or crisis that society has ever experienced. It is not a mistake that "President Biden considers the climate crisis to be a top priority from a foreign policy and national security perspective" (U.S. Department of State, 2021)[51]. Despite the size of the climate change problem appearing to be insurmountable, a commonly circulating plan is to implement programs and policies that incentivize a market to produce solutions. A capitalistic-focused plan is insufficient for solving climate change. There is a reason why company executives do not hold official seats of power inside the Pentagon of the United States. A plan that could potentially solve climate change is the implementation of a **climate corps or something like it**.

    (a) If a person wants to defend their country from enemies - foreign and domestic - they can sign up at their nearest recruiting office to serve their military. However, if a person wants to contribute to climate action, they can – what? Society is not going to solve climate change accidentally. Moreover, as much as I love grassroots organizing, it is, at best, a means to an end. We cannot expect small climate militias to simultaneously conquer the fossil fuel industry and climate change part-time on shoestring budgets.

    (b) The creation of concentration camps is not speculation. Countries have become more authoritarian in the past few decades. With increased tensions during the migration of millions (upwards of hundreds of millions) of refugees, it is unlikely that struggling host countries will be able to respond compassionately.

        i. For instance, some studies show a strong trend in the rise of authoritarian rule around the world.

            A. "The global order is nearing a tipping point, and if democracy's defenders do not work together to help guarantee freedom for all people, the authoritarian model will prevail" ("global expansion of authoritarian rule," 2022)[18].

B. "The present threat to democracy is the product of 16 consecutive years of decline in global freedom. A total of 60 countries suffered declines over the past year, while only 25 improved. As of today, some 38 percent of the global population live in Not Free countries, the highest proportion since 1997. Only about 20 percent now live in Free countries" ("global expansion of authoritarian rule," 2022)[18].

ii. A specific example of this authoritarianism is the handling of immigrants at the U.S. southern border.

A. "They are concentration camps. I want to talk to the people that are concerned enough with humanity to say that 'never again' means something. The fact that concentration camps are now an institutionalized practice in the 'home of the free' is extraordinarily disturbing and we need to do something about it," says Alexandra Ocasio Cortez (Most, 2019)[34].

iii. An expert in Jewish studies recognizes the need for establishing early warning standards against atrocities such as concentration camps and recommends an update to the "never again" slogan.

A. "A better slogan, instead of 'never again,' is the German 'Wehret den Anfängen,', which means: 'Resist the beginnings.' How can we be indifferent when we see thousands of innocent civilians, fleeing murder and abuse in their countries of origin, rounded up and placed in detention centers, abused, families separated, people placed in cages, children dying?" asked "Michael Zank, a College of Arts & Sciences professor of religion and Jewish studies and director of BU's Elie Wiesel Center for Jewish Studies" (Most, 2019)[34].

iv. With many of the harsh U.S. border policies and programs from the Trump-Pence administration still in place, the inability to provide proper support to climate refugees could be among the worst tragedies in U.S. history.

A. "[Afghanistan, Burma, India, Pakistan, North Korea, Guatemala, Haiti, Honduras, Nicaragua, Colombia and Iraq] will lack the financial resources or governance capacity to adapt to climate change effects, heightening the risk of instability-induced migration and displacement flows—including to the U.S. southern border" (National Intelligence Estimate: Climate change and international responses increasing challenges to U.S. national security through 2040, 2021, p. 17)[36].

v. This discussion is only referring to the United States. With the influx of hundreds of thousands - potentially millions - of refugees, it is unlikely that other countries will have better outcomes, especially considering that they have even fewer resources at their disposal.

(c) Aside from military defence forces such as the U.S. Department of Defense, there does not exist a workforce large enough to tackle the operational objectives required to prevent a catastrophic collapse of {civilization and, potentially, the global ecosystem}. What is needed is a **climate corps or something like it**. If a civilian climate corps is not implemented soon enough, the something-like-it will be in the form of martial law.

i. In fact, Professor Hoornweg even predicted in the Spring of 2020 that governments will implement martial law to address climate change within the next ten years.

Preface: Context: Delusion

1. The climate denial narrative is beginning to weaken; however, delusion is more prevalent than ever. I am trying to convey here that the sense of urgency that was required half a century ago is insufficient for the present day and increasingly so as time progresses. Everyone on this planet is a climate activist; they are just not aware of it yet.

2. The climate action discussion often lacks the acknowledgement of the opposition forces that manufactured and maintained the denial narrative for decades. The fossil fuel industry is the most profitable and one of the most powerful industries the world has ever seen. Their publicly available business models do not include plans to transition away from fossil fuels in time to make a difference. It is a scientific fact that their business plans are literally designed to create an unmanageable climate crisis. They have knowingly done so. Their knowledge of the impending dangers is well documented.

   (a) Current plans are heading towards 2 degrees warming, assuming policies and agreements remain in place.

    i. "Without taking into account the economic benefits of reduced adaptation costs or avoided climate impacts, global Gross Domestic Product (GDP) would be just a few percentage points lower in 2050 if we take the actions necessary to limit warming to 2°C (3.6°F) or below, compared to maintaining current policies" (Luz, 2022)[31].

  (b) Assuming conditions and progress remain unchanged is a dangerous assumption to make. The following might be difficult to believe, but the U.S. Republican party once supported climate action. In fact, they were among the strongest proponents in the world of the work done by the newly formed UN IPCC.

    i. "The George H. W. Bush administration entered office in 1989 with plans to build upon this success. Already in 1988, the U.S. had supported creation within the UN of the Intergovernmental Panel on Climate Change (IPCC) to carry out systematic research into the causes of global climate change and to assess potential strategies to address it" (Wampler, 2015)[54].

  (c) Unfortunately, when the Bush-Cheney administration came into power, the U.S. support of climate change policies was reversed – to say the least.

    i. "[T]he George W. Bush administration saw the pendulum swing back to disengagement and broad criticism of the Kyoto Protocol, as Bush pulled the U.S. out of the agreement and gave overriding priority to economic interests, primarily energy, a position driven by Vice President Richard Cheney" (Wampler, 2018)[55].

  (d) The Trump-Pence administration further dismantled crucial support in the global fight against climate change.

    i. "The most extreme manifestation of this opposition and denial is now seen in the Trump administration, which announced in June 2017 that it would withdraw from the 2016 Paris climate change treaty, and has rejected the idea of working with traditional allies in key global institutions such as the G7 economic summits that in the past were seen as essential forums for contributing to climate change negotiations" (Wampler, 2018)[55].

3. The fossil fuel industry is pushing for the development of direct air carbon capture (or Direct Air Capture, DAC) as a solution to climate change and, in some cases, as an argument for a substantially slower transition. Unfortunately, this is not feasible and the reason why is simple.

Even if a scalable DAC solution was created, there does not exist enough clean electrical energy to power such a solution. This conclusion can be achieved with a back-of-the-envelope calculation as follows:

(a) The theoretical lower bound of energy to separate carbon dioxide from the air at 400 ppm (gigajoules per tonne of carbon dioxide) (not counting the energy to compress, deliver, store the captured carbon dioxide, or regenerate the capturing medium):

  i. $\mathbf{0.43}$ GJ/tCO_2 (David Keith, Kenton Heidel, & Robert Cherry, 2010, p. 110)[14]

  ii. This lower bound is based on the negation of Gibbs free energy of mixing. Essentially, the potential energy lost {by the gas system} in the mixing of air (minus carbon dioxide) and carbon dioxide. This is, of course, equivalent to the potential energy gained {by the gas system} in the unmixing (or mixing with time-reversed) of carbon dioxide from the air. This minimum energy is explained more succinctly in the paper *Capturing CO_2 from the Atmosphere*[14] – one of the most effective and efficient explanations regarding the reality of DAC, in my opinion.

(b) The approximate amount of gigatonnes of carbon dioxide needed to be removed to go from {**417.4** ppm (present) to **278** ppm (pre-industrial)} (NASA Scientific Visualization Studio, 2022)[35] with the conversion factor of **7.8** Gt CO_2/ppm (Oak Ridge National Laboratory, 2012.)[40]:

  i. $7.8\dfrac{\text{GtCO}_2}{\text{ppm}} \cdot (417.4 - 278)\ \text{ppm} = \mathbf{1087.9}\ \text{GtCO}_\mathbf{2}$

(c) The total amount of energy to accomplish this is:

  i. $467813947893\ \text{GJ} \approx \mathbf{129948}\ \text{TWh}$

(d) **To put this into context:** In 2019, the entire world generated approximately **27044** TWh of electrical energy (IEA, 2021)[25]. Keep in mind that this was by no means near 100% clean energy production, but let us assume that it was for the sake of argument. If there existed (i) a scalable (near perfect) carbon-capturing device (with the efficiency described above) that could reduce the carbon dioxide concentration to pre-industrial levels, (ii) with all the electricity in the world dedicated to this capturing process, AND (iii) with net-zero emissions already achieved, it would take the following amount of time:

  i. $129948\ \text{TWh} \cdot \left(27044\dfrac{\text{TWh}}{\text{year}}\right) \approx \mathbf{4.8}\ \text{years}$

4. The lower bound estimate for how long it would take to return to pre-industrial levels might seem encouraging at first glance. However, this kind of technology does not exist and will never exist. The amount of energy to remove carbon dioxide may very well be ten times the minimum, which equates to a drastic change in the amount of time and energy required to remove it from the atmosphere.

5. Any reasonable person would find it inconceivable that the fossil fuel industry is unaware of the above calculation (the fossil fuel industry relies heavily on scientific minds such as chemists, physicists, and mathematicians, that have the ability to do the above calculations faster than I did, and they could even be intuitively aware without having to calculate at all). Yet, the fossil fuel industry continues to push a narrative regarding the hopefulness of DAC technology. If the fossil fuel industry were serious about DAC technology, they would be investing in the implementation of clean energy to the extent required to support the capturing of carbon dioxide.

   (a) If the fossil fuel industry is aware of the above calculation, their behaviour is arguably malicious towards humanity – some might even say evil (at the very least, without good intentions).

   (b) If the fossil fuel industry is unaware of the above calculation, they are undeniably incompetent.

   (c) Regardless, humanity must not give them the ability to make decisions regarding the future of life on Earth.

6. Suppose the fossil fuel industry was implementing such a large scale of clean energy (essentially doubling the global electricity output, but all from clean energy). It would beg the question of why we would need so much fossil fuel production since there would exist an abundance of clean energy.

7. The fossil fuel industry is more than willing to go to war with other countries, organizations, and even tribes of people to maximize profits. So the idea that the fossil fuel industry will voluntarily step down is a grand delusion. On the contrary, it is more likely that they will become more aggressive as the consequences of climate change become more devastating with time.

(a) The fossil fuel industry went to war and committed genocide in Iraq over oil. This statement is not meant as an oversimplification regarding the historically complex geopolitical dynamics in the Middle East. Rather, it is commentary regarding the commonly held belief that the primary motivation was not to spread democracy, but instead to exploit the resources of a country. The U.S. fossil fuel companies were awarded multi-billion-dollar oil contracts in Iraq – especially Halliburton (Dick Cheney was the CEO of Halliburton immediately before becoming Vice President). Capitalists capitalized on a catastrophe.

   i. The Bush-Cheney administration was not the first time that Cheney pushed for a war with Iraq. In fact, during the Bush-Quayle administration, Cheney (Secretary of Defense) instructed Paul Wolfowitz (Under Secretary of Defense for Policy) to "[s]et up a team, and don't tell Powell or anyone else." regarding the development of "an alternative plan for the Gulf War against Iraq" (Battle, 2010)[3], in late 1990.

   ii. Despite there being "no indications that there remains in Iraq any physical capability for the production of amounts of weapon-usable nuclear material of any practical significance" (Battle, 2010)[3], the U.S. decided to invade Iraq.

   iii. In 1998, during an official meeting with oil and gas associations, Dick Cheney stated that "[y]ou've got to go where the oil is" (Battle, 2010)[3], regardless of the geopolitical dynamics.

   iv. The oil magnate and financial supporter of the Bush-Cheney administration, T. Boone Pickens, stated to the U.S. Congress in 2009 that Iraq is "opening them (oil fields) up to other companies all over the world ... We're entitled to it. Heck, we even lost 5,000 of our people, 65,000 injured and a trillion, five hundred billion dollars .... We leave there with the Chinese getting the oil." (Battle, 2010)[3].

   v. Pickens failed to mention that "[t]here have been between 184,382 and 207,156 Iraqi civilians killed by direct violence since the U.S. invasion. [...] The actual number of civilians killed by direct and indirect war violence is unknown, but likely much higher" (Crawford, Lutz, & Saleh, 2021)[13].

      A. The usage of the term genocide is controversial among historians. This term is conveniently avoided when referring to conflicts in, which the U.S. and their allies have been involved. However, if someone coins a term that describes the mass killing of civilians outside of mutually declared war and the brutalization of their culture, I will use it.

(b) The Bush-Cheney Iraq invasion was supposedly in response to the 9/11 attack and to liberate the Iraqi people from their dictator, Saddam Hussein, based on the controversial intelligence reports claiming Iraq was developing weapons of mass destruction. Weapons of mass destruction were never found. Iraq groups were not responsible for the 9/11 attack. What reasonable person would consider {the U.S. taking control of oil in Iraq and killing groups of people that were attempting to take their resources back}, in any way, an accident? Hundreds of thousands of innocent Muslim people were killed. Entire families were killed. Children were killed. CHILDREN. It was done knowingly and with intent. People can claim that the sky is not blue if they would like to, and, unsurprisingly, some do.

    i. The following is the definition of **genocide** according to the UN Genocide Convention:

    ii. "Article II In the present Convention, genocide means any of the following acts committed with intent to destroy, in whole or in part, a national, ethnical, racial or religious group, as such:

        A. (a) Killing members of the group;

        B. (b) Causing serious bodily or mental harm to members of the group;

        C. (c) Deliberately inflicting on the group conditions of life calculated to bring about its physical destruction in whole or in part;

        D. (d) Imposing measures intended to prevent births within the group;

        E. (e) Forcibly transferring children of the group to another group" (United Nations, 1951, p. 1)[53].

    iii. The definition requires intent. From my perspective, the word intent can be used as a weasel word. A weasel word is a word that blurs the meaning of statements. For instance, it is common to claim that intent is difficult to prove; however, in some cases, the exact intended result is irrelevant. More specifically, consider the bombing of a funeral because there is a suspected terrorist present and knowingly killing an entire group of innocent people (civilian non-combatants). It is possible to argue that killing the entire group was not the exact intent of the act. The official intents of drone kill orders are to extra judiciously execute a suspected terrorist. It is unlikely that the official documents will express the intention of killing innocent civilians – this would be a self-admitted war crime.

iv. If the above is not convincing enough, let us take the legal definition of **intent** to a logical absurdity. Consider a person that throws another person off the top of a tall building. The perpetrator can legally claim that they did not intend to kill the victim, but instead wanted to see if the victim could fly. Suppose the perpetrator displays a reasonably sound mind and a minimum understanding of physical reality. In that case, it is a reasonable conclusion that they committed the crime in the full knowledge that the act would kill the victim with virtual certainty. However, this does nothing to prove intent. So is it manslaughter? NO, that is a bad faith argument; it is murder, and the Iraq invasion was a genocide.

v. The Bush-Cheney Iraq invasion knowingly [UN Genocide Convention: Article II, "intent"] killed [UN Genocide Convention: Article II, "any"; Article II(a)] hundreds of thousands of innocent Iraqi Muslims [UN Genocide Convention: Article II, {"national", "religious"}] over the course of two decades. The Bush-Cheney Iraq invasion knowingly [UN Genocide Convention: Article II, "intent"] overthrew the governing body, destabilized the country, and destroyed [UN Genocide Convention: Article II, {"any", "destroy"}; Article II(c)] the Iraqi control of their natural resources against their will.

   A. In case there are objections regarding the point regarding the targeting of a particular religious group, consider the following counter-factual thought experiments:

      (i) If the Bush-Cheney administration invaded Germany in response to 9/11, it is conceivable that the international community would not have been as silent or complicit.

      (ii) If the Bush-Cheney administration invaded Israel in response to 9/11, killed hundreds of thousands of Jewish Israelis, and took control of their energy resources, it is unlikely the international community would have been as silent or complicit. It is conceivable that this invasion would be considered genocide without question.

(c) The fossil fuel industry has paid politicians and lobbyists to prevent effective climate action policies.

    i. According to journalists at Influence Map, "[the five largest oil companies (ExxonMobil, Royal Dutch Shell, Chevron, BP and Total)] have invested over $1Bn since the Paris Agreement on misleading climate lobbying and branding activities. The overriding intention and net result of these efforts has been to stall binding and increasingly crucial policy designed to implement the Agreement by national governments. " (Influence Map, 2019)[27].

(d) The fossil fuel industry hired a counter-intelligence organization of ex-military members to target activists and infiltrate groups taking part in the protest of the Dakota Access Pipeline, including fellow veterans. It was a protest movement that I openly supported and funded despite serving in the U.S. military at the time.

    i. "A shadowy international mercenary and security firm known as TigerSwan targeted the movement opposed to the Dakota Access Pipeline with military-style counterterrorism measures, collaborating closely with police in at least five states, according to internal documents obtained by The Intercept" (Brown, Parrish, & Speri, 2017)[7].

(e) The industry has lobbied to create catch-all laws that label activists as {terrorists attacking energy infrastructure}.

    i. "The first critical infrastructure law aimed at pipeline protesters passed in the immediate aftermath of the Indigenous-led uprising against the Dakota Access pipeline near the Standing Rock Sioux Reservation in North Dakota. The fossil fuel industry, in partnership with law enforcement, convinced legislators in numerous states that they were at risk of their own local Standing Rock — with accompanying security costs. Oklahoma passed an anti-protest law in May 2017, and by the end of the year, ALEC had developed the law into model legislation" (Brown, 2020)[6].

8. The fossil fuel industry is, by many measures, the most dangerous group of individuals on the planet. Not recognizing this is perhaps the greatest delusion of all.

9. I write in open defiance without fear. Not because I believe I am outside the reach of the brutal authoritarian economic regime that is the fossil fuel industry. I write in open defiance

without fear because the consequences of not doing enough to stop the worst outcomes of climate change are infinitely worse than anything the fossil fuel industry can conjure. They view people like me (those actively fighting for climate action) as the enemy. The feeling is more than mutual, I assure you.

Preface: Critique

1. It would be exceedingly hypocritical of me to critique the lack of effective solutions in society regarding climate change and not accept critique myself.

2. Contained in this thesis are ideas that I have developed with the intention of providing a meaningful contribution to climate change solutions. I welcome the reader to not hold back any and all critiques. If this thesis can be significantly improved, I will do everything that I can to make improvements. If this thesis is not salvageable as a contribution to climate change solutions, I will move on to the next area where I might make the greatest possible contribution to the climate fight.

3. As a personal trainer, a bit of wisdom I would share with clients is, "You don't have to be honest with me; just be honest with yourself." Do you honestly believe that what we are doing as a society is enough to prevent the worst atrocity in human history?

Part I
Introduction

Part Contents:

1. Overview
2. General Problem
3. Specific Problem
4. Hypothesis
5. Summary

"No."

– Rosa Parks

1 Introduction: Overview

1. Half a century ago, humanity had the luxury of dedicating nearly all climate action resources to a single front, prevention. In the present day, the distribution of climate action resources is now spread across multiple fronts, including the accelerating humanitarian crises and the accompanying conflicts. The reality of multiple fronts is a key aspect that is not taken into account by the business plans of the fossil fuel industry since they interpret scientific reports as a user guide for maximizing profits at the risk of mass extinction. The international community is already struggling with transitioning to renewable technology. The compounding consequences of climate change will not make the global energy transition any easier. In fact, it is possible that it could bring this transition to a halt.

(a) The response to the {2022 invasion of Ukraine and the consequences on energy supply} open the door to more climate action avoidance. An example of such a response is the idea that "[w]e definitely at this time need to include all available resources," and "[w]e cannot ignore or say we are going to abandon certain production. It's just not the right time, whatever reason you have" (Al Jazeera, 2022)[2].

(b) According to the 2022 reports by the IPCC, the climate crisis will affect virtually every human on Earth, directly or indirectly.

   i. "Climate variability and change already negatively impacts the health of tens of millions of Africans through exposure to non-optimal temperatures and extreme weather, and increased range and transmission of infectious diseases (high confidence)" (Pörtner et al., 2021, §9.10.1)[43].

   ii. "Dryland populations exposed (vulnerable)to water stress, heat stress, and desertification are projected to reach 951 (178) million at 1.5°C, 1152 (220) million at 2°C, and 1285 (277) million at 3°C of global warming" (Skea et al., 2019, p. 50)[46].

(c) Experts in U.S. national security currently project an increase in conflicts around the world as the consequences of climate change become progressively worse.

   i. The U.S. National Intelligence Council "assess that most countries that rely on fossil fuel exports to support their budgets will continue to resist a quick transition to a zero-carbon world because they fear the economic, political, and geopolitical costs of doing so" (National Intelligence Estimate: Climate change and international responses increasing challenges to U.S. national security through 2040, 2021, p. 17)[36].

   ii. To make matters worse, "[t]he reduction in sea ice already is amplifying strategic competition in the Arctic over access to its natural resources. Elsewhere, as temperatures rise and more extreme effects manifest, there is a growing risk of conflict over water and migration, particularly after 2030, and an increasing chance that countries will unilaterally test and deploy large-scale solar geoengineering—creating a new area of geopolitical disputes" (National Intelligence Estimate: Climate change and international responses increasing challenges to U.S. national security through 2040, 2021, p. 17)[36].

2. With the impending crises (including the increases in tensions and number of conflicts), the

idea that it is not the right time to transition to renewables could potentially mean it will never be the right time. This narrative also lacks an appreciation for the large array of climate change problems humanity faces beyond renewable energy systems.

3. This thesis aims to contribute to the vast field of climate change research that ranges from the understanding of climate change problems to the implementation of climate change solutions. In particular, the minimization of costs in the transition to green technologies emphasizes the upgrading of electrical infrastructure needed to support the transition to 100% EV transportation.

4. The following is a bird's-eye view of the intellectual idea space of climate change and the location of this thesis within it:

5. **Climate Change (CC):**
    (a) Understand CC Problems
        i. Modelling
            A. Atmospheric
            B. Sociological
            C. ...
        ii. ...
    (b) **Create Scalable CC Solutions**
        i. Reduction of Harm
            A. Global Cooling Actions (unproven)
            B. Disaster Response
            C. ...
        ii. **Reduce Net Greenhouse Gas Concentrations**
            A. Dismantle the Fossil Fuel Industry
            B. **Transition to Green Energy Technology**
            C. Transition to Sustainable Agriculture
            D. GHG Capture Technology

E. ...

iii. ...

(c) ...

6. The location of this thesis is within the Transition to Green Energy Technology idea space. Specifically, the transition to 100% EV requires an upgrade in electrical infrastructure to meet the demands of charging on the network.

7. This introduction section aims to present the lineage of ideas that provide the context and justification for the construction of the model presented in this thesis. Although the model is specific to the optimization of upgrade schedules in the context of electrical transformers, the justification of this thesis as a whole begins with the general problem of climate change.

2 Introduction: General Problem

2.1 Introduction: General Problem: Climate Change (CC)

1. The scientific debate regarding the existence of climate change has long been settled despite (social, cultural, geopolitical, and corporate) perspectives, which claim otherwise. The general problem of climate change in this thesis is the set of problems that can be understood and solved scientifically (as opposed to direct conflict with those opposing climate action).

2.1.1 The Consequences of Human-caused CC

1. Although it may seem that this area of research has received a significant amount of attention, there are certain consequences of climate change that may be beyond the abilities of modern-day technology because of the size of the problem (the entire world). The following is a small selection of consequences that are already well understood.

2.1.1.1 Present Consequences

1. Climate change is often referred to in the future tense. However, at present, "[w]idespread, pervasive impacts to ecosystems, people, settlements, and infrastructure have resulted from

observed increases in the frequency and intensity of climate and weather extremes, including hot extremes on land and in the ocean, heavy precipitation events, drought and fire weather (high confidence)" (Pörtner et al., 2021, SPM.B.3.3)[43].

2.1.1.2 Imminent Consequences

1. Suppose the current political response to the recent IPCC reports indicates the scenario humanity is heading. In that case, it appears unlikely that the average global temperature will be kept below 1.5°C. The amount of work required to achieve this goal would transform the global economy. Considering the current market is dependent on fossil fuels, if humanity is to get on track with climate goals, the policies and programs would need to be established and pursued immediately (measured in days, not weeks). The immediacy for action is due to the latency in climate solutions – such as standing up organizations, training a workforce, and distribution of funding. It does not appear to be the case that humanity has the policies and programs necessary, and it will have consequences in the near term.
   (a) "Global warming, reaching 1.5°C in the near-term, would cause unavoidable increases in multiple climate hazards and present multiple risks to ecosystems and humans (very high confidence)" (Pörtner et al., 2021, SPM.B.3)[43].
   (b) "Near-term warming and increased frequency, severity and duration of extreme events will place many terrestrial, freshwater, coastal and marine ecosystems at high or very high risks of biodiversity loss (medium to very high confidence, depending on ecosystem)" (Pörtner et al., 2021, SPM.B.3.1)[43].

2.1.1.3 Pending Consequences

1. With the anticipated growth of the fossil fuel industry and the weakness of political leaders to stand up against them, the consequences will be severe, to say the least.
   (a) For example, the "[b]iodiversity loss and degradation, damages to and transformation of ecosystems are already key risks for every region due to past global warming and will continue to escalate with every increment of global warming (very high confidence). In terrestrial ecosystems, 3 to 14% of species assessed will likely face very high risk of

extinction at global warming levels of 1.5°C, increasing up to 3 to 18% at 2°C, 3 to 29% at 3°C, 3 to 39% at 4°C, and 3 to 48% at 5°C" (Pörtner et al., 2021, CCP4.1.1)[43].

2.2 Introduction: General Problem: Greenhouse Gas (GHG)

2.2.1 Introduction: General Problem: GHG: Concentrations

1. There are counter-arguments to the modern-day models of the environment of Earth. Such criticisms are that the models are inaccurate. The simulations used to predict outcomes under various conditions are from models that are arguably some of the most advanced technology in the world. However, climate models never needed to progress as far as they have to determine that changes need to be made in the way society operates. In order words, it never needed to go further than the causal model of atmospheric CO_2 concentrations and its effect on global temperature.

2. Even with the modest accounting of carbon dioxide emissions, the potential dangers of continuing to emit greenhouse gases uncontrollably are obvious. The potential dangers are obvious in the sense that it only requires a few logical steps and a basic understanding of science to come to this conclusion.

   (a) An increase in carbon dioxide concentration will cause an increase in average surface temperature.

   (b) There exists an average surface temperature that will cause an ecosystem collapse.

   (c) Therefore, there exists a change in carbon dioxide concentrations that will cause an ecosystem collapse.

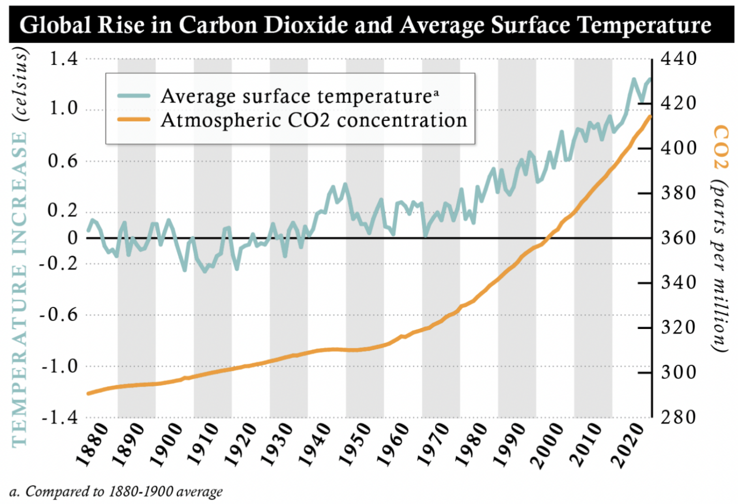

Figure 1:
This is a plot of the global concentration of atmospheric CO_2 and the average surface temperature from the year 1880 to 2020 (National Intelligence Estimate: Climate change and international responses increasing challenges to U.S. national security through 2040, 2021, p. 2)[36].

1. Figure 1 demonstrates the time series of human carbon dioxide emissions (cause) and the resulting average surface temperature increases (effect).

2. There is a common misconception about what averages represent. For instance, increasing the temperature of a home by two degrees may be a perceptible change, but it is not an unlivable change. So why is it so concerning if the average temperature of the Earth is increased by a few degrees?

3. In science (and many other fields), the average (mean) value is typically accompanied by the standard deviation. These two values are meant to approximate a distribution of data. These values imply a normal distribution – a bell curve. Specifying the mean and deviation is sometimes communicated implicitly.

   (a) For instance, some scientific result is described with the following numbers: 100 ± 10 . This result has an average of 100 and ranges from 90 to 110. The ± 10 is not necessarily a standard deviation. In some cases, $\pm$ represents an interval (such as a confidence or prediction interval) but, regardless, it is an effort to convey the approximate behaviour of an entire set of data with two values. Whether or not these values are reasonable approximations depends on context.

4. Understanding the distribution behind the average and deviation values is essential since the misinterpretation of their meaning can have tangible consequences.

   (a) For instance, if a seawall is 10 meters high and the average sea level is 7 meters high, it might appear that the seawall is sufficiently tall to prevent flooding. Further, it might appear that increasing the average sea level by 1 meter (to 8 meters) would not change the effectiveness of the seawall; however, it depends on the distribution of the sea levels throughout time.

5. The following is a simulated example of sea level measurements in reference to a seawall:

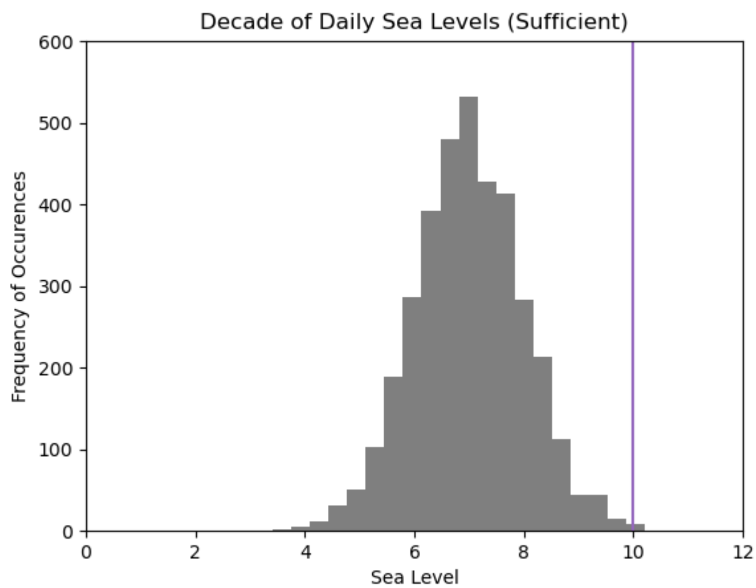

Figure 2:
This is a histogram plot of simulated daily sea level measurements over a decade with an average sea level measurement of **7 meters** with a standard deviation of **1 meter**. The grey bars represent the number of times that measurement occurred. The purple line represents the seawall height of 10 meters.

1. In Figure 2, with {an average of 7 meters and a standard deviation of 1 meter}, there are 8 measurements where the sea level is above the height of the seawall for an entire decade.

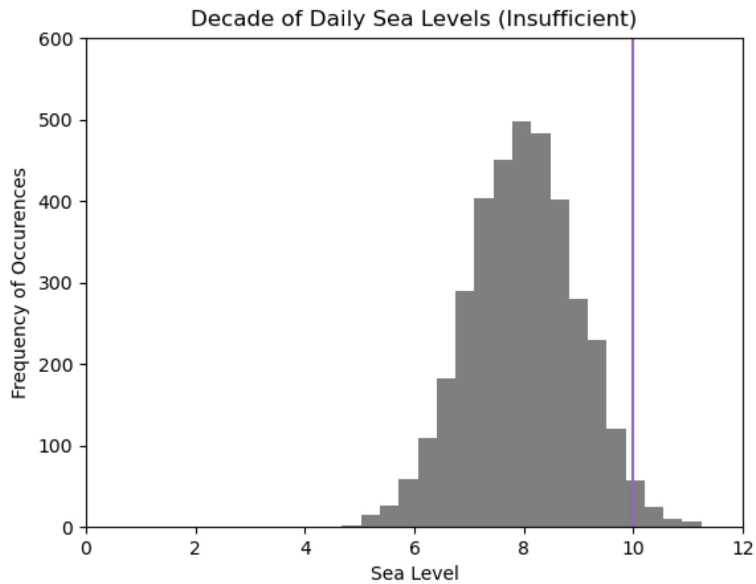

Figure 3:
This is a plot of simulated daily sea level measurements over a decade with an average sea level measurement of **8 meters** with a standard deviation of **1 meter**. The grey bars represent the number of times that measurement occurred. The purple line represents the seawall height of 10 meters.

1. In Figure 3, with {an average of 8 meters and a standard deviation of 1 meter}, there is a total of 8 measurements where the sea level is above the height of the seawall for an entire decade.

2. To put this into context, for the sake of argument, say that a city has a dedicated amount of money to put towards flood damage and that this amount of money is sufficient for a relatively small number of mild floods. Figure 2 demonstrates a set of measurements where the city would have a sufficient budget to repair the damage from the 8 mild floods. In contrast, Figure 3 demonstrates a set of measures where the city would have an insufficient budget to repair the damage from the 42 mild to severe floods. And this is only from a change of one meter on average.

3. Because of this property of distributions and a slight increase in the average can equate to devastating consequences. For example, in the context of climate change, a slight change in average global temperature might equate to extended heat waves and droughts that are beyond

the tolerance of various subsystems in the ecosystem.

2.3 Introduction: General Problem: In Canada

2.3.1 Industry Share of GHG Emissions

1. There exist many sources of GHG emissions, and each source must be reduced to achieve all the climate change emission goals. On the path to transitioning to clean energy, certain GHG emission sources have higher priorities than others. In Canada, for instance, a decrease in the GHG emissions from transportation can equate to a reduction in GHG emissions from oil and gas production, as far as domestic products are concerned.

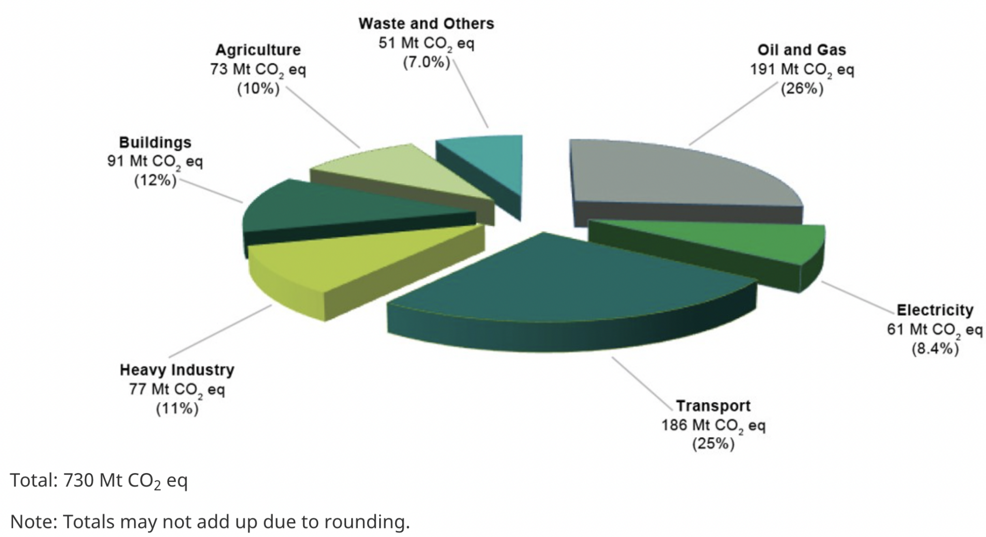

Figure 4:
This is a graphic of Canadian GHG emissions per industry expressed in megatonnes of carbon dioxide equivalent in 2019 (Environment and Climate Change Canada, 2021)[15].

1. Figure 4 demonstrates the various sources of GHG emissions in Canada from 2019. Observing this graphic of GHG emissions in the context of eliminating all of these GHG sources to achieve the goal of net-zero is a challenge, to say the least. Fortunately, Canada has implemented several policies and programs set to overcome this challenge in the following years.

2.3.2 Transitioning Away from Fossil Fuels

1. One such ambitious plan is "[t]he 2030 Emissions Reduction Plan is an ambitious and achievable roadmap that outlines a sector-by-sector path for Canada to reach its emissions reduction target of 40 percent below 2005 levels by 2030 and net-zero emissions by 2050" (Service Canada, 2022)[44].

2.3.3 Transitioning Vehicles to Green Energy

1. To complement the 2030 Emissions Reduction Plan, "the Government of Canada is setting a mandatory target for all new light-duty cars and passenger trucks sales to be zero-emission by 2035, accelerating Canada's previous goal of 100 percent sales by 2040" (Transport Canada, 2021)[49].

3 Introduction: Specific Problem

3.1 Introduction: Specific Problem: Electrical Grid

1. Although the transition to electric vehicles is certainly a difficult problem in and of itself, the large increases in demands on electric grids due to electric vehicle charging is a substantial problem that requires addressing.

    (a) It is worth noting that although the zero-emission vehicles themselves do not emit GHGs during usage, the energy to charge the vehicle may emit GHGs. Particularly in any place that has yet to transition to zero-emission electricity generation. Different locations have different GHG emission profiles, which determine how effective transitioning is relative to other locations. An important aspect of GHG emission profiles is the concept referred to as carbon intensity (the mass of GHG emissions in carbon dioxide equivalent per amount of energy produced – often grams of carbon dioxide per kilowatt-hour). For reference:

        i. Carbon intensity for the U.S.: 386 grams of CO2e/kWh (0.85 pounds of CO2e/kWh) in 2020[52]

        ii. Carbon intensity for Canada: 120 grams of CO2e/kWh in 2019[8]

iii. Carbon intensity for Alberta: 620 grams of CO2e/kWh in 2019[8]

iv. Carbon intensity for Ontario: 30 grams of CO2e/kWh in 2019[8]

(b) Carbon intensity in the context of electric vehicle transition can be conceptualized using back-of-the-envelope calculations as follows:

i. As of 2022, according to Natural Resources Canada, the most energy efficient internal combustion vehicle on record is the 2021 Hyundai IONIQ Blue (Hybrid)[39]:

A. Emissions (mass of carbon dioxide per distance): 94 g CO_2/km

B. Fuel Consumption (fuel volume per distance): {City: 4 L/100 km, Highway: 3.9 L/100 km}

ii. As of 2022, according to Natural Resources Canada, one of the most energy efficient electric vehicles on record is the 2021 Hyundai IONIQ Electric (Tied for 1st for city energy efficiency and tied for 2nd for highway energy efficiency)[38]:

A. Emissions (mass of carbon dioxide per distance): 0 g CO_2/km

B. Fuel Consumption (fuel volume per distance equivalent): {City: 1.6 L_e /100 km, Highway: 1.9 L_e /100 km}

C. Fuel Consumption (energy per distance): {City: 14.5 kWh/100km, Highway: 17.4 kWh/100km}

iii. The following **carbon intensity bound** calculation uses the specifications of the two vehicles (from 1bi and 1bii) above since their body styles are virtually identical (more precisely, the drag coefficients):

A. The mass of carbon dioxide per distance (1biA) divided by the energy per distance (1biiC {Highway}): $\left(\frac{94}{1} \cdot \frac{\text{g CO}_2}{\text{km}}\right) \cdot \left(\frac{100}{17.4} \cdot \frac{\text{km}}{\text{kWh}}\right) \approx \frac{540}{1} \cdot \frac{\text{g CO}_2}{\text{kWh}}$

B. The reasoning for dividing by the electric vehicle (instead of internal combustion engine vehicle) energy per distance is to provide a lower bound of carbon intensity. The calculated carbon intensity is larger if the internal combustion engine vehicle is used. The electric vehicle efficiency (1biiB) is approximately double that of the hybrid version of the vehicle (1biB). Although using the hybrid efficiency value would be a more accurate representation of carbon intensity for that particular vehicle, it would not be a guaranteed lower bound for all vehicles on Canadian records.

C. All internal combustion engine vehicles (including non-plug-in hybrids) (in the records of Natural Resources Canada) have carbon intensity values greater than 540 g CO_2/kWh. This bound is below the carbon intensity of electricity generation in Alberta; however, the actual carbon intensity values of vehicles are above that of Alberta electricity generation.

D. In the case of larger vehicles (referred to as light trucks such as Sport Utility Vehicles (SUV) and Pickup Trucks), the lower bound is calculated using the same method from 1iiiA, but using the emissions from the most efficient SUV on record (2021 Ford Escape Hybrid: 136 g CO_2/km[37]), the resulting carbon intensity bound is approximately 782 g CO_2/kWh.

(c) The calculation results from 1b are meant to compare the emissions of electricity generation and internal combustion engines. In particular, the zero-emission potential of electric vehicles (energy usage and energy production source) versus the non-zero-emission limitations of internal combustion engines (energy usage and energy production source). In other words, electric vehicles can approach a constant amount of emission (any emissions during manufacturing) for the life cycle of the vehicle. In contrast, the emissions of internal combustion engine vehicles will always be variable to the time and intensity of usage for the life cycle of the vehicle.

2. As an example of the robustness of the Canadian plan to tackle climate change problems, there is a specific program in place to research the problem of optimizing electric grids as the Canadian vehicle market transitions to 100% zero emissions.

   (a) “As transportation is one of the main GHG contributors, electrification can help the sector move to lower emission alternatives. Mass transportation electrification, however, is a new research field where there is a lack of knowledge on the impact that large electric vehicle (EV) fleets will have on the transmission and distribution electricity grids, and on the required charging infrastructure” ("Optimizing electric grids and charging infrastructure for mass electric vehicles penetration," 2020)[42].

3.2 Introduction: Specific Problem: Transformers

1. The parts of electric grids that are particularly vulnerable to the large increases in demands from electric vehicles are transformers. This vulnerability is due to many factors, but not the least of, which is the way that transformers have been sized prior to the development of climate change plans.

2. Typically, transformers are sized according to the maximum anticipated demand for the life of the equipment due to the substantial cost of installing and uninstalling equipment. However, this kind of installation procedure has been relatively sufficient as the changes in most residential demands are well within the capacity of the transformers.

3. Unfortunately, "higher demand for cooling owing to rising air temperatures can exacerbate the burden" (Chattopadhyay, Bazilian, & Chattopadhyay, 2019)[10] on electric grids. The increase in cooling demands and other factors "can lead to power outages, increased electricity prices, and increased maintenance and capital costs – along with damaging economic, environmental and public health consequences" (Chattopadhyay, Bazilian, & Chattopadhyay, 2019)[10].

4. Specifically, the hot summer days require cooling systems to operate at higher intensities and for longer periods of time. The combination of these demands stresses the capacity of transformers.

5. Although the risks to electric services, such as power outages due to heat waves, are typically associated with warmer climates such as locations in the United States, the Canadian government has been advised to take precautionary steps to "address the growing risk of unprecedented peaks in summer cooling demand" (Canadian Electricity Association, 2016, p. 42)[9].

6. The transition to electric vehicles will further burden the electric grids already stressed by the consequences of climate change.

7. For instance, in a study of this problem in 2016, it was found that "[o]ut of the 118 transformers in the simulated area [of Ajax, Ontario, Canada], the largest amount with EV connected to was 48 (Fig. 2), among those 48 there was an average load increase of 27.2 kW. Out of the 48 transformers, 12.5% were overloaded (Fig. 3). The results represent the highest peak of

transformer loads within the 24-hour simulation" (Jalali, Hills, El-Khatib, Pazzi, & Hoornweg, 2016)[28].

(a) A simple solution would be to upgrade all transformers with the capacity that would cover even the largest increase in energy demand; however, this would be one of the most expensive possible solutions.

(b) A less expensive solution would be to upgrade transformers as they begin to reach maximum capacity; however, by design, this solution allows for equipment failure. Although there is an acceptable amount of overloading for short periods of time, extended overloading of transformers puts exceptional wear. These factors reduce the value of transformers that otherwise could have been reinstalled elsewhere. Further, extended overloading of equipment can also be a safety hazard.

(c) A more cost-effective solution to this problem may require a significantly high resolution of electric grid data in order to predict demands accurately and upgrade accordingly.

3.3 Introduction: Specific Problem: Upgrades

1. With the context of the above and the potentially high costs associated with upgrading electrical transformers, the question of this thesis is: what is the most optimal way of performing upgrades throughout the transition to 100% electric vehicles? This question is answered by developing a model that manages this intractable problem. In general, finding **exact** solutions to instances of this problem is intractable. Conversely, finding approximate solutions to instances of this problem is tractable – assuming that an approximation method exists.

    (a) The moments when upgrades can occur are referred to as upgrade periods.

    (b) The scale of this problem grows exponentially with the increase in the number of locations of electrical transformers (nodes) and the number of upgrade periods.

    (c) The number of upgrade schedule combinations between all nodes is the result of this exponential growth.

    (d) The model is the discrete optimization of upgrade scheduling.

4 Introduction: Hypothesis

4.1 Introduction: Hypothesis: Overview

1. In this thesis, the hypothesis evaluation yields a limited result as this thesis is only the initial study of the model. The model is a construction of an optimizer that intends to minimize total cost and satisfy budget constraints. The methods of analysis only definitively check for the existence of an optimization model that can produce results that converge to an exact solution. A stronger result would be the mathematical proof of model performance under all input conditions.

    (a) The term **input conditions** in this **Introduction: Hypothesis** section refers to the input data to the model, which includes structures such as the cost curves of equipment and energy demands. The input details are explained in more detail in the **Model: Technical Layer** section.

4.2 Introduction: Hypothesis: Null Hypothesis

1. The null hypothesis of this study is that there do not exist input conditions where the model presented in this thesis can produce approximately optimal schedule combinations, which converge to an exact solution as the resolution of the approximation increases.

4.3 Introduction: Hypothesis: Alternative Hypothesis

1. The alternative hypothesis of this study is that there do exist input conditions where the model presented in this thesis can produce approximately optimal schedule combinations, which converge to an exact solution as the resolution of the approximation increases.

4.4 Introduction: Hypothesis: Evaluation

1. The model performance is evaluated under various conditions to determine if the null hypothesis is rejected. If the results under these conditions demonstrate a trend of monotonically optimal solutions, then the null hypothesis is rejected. More rigorously, if it is mathematically provable that the problem data structure demonstrates the above-mentioned trend, the alternative hypothesis is accepted.

5 Introduction: Summary

1. The key takeaways from the thesis introduction:

   (a) The general problem of climate change and the potential dangers have been known for half a century. Sufficient steps have yet to be taken to avoid the worst outcomes of climate change. There are a plethora of consequences for this inaction in the present moment, the near future.

   (b) The sources of GHG emissions and their atmospheric concentrations are well studied. This understanding sets the stage for the fight against climate change in the world and, more specifically, in Canada.

   (c) Canada has a large system of programs and policies that are in place to address climate change. At the highest level, Canada has set goals that will satisfy and potentially exceed international climate agreements. These programs and policies include the transition away from fossil fuels -- specifically the transition to a 100% zero-emission vehicle market.

   (d) The problem with transitioning to 100% zero-emission vehicles extends beyond the supply and demand of the market. More exactly, the increase in electrical demand from electric vehicles will be beyond the capacity of various parts of most electric grids around Canada (and the world).

   (e) This thesis is focused on addressing the part of the electric grid that is particularly vulnerable to such large increases in demands – the electric transformer. Specifically, the purpose is to study the optimization of upgrade scheduling of electric transformers in support of electric vehicle transition. The model of this thesis is, in a sense, a method for performing the optimization of upgrade scheduling.

   (f) The null hypothesis of this study is that there does not exist a circumstance where the model will produce results that converge to the exact solution. The alternative hypothesis is that there does exist such a circumstance. The hypothesis evaluation is core to the thesis; however, this study goes beyond this evaluation for a deeper understanding of the optimization problem as a whole.

Part II
Model

Part Contents:

1. Literature Review
2. Overview
3. Descriptive Layer
4. Technical Layer
5. Rigorous Layer
6. Summary

"Your words, O Hares! are good;, but they lack both claws and teeth such as we have."

\- Aesop, *The Hares and the Lions*[1]

1 Model: Literature Review

1. The general problem of resource management, such as upgrade scheduling, is part of many fields of study and practice. The fields that work on resource management problems include (but are not limited to) mathematical optimization, computational theory, and operational research. Although the motivations and goals of each field may differ, there is a significant overlap in terms of algorithms used to solve these problems.

2. It is not feasible to list all the optimizations, so the following are only some of the most common methods:

(a) Monte Carlo:

  i. Although there are many variations, the core of every Monte Carlo method is based on random sampling to produce a result. In the case of discrete optimization, feasible solutions are found by iteratively taking random samples.

  ii. Regarding the origins of the Monte Carlo method, according to a student and collaborator of Enrico Fermi (Emilio Segrk)," Fermi had invented, but of course not named, the present Monte Carlo method when he was studying the moderation of neutrons in Rome" (Metropolis, 1987, p. 4)[33] in the early 1930s. Despite the method existing for nearly a century, its effectiveness has not diminished in comparison to other methods, especially in problems that are exceptionally intractable or not well understood.

  iii. The effectiveness of the Monte Carlo is due to its generalized and flexible attributes. These attributes enable the method to be accelerated using the context of a problem (also known as heuristics) and work in tandem with other methods.

  iv. Recent research into industrial optimization implements the Monte Carlo method that is further "optimized by the genetic algorithm, which is one of the most important metaheuristic search algorithms" (Ma & Lv, 2019, p. 14)[32]. An application for the methods in this research "solves an optimization problem in the manufacturing industry," which is similar to the thesis problem except that their research uses multiple objectives.

(b) Genetic algorithms:

  i. The core of genetic algorithms is based on principles from biological genetics, such as the construction of genotypes and iterative selection (evolution). How the genotypes are constructed can depend on the context of the problem it is solving. Once the initial genotypes, or prospective solutions, are constructed, the fitness of the genotypes is evaluated.

    A. An example of prospective solutions in the context of scheduling optimization can be a selection of potential schedules.

    B. The meaning of the term fitness depends on the context of the problem. In scheduling optimization, the term **fitness** refers to how optimal the genotype results are.

ii. Once the fitness levels of the genotypes are evaluated, the top performers are selected for breeding. The process continues iteratively until some condition is achieved.

iii. In recent research, a genetic algorithm is used "to generate diverse sets of scenarios for two-stage optimization problems under uncertainty. It shows the potential of population-based algorithms, namely genetic algorithms, to generate sets of scenarios when the fitness function is designed to favour the diversity of the population" (Oliveira, Carravilla, & Oliveira, 2022, p. 1141).

iv. From the research paper referenced during the discussion of Monte Carlo methods, the results show that their implemented genetic algorithm "speeds up the Monte Carlo simulation dramatically" (Ma & Lv, 2019, p.)[32]. This reference demonstrates the relevance of genetic algorithms in modern optimization methods and their compatibility with other methods.

(c) Dynamic programming:

i. The concept of dynamic programming relies on the ability to split the problem into subproblems that can be optimized independently and avoid computing results more than once. A key example demonstrating the effectiveness of dynamic programming is its ability to reduce the computation steps when recursively computing Fibonacci numbers.

ii. The open source project OR-Tools by Google is the international leader in optimization methods. Specifically, Google OR-Tools has won first place at the international constraint programming competition by MiniZinc since 2013[19].

iii. A common problem that uses dynamic programming is the knapsack problem. In general, the problem objective is to optimize the value of its contents without exceeding its capacity.

iv. According to Google OR-Tools documentation, "there are several ways to solve knapsack problems. One of the most efficient is based on dynamic programming (mainly when weights, profits and dimensions are small, and the algorithm runs in pseudo polynomial time). Unfortunately, when adding conflict constraints the problem becomes strongly NP-hard, i.e. there is no pseudo-polynomial algorithm to solve it.

That's the reason why the most of the following code is based on branch and bound search" (Google OR-Tools, 2020)[20]. The branch and bound method is often used in integer programming algorithms. The point of this reference is to show that the concept of dynamic programming is still in use in the most efficient optimization tools in the world.

(d) Integer programming:

i. A core concept for solving integer programming problems is known as branch-and-bound.

A. The first concept, branching, refers to the expression of the problem in the form of a tree (specifically a polytree), which saves computational memory or instructions by eliminating the need to express the combinatorial search space of such a problem. For example, the navigation of a binary tree can reduce the computational complexity of expressing all binary permutations independently by as much as a linear order in proportion to the input size (the depth of the tree).

B. The second concept, bounding, refers to the iterative pruning of branches of the tree. For example, consider a binary-tree problem where all individual costs and constraints are greater than or equal to zero. If the first part of a branch exceeds some constraints, the algorithm does not need to evaluate any downstream branches.

ii. As mentioned in the discussion of dynamic programming, branch-and-bound can be implemented when methods, such as dynamic programming, cannot solve specific problems efficiently. According to Google OR-Tools guidance regarding mixed integer programming (MIP) problems (integer programming is a special case of MIP), their mathematical programming (MP) solver (MIP is a special case of MP), MPSolver is a collection of several MIP solvers, "which use standard branch-and-bound techniques" (Google OR-Tools, 2021)[20]. The point of this reference is to show that the concept of branch-and-bound to solve integer programming problems is still in use in the most efficient optimization tools in the world.

3. As discussed in the **Appendix: Computational Complexity** section, many optimization problems are similar. They are similar in the sense that they can be translated from the

original form into the form of another problem. This translation process is sometimes referred to as a reduction (specifically, Karp reduction).

4. Regarding the methods discussed above, **integer programming** is notably similar to the model presented in this thesis. However, it differs in dimensionality and has additional restrictions that do not commonly occur in practice.

   (a) Consider the following integer programming problem:

$$\text{Minimize:} \quad \sum_{\forall n} \left(\sum_{\forall m} a_{mn} \right) x_n \tag{1}$$

Subject to:

$$\left(\left(\sum_{\forall n} a_{mn} x_n \right) \leq b_m \right)_{\forall m} \tag{2}$$

Where:

a_{mn} is a cost associated with x_n

b_m is a budget constraint

x_n is an equipment decision

$a_{mn} \geq 0$

$b_m > 0$

$x_n \in [0, 1] \in \mathbb{Z}$

$m \in [1, M] \in \mathbb{Z}$

$n \in [1, N] \in \mathbb{Z}$

   (b) The thesis problem can be expressed in terms of the above integer programming form, which allows for any of the best-established optimization methods to be applied. However, the translation of the thesis problem into the above form requires an exponential number of steps in proportion to the input size. The number of steps involved in translation is one of the justifications for the model presented in this thesis and is explained in more detail in the following sections.

(c) In discrete optimization, especially operational research, the methods implemented typically do not consist of a single optimization concept; instead, they can consist of multiple concepts that can occur in stages or change depending on the parameters of the problem. According to the literature survey, the component concepts of the optimization model presented in this thesis are not unique within the context of current methods used in study and practice; however, the overall implementation of the optimization model as applied to upgrade scheduling appears to be original.

2 Model: Overview

1. The optimization of budgets is a common problem in civilization and even in nature. It is common in economics and less traditional settings such as individual resource management. Although problems regarding the optimization of budgets come in many forms, they can be converted into a universal form known as graph problems. Refer to **Appendix: Model: Overview** and **Appendix: Computational Complexity** for a more detailed explanation of graph problems, optimization, and computational complexity as it relates to this thesis.

2. The purpose of the model is to transform the problem in a way that makes the problem computationally manageable. This transformation is performed using an optimization method that was incepted by Professor Jane Breen in 2019, the idea of nesting graphs (or nesting sub-graph, to be more precise) approximations. For example:

   (a) The following figure is an arbitrary instance of data to demonstrate the concept of nesting graphs (each data point representing a node):

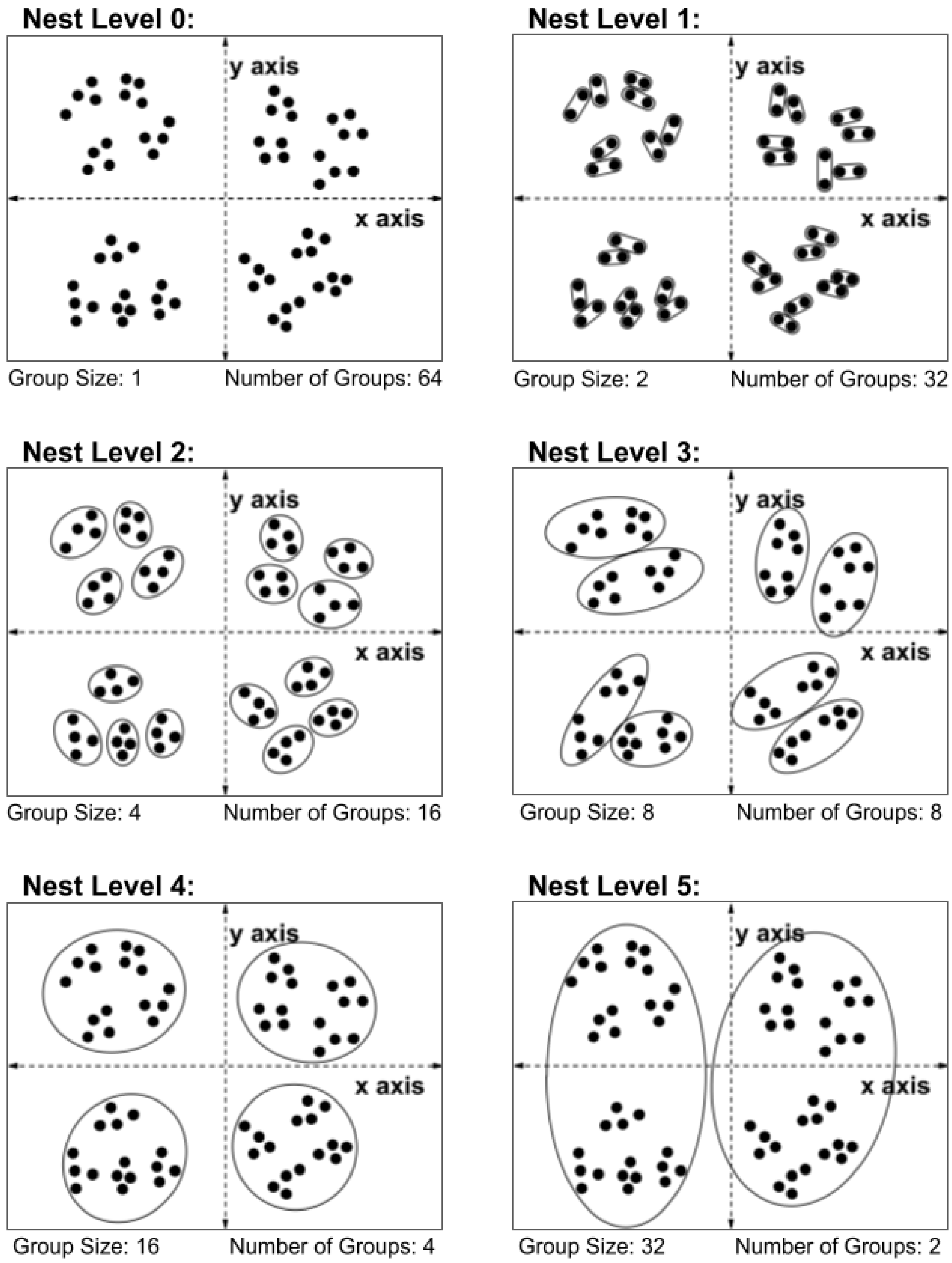

Figure 5:
The above collection of images demonstrates the various levels of nesting, which correlates with the approximation resolution. The highest resolution corresponds to the largest number of groups. Nesting Level 5 has the lowest resolution with only two groups. Nesting Level 0 has the highest resolution with 64 groups.

3. In Figure 5:

    (a) **Nest Level 5** has the lowest resolution, with only two groups representing all 64 data points.

    (b) **Nest Level 4** has the second lower resolution, with four groups to represent all 64 data points.

    (c) The pattern continues for **Nest Level 3** and further.

    (d) As has been stated, a decrease in nest level corresponds to an increase in resolution; however, the higher resolution groups are contained in the groups from the lower resolution groups – hence the term **nesting**.

4. Although the process of nesting is not strictly necessary to approximate the data in the problem, the nesting guarantees that as the resolution increases, their respective results become more accurate (more optimal). Non-nested approximations cannot guarantee convergence.

    (a) A reason convergence is important is that it provides the model ability to work in tandem with other optimization methods. Certain optimization methods do not provide information as to **how** optimal their results are. The model presented in this thesis is capable of providing such information – namely, the equivalent level of nesting. As significant as this concept is to the model, it is not explored in any significant detail as the main focus of this thesis is to establish the basic validity of the model performance.

    (b) The property of convergence is explained in more detail in the sections following this section, **Model: Overview**.

5. In Figure 5, the dataset of nodes has only two dimensions. In the context of the model, the number of dimensions of the dataset of nodes is equal to the number of upgrade periods. The number of upgrade periods is the maximum number of upgrades that can occur in the window of time being considered.

    (a) For example: For some arbitrary instance of the upgrade problem: If any node is capable of being upgraded once per year over a 20-year window of time, then the maximum number of upgrade periods is 20. In this case, the upgrade schedule dataset has 20 dimensions. Each data point is a vector (list) of upgrade costs for each upgrade period.

(b) The details of the model are explained in more detail later in the thesis. However, at this moment in the model overview, the important point to understand is that the optimization method relies on (nested) clusters of schedule costs to approximate groups of nodes. In other words, although the model uses more dimensions than the two from Figure 5, the basic premise remains unchanged.

6. The model presented in this thesis is a three-stage optimization method that relies on nesting approximations of upgrade schedule costs and a modified brute force optimizer. This three-stage optimization method is referred to as the Breen Optimization Method (BOM).

7. In brief, the three distinct stages of BOM:

   (a) **Stage One:** To optimize the individual node schedules (converting the input demands into potential upgrades).

   (b) **Stage Two:** To simplify the node schedules through approximation to find an approximately optimal solution.

   (c) **Stage Three:** Select the most optimal schedules that fit the approximated optimal solution.

8. In reference to electric grids and upgrades:

   (a) **Stage One:** With the input, such as {all of the equipment for upgrades and the projection of transformer demands}, all possible upgrade schedules are calculated for each transformer location. These calculated schedules are then outputted to Stage Two.

   (b) **Stage Two:** The purpose of this model is to find an optimal combination of upgrades, but the output (exact schedules) from Stage One is too large to be optimized (by brute-forcing all combinations), so this stage (Stage Two) simplifies Stage Two output by grouping similar schedules together. These simplified schedules are then outputted to Stage Three.

   (c) **Stage Three:** With the simplified schedules from Stage Two (approximate schedules), all the possible combinations are calculated. The most optimal schedule combination that satisfies the budget constraints is converted from the approximation form to the exact form. The approximate form does not represent actual upgrade schedules. Similar to how the budgets constrain the entire problem (all the nodes), the approximate solution constrains individual nodes.

i. With respect to the combination lock example (from 6aiiC above), the approximate solution determines the maximum settings for each individual dial to ensure that none of the combined schedule costs go above the budget constraints across all upgrade periods.

9. The model is presented in three layers:

    (a) **Descriptive Layer:** This first layer contains descriptive details of the model. It describes how the main components of the model apply to the problem of optimizing upgrade scheduling.
    (b) **Technical Layer:** This second layer contains technical details of the model. It serves the purpose of explaining and justifying the mechanical structure of the model.
    (c) **Rigorous Layer:** This third layer contains the rigorous details of the model. It serves to mathematically prove the model output approaches the exact optimal combination of schedules.

3 Model: Descriptive Layer

1. As discussed briefly in the **Model: Overview** section, the main problem that the model in this thesis is designed to solve is determining the most optimal allocation of funds for each transformer (more generally, for each node) to ensure electrical capacity during significant increases in demand while satisfying budget constraints.

2. Before determining funding allocation under budget constraints, all the possible upgrade schedules for each individual node must be constructed, and the most optimal equipment upgrades need to be calculated. Although the process of determining when upgrades are needed determines the minimum upgrades for each node across all upgrade periods, it does not indicate, which equipment is the most optimal (most cost-effective) upgrade. Determining the most optimal equipment for each schedule for a node is an optimization that is independent of all the other nodes, so this stage does not require the use of budget constraints.

3. Once the optimal schedules are calculated for each node, the optimal combination of these schedules could theoretically be calculated with unlimited time and computational resources.

However, realistically, the number of combinations is intractable for even a small neighbourhood grid (see example 6a from the **Model: Overview** section). The intractability of the problem at this point necessitates the use of an approximation.

4. There are many ways to approximate intractable problems to make them tractable. The approximation stage (State Two) uses the technique of grouping nodes with similar schedule costs together. In a sense, a group approximation is a representative node, which takes the maximum costs from all the nodes. The exact process is explained in more detail, but effectively, the group approximations shrink the problem down to a computationally manageable size while ensuring that the group approximations do not underrepresent their members.

5. So instead of using brute force to evaluate all the upgrade schedule combinations, brute force is used to evaluate all the **approximate** upgrade schedule combinations. Among all of the approximate upgrade schedule combinations, which do not go over the budget constraints, the combination with the lowest overall cost is selected as the (approximate) optimal result.

6. Finally, once the approximate optimal result is calculated, each approximation group is allocated a maximum amount of funding (per upgrade period) for each of their member nodes. For each node, the schedule with the lowest overall cost, which does not go beyond any of the maximum funding per upgrade period, is selected as the optimal schedule. The collection of optimal schedules from each node is the optimal result returned by the model.

7. Each stage of the model has three distinct components:

   (a) **Input:** These are the values used to initialize their respective stage of the model.

   (b) **Structure:** The structure of each stage is composed of attributes and methods that are used to process input into {intermediate values or stage output}.

   (c) **Output:** The output is the data that is returned once the methods conclude their processes.

8. Although the description of the model is abstract, many of the descriptions are described in relation to a toy electrical grid example.

   (a) The toy electric grid is a toy example in the sense that it only includes as much detail as necessary to demonstrate the functionality of the model. For instance, there are plans for electricity providers to allow EV owners to put energy back onto the grid. In this type of scenario for a real-world application, this concept would need to be included in the model. Although it is not included in this thesis, the model can be modified to accommodate such an extension.

3.1 Model: Descriptive Layer: Stage One

1. The essence of the first stage is to process each set of node demands to create optimized upgrade schedules.
   (a) The nodes are processed independently. This means the calculation of one node is irrespective of any other node calculations.
   (b) Nodes are processed dependently in stage two. This is when the budget constraints are used since the cost of all nodes is summed.
   (c) The most optimal set of equipment for each conceivable upgrade schedule is computed by brute force.
2. In the case of an electrical grid, the input is the projected demands of transformers over some chosen amount of time (upgrade window).
3. The first stage of the model can fit any demand data and an equipment list -- as long as there exists equipment with enough capacity for all demands.

3.1.1 Model: Descriptive Layer: Stage One: Input

1. The core input for stage one is the demand time series data. In the case of the toy example, the input would be the projected demand data for each transformer.
2. The auxiliary inputs for stage one are:
   (a) The number of periods the time series will be split into.
   (b) The equipment catalogue and inventory.
   (c) The cost of uninstalling and installing each piece of equipment.

3.1.1.1 Grid Demands

1. In the toy electrical grid context, it is a dataset of time series of energy demands similar to 6, except the input is over 20 years instead of seven days.

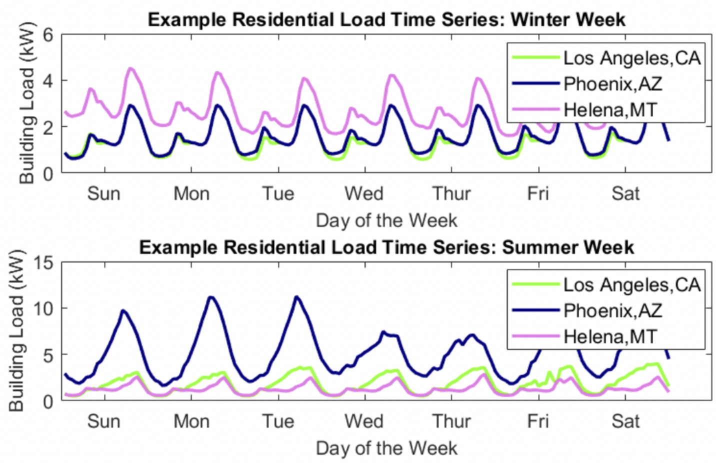

Figure 6:
The figure above contains time series plots of residential energy demand over the span of one week. Each plot shows the time series demands from Los Angeles (California), Phoenix (Arizona), and Helena (Montana) for one week in winter (top plot) and for one week in summer (bottom plot).

2. In Figure 6, the property worth noting in the above figure (Li, Yeo, Bornsheuer, & Overbye, 2021, p. 2)[30] is the cycle of peaks and troughs. This property is particularly important when considering the addition of EV charging because of the potential for overloading transformers during peak hours. The specific amount of energy in the plot is not of any particular importance.

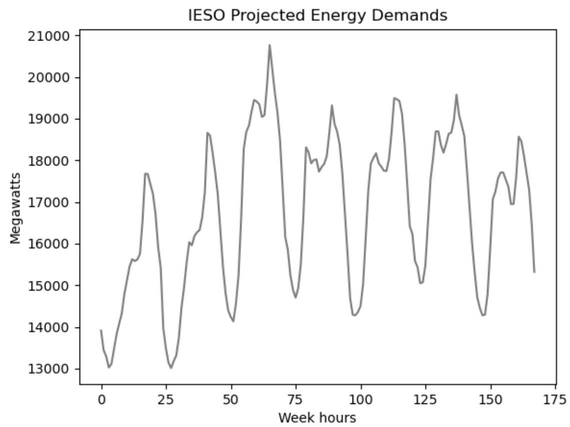

Figure 7:
This is a plot of an excerpt from the forecasted electrical energy demands for Ontario from 2020 to 2040 from the Independent Electricity System Operator (IESO) predictive model (IESO, 2021)[26]. The excerpt covers the first week of January 2020.

1. The scale of the Independent Electricity System Operator (IESO) forecast for Ontario (vertical axis in MWs) is significantly greater than the scale in the residential demands from Figure 6 (vertical axis in kWs); however, the cyclical nature of demand is still present. The scales are significantly different because Figure 7 is the accumulative demands from all of Ontario, whereas Figure 6. Again, in the context of the toy electrical grid, the important property is the cyclical nature of demand as opposed to the specific quantity of demands.
   (a) This property is important simply for the fact that the synthetic data used to test the model is derived from projected demand for all of Ontario, but normalized. Once the data is normalized, the original scale of the energy demand no longer exists, and all that remains is the cyclical nature.
   (b) How the synthetic data is constructed is discussed in more detail in **Part III Method**.
2. Although the scale of Figure 7 is in thousands of megawatts for the purpose of constructing a toy example for the model, the data is sufficient for deriving approximate residential energy demands. This being said, it is not meant to be an accurate representation of any real-world energy demands, but rather to include an element of realism.
   (a) The way that the toy electrical grid demands are derived by using the ratio between total

energy consumption in Ontario and average energy consumption per household in Ontario ("Household energy consumption, Canada and provinces," 2017)[24].

  i. To remark further on how this is not representative of real energy demands, the IESO projected demands include demands from residential as well as other customers such as commercial.

(b) Once the ratio is applied to scale the IESO demands, it is used as the average virtual household. Virtual households are then created by sampling from the average with noise and summed together in their respective virtual transformers. These virtual transformers represent the nodes for the model.

3.1.1.2 Number of Periods

1. The number of upgrade periods in the upgrade window.

2. For example, with the following synthetic dataset that has an upgrade window from 2020 to 2040:

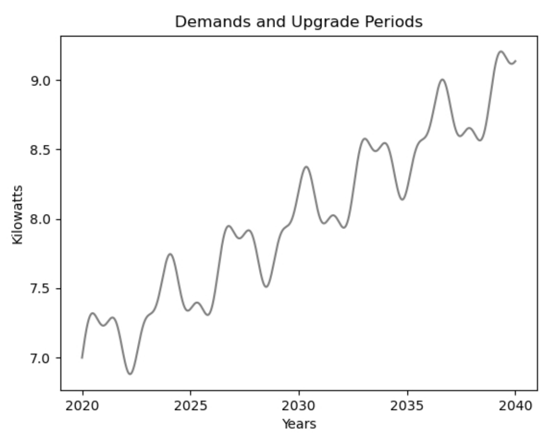

Figure 8:
This plot is simply the combination of two sine waves and does not represent any specific pattern of demands.

1. If the number of upgrade periods is four, then the synthetic dataset would be split as follows:

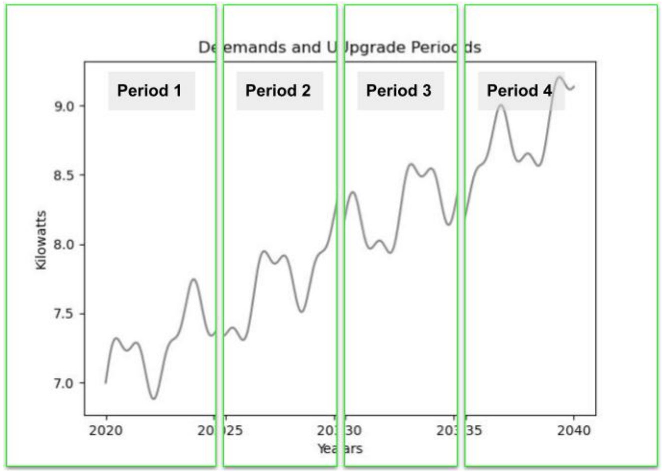

Figure 9:
This plot shows how the upgrade periods modify the input demand data.

1. This does not necessarily mean that the periods need to be evenly spaced. In fact, it may be more optimal to strategically divide the upgrade periods, but this is not explored in the thesis for the sake of simplicity.

3.1.1.3 Equipment Catalogue

1. The following table represents an equipment catalogue as an input data structure to the model.

    (a) The 'Efficiency' is a function that approximates the efficiency of the electrical transformer at various loads within its capacity.

    (b) The 'Value' is the cost of purchasing the equipment.

    (c) The 'Install' is the cost of labour, machinery, and supporting materials to install the equipment.

    (d) The 'Uninstall' is similar to the 'Cost to Install', but typically costs less. For instance, supporting materials are not required during an uninstall.

Rating	Efficiency	Value	Install	Uninstall
50 kVA	f_{50}(load)	\$6.0K	\$1.5K	\$1.1K
55 kVA	f_{55}(load)	\$6.5K	\$1.6K	\$1.2K
60 kVA	f_{60}(load)	\$7.0K	\$1.7K	\$1.3K
65 kVA	f_{65}(load)	\$7.5K	\$1.8K	\$1.4K
70 kVA	f_{70}(load)	\$8.0K	\$1.9K	\$1.5K
⋮	⋮	⋮	⋮	⋮

Table 1:
This table does not represent any real costs and is only for conveying the equipment catalogue data structure.

1. There is, of course, more to equipment catalogues in reality. For instance, the cost to replace a pad-mounted transformer might be significantly less than the cost of replacing a pole-mounted transformer. It also depends on the type of location. Also, the cost of a transformer of a particular rating can vary greatly from one brand to the next and from materials used to construct the equipment, such as aluminum windings vs copper windings. In this thesis, the equipment catalogue is kept simple since the construction of a realistic equipment catalogue is not the point of this study.

3.1.1.4 Equipment Inventory

1. The equipment inventory is fairly similar to the equipment catalogue, with slight differences. Although they are similar, the equipment inventory represents the equipment that is already in possession of an electrical energy provider (installed or in storage), and the equipment catalogue represents the equipment that is not yet in possession of an electrical energy provider. As an analogy, a {dresser full of clothing} is to an {equipment inventory} as a {clothing catalogue} is to an {equipment catalogue}.

   (a) The 'Efficiency' in a real-world application would decrease over time in service, but it is not an attribute that is included.

   (b) The 'Value' in a real-world application would decrease over time in service, but it is not an attribute that is included.

   (c) The 'Install' is the cost of labour, machinery, and supporting materials to install the equipment.

(d) The 'Uninstall' is similar to the 'Cost to Install', but typically costs less. For instance, supporting materials are not required during an uninstall.

(e) The 'Location' is the location of the equipment on the electrical grid, such as a node where it is installed or where the equipment is stored.

Equipment	Efficiency	Value	Install	Uninstall	Location
55 kVA	f_{55}(load)	\$6.5K	\$1.6K	\$1.2K	Node 0
55 kVA	f_{55}(load)	\$6.5K	\$1.9K	\$1.7K	Node 1
65 kVA	f_{65}(load)	\$7.5K	\$0.8K	\$1.5K	Node 2
65 kVA	f_{65}(load)	\$7.5K	\$2.8K	\$1.8K	Node 3
65 kVA	f_{65}(load)	\$7.5K	\$1.8K	\$1.4K	Node 4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100kVA	f_{100}(load)	\$10.5K	\$4.2K	\$2.7K	Storage 0

Table 2:
This table does not represent any real costs and is only for conveying the equipment inventory data structure.

1. Similar to the equipment catalogue, the equipment inventory is also a simplification.

3.1.2 Model: Descriptive Layer: Stage One: Structure

1. The core methods of stage one are to convert the input into a set of upgrade schedules for each node and to optimize those schedules.

2. The following are two main sets of algorithms used in this stage of optimization.

3.1.2.1 Create Node Upgrade Schedules

1. The number of periods can be used to calculate the number of schedules that nodes will have. Alternatively, the number of schedules can also be used to calculate the number of periods. This is important for scalability reasons since it is likely that a real-world application would start with the amount of computing resources that are available and then determine the parameters, such as the number of schedules. The number of schedules that can be computed by the model also determines how many upgrade periods can be computed.

(a) An increase in the number of periods equals an exponential increase in the number of schedules.

(b) Suppose the amount of computational resources is known. In that case, the number of schedules can be derived using the time complexity of the model.

2. For the sake of simplicity, the number of periods is used to create upgrade schedules.

3. There are several ways of expressing all possible upgrade schedules. The number of possible schedules can be determined as follows:

$$n_{\mathrm{p}} := \text{Number of periods}$$
$$n_{\mathrm{s}} := \text{Number of schedules}$$

$$n_{\mathrm{s}} = 2^{n_p - 1} \tag{3}$$

4. The number of possible schedules is two to the power of the number of periods minus one because the first period (and onward) must always be evaluated if an upgrade is needed. Suppose the first period is not evaluated for an upgrade. In that case, the default behaviour is to not upgrade equipment during that period. If the equipment in the first period has insufficient capacity, grid failure becomes imminent.

(a) To clarify the reasoning for the first period always being evaluated, consider the following analogy of planning a seven-day long road trip in an EV (equivalent to seven upgrade periods):

i. If the goal is to spend the least amount of money on charging, but there is a set budget for each day, a charging schedule {for, which stations to stop at and how much it costs at each station} must be evaluated.

ii. The status of the EV must always be evaluated for the first leg of the trip, regardless of whether the first leg of the trip lasts more than one day. The first leg always includes the first day, so the first day is always evaluated.

iii. The first day is not receiving preferential treatment; it is simply that the first leg is from the first day onward.

iv. Assuming the EV battery has enough capacity to drive the entire seven days without recharging, if a schedule is to drive from Monday morning to the next Sunday evening and only charge on Monday, then that Monday charge must be sufficient for that entire week. If the charge is sufficient for the entire week, there is no need to evaluate if the EV needs to be charged on any other day.

(b) Equivalently, for electrical transformers, if a schedule calls for an upgrade to cover the demand over the course of an entire 20-year long (one upgrade period per year) upgrade window, the upgrade is performed during the first upgrade period. Once that upgrade is installed, the transformer has the capacity for the entire 20 years, so no other upgrade period needs to be evaluated. Similarly, if the schedule calls for an upgrade to cover the first 10 years, then a second upgrade to cover the remaining 10 years, the upgrade on the first year must have the capacity for the first 10 years, so no other years before the 10th year require an upgrade. Further, the second upgrade must cover the remaining 10 years, which means that the upgrade is performed at the 10th year, and no other years require upgrades since the capacity is sufficient for those years.

(c) In Table 3 below, the schedules are represented in binary. A **1** indicates an upgrade; however, that upgrade must cover the period in which the **1** occurs and also cover the consecutive periods with **0** until another **1** occurs (or until the end of the upgrade window). An equivalent schedule representation is presented in Table 4 without binary. Similar to the road trip analogy, a leg of a trip is equivalent to a schedule slice.

5. Every schedule is a combination of components known as slices. A slice size can be a minimum of one period or a maximum of all periods.

6. For example, an instance of four periods:

Schedule	Period 1	Period 2	Period 3	Period 4
1	1	0	0	0
2	1	0	0	1
3	1	0	1	0
4	1	0	1	1
5	1	1	0	0
6	1	1	0	1
7	1	1	1	0
8	1	1	1	1

Table 3:
This table demonstrates the one-hot or binary representation of all possible schedules with four upgrade periods.

7. To explain Table 3 in more detail:

   (a) For schedule 1, the binary is **1000**. What this means is that the first (and only) slice includes periods {1, 2, 3, 4}. Every schedule slice is evaluated for an upgrade, so schedule 1 must cover the entire upgrade window.

   (b) For schedule 2, there are two slices:

      i. Slice 1: Periods {1, 2, 3}

      ii. Slice 2: Period {4}

   (c) For schedule 3, there are two slices:

      i. Slice 1: Periods {1, 2}

      ii. Slice 2: Period {3, 4}

   (d) For schedule 4, there are three slices:

      i. Slice 1: Periods {1, 2}

      ii. Slice 2: Period {3}

      iii. Slice 3: Period {4}

   (e) And further.

Schedule	Binary	Period 1 Period 2 Period 3 Period 4
1	1000	[Upgrade - >]
2	1001	[Upgrade ->][Upgrade - ->]
3	1010	[Upgrade - - - - - - - - - - - >][Upgrade - - - - - - - - - - - >]
4	1011	[Upgrade - - - - - - - - - - - >][Upgrade - ->][Upgrade - ->]
5	1100	[Upgrade - ->][Upgrade ->]
6	1101	[Upgrade - ->][Upgrade - - - - - - - - - - - >][Upgrade - ->]
7	1110	[Upgrade - ->][Upgrade - ->][Upgrade - - - - - - - - - - - >]
8	1111	[Upgrade - ->][Upgrade - ->][Upgrade - ->][Upgrade - ->]

Table 4:
This table demonstrates the slice representation of all possible schedules with four upgrade periods.

8. In Table 4:

   (a) Schedule slice descriptions:

      i. Schedule 1 requires that the 1st upgrade must cover the entire upgrade window.

      ii. Schedule 2 requires that the 1st upgrade must cover periods {1, 2, 3}, and the 2nd upgrade cover period {4}.

      iii. Schedule 3 requires that the 1st upgrade must cover periods {1, 2}, and the 2nd upgrade cover periods {3, 4}.

      iv. Schedule 4 requires that the 1st upgrade must cover periods {1, 2}, the 2nd upgrade cover period {3}, and the 3rd upgrade cover period {4}.

      v. Schedule 5 requires that the 1st upgrade must cover period {1}, and the 2nd upgrade cover periods {2, 3, 4}.

      vi. Schedule 6 requires that the 1st upgrade must cover period {1}, the 2nd upgrade cover periods {2, 3}, and the 3rd upgrade cover period {4}.

      vii. Schedule 7 requires that the 1st upgrade must cover period {1}, the 2nd upgrade cover period {2}, and the 3rd upgrade cover periods {3, 4}.

      viii. Schedule 8 requires that the 1st upgrade must cover period {1}, the 2nd upgrade cover period {2}, the 3rd upgrade cover period {3}, and the 4th upgrade cover period {4}.

   (b) The term "Upgrade" in "[Upgrade - >]" refers to the evaluation of whether or not an upgrade is necessary for that slice. If an upgrade is necessary, the first period of the slice is when the upgrade occurs. Each schedule starts with a slice that includes the first period of the upgrade window.

(c) If the first period is not included, the following schedule binary would be possible: **0000**. What this schedule binary means is that none of the periods are evaluated for upgrades. At the surface level, it does not appear that this upgrade schedule is flawed; however, it does not guarantee sufficient capacity.

   i. To explain this further, the only way that schedule binary **0000** can be a valid schedule is if no upgrades are necessary. If no upgrades are necessary, the result of all the schedule evaluations will return the exact same result. The upgrade schedules are on-demand upgrades. Assuming any of these schedules allowed on-demand would be a potential source of confusion. The schedule binaries have a property, which dictates **if and when** upgrades can occur. If there is not a period to evaluate an upgrade according to a schedule, then no upgrades can occur. This property creates the necessity for the first slice always contains the first period, and the first period always contain an upgrade evaluation for the first slice.

   ii. In the context of the EV road trip example (from 4c), if the EV battery is fully charged and can last the entire trip, the evaluation of whether charging is necessary and less expensive than not charging is trivial. Conversely, if the beginning of the trip does not always check the battery charge, then it allows for the battery to not have enough charge to reach the next charging station.

3.1.2.2 Optimize Node Upgrade Schedules

1. With the upgrade schedules created, the optimization of the schedules can commence. For each schedule slice, every piece of equipment in the catalogue is evaluated sequentially to determine the equipment that has the lowest total cost across its slice.

2. For example: For {schedule 1 from Table 3}, the current implementation of the model evaluates every piece of equipment in the catalogue and compares the overall cost over the upgrade slice. The same process is applied to the rest of the schedule slices.

3. There are other ways to compute the optimal equipment other than evaluating all the equipment; however, the number of equipment is negligible with modern computing. This brute force method also allows for parallel processing, while other methods may require sequential processing.

(a) The reason the number of equipment is a negligible factor with modern computing power is that the number of equipment items from any distributor catalogue is anywhere from hundreds to hundreds of thousands. For instance, the Grainger (civil engineering equipment distributor and manufacturer) online catalogue has roughly "500,000" ((Grainger Canada, 2022)) items; however, the number of electrical transformers that are suitable for residential electricity distribution is a subset of 500,000 items. The number of suitable equipment from a catalogue is most likely on the scale of hundreds or fewer.

(b) In terms of computational complexity, the number of pices of equipment contributes to a constant multiplier, which does not change with a change in problem size. In some instances, a constant multiplier can require significant computational resources, but not in the case of the model presented in this thesis since the other terms increase exponentially with an increase in problem size.

3.1.3 Model: Descriptive Layer: Stage One: Output

1. The first output object is the costs associated with the optimized upgrade schedules for each node. The second output object is the equipment associated with the optimized upgrade schedules for each node.
2. For example, the schedule costs:

Schedule	Period 1	Period 2	Period 3	Period 4
1	$9K	$1K	$2K	$7K
2	$10K	$5K	$1K	$4K
3	$1K	$8K	$7K	$5K
4	$4K	$1K	$2K	$2K
5	$3K	$8K	$3K	$6K
6	$8K	$1K	$2K	$4K
7	$9K	$9K	$1K	$4K
8	$6K	$2K	$3K	$7K

Table 5:
This table demonstrates the schedule costs for each period of Node 1 (the specifics are arbitrary and serve the purpose of explanation only).

1. From the same example in Table 5, the associated schedule equipment:

Schedule	Period 1	Period 2	Period 3	Period 4
1	95	95	95	102
2	100	100	100	103
3	27	87	99	99
4	35	35	35	35
5	30	81	81	85
6	80	80	80	81
7	105	112	112	115
8	67	67	67	75

Table 6:
This table demonstrates the schedule equipment for each period of Node 1 (the specifics are arbitrary and serve the purpose of explanation only).

1. In Table 5, the 'Period' column values represent the cost associated with each 'Schedule' row.

2. In Table 6, the 'Period' column values represent the equipment index associated with each 'Schedule' row. The equipment indexes refer to elements in the equipment catalogue. When the same equipment index occurs across multiple rows, there is no scheduled upgrade for those periods. When the equipment index changes from one row to the next, there is a scheduled upgrade where the new equipment index occurs.

3.2 Model: Descriptive Layer: Stage Two

1. The purpose of stage two is to convert the stage one output into an approximate form that makes the task of optimization tractable.

3.2.1 Input

3.2.1.1 Optimized Upgrade Schedules

1. See Stage One: Output.

3.2.1.2 Number of Node Groups

1. A node group is a cluster of nodes that have similar schedule costs. 'Similar' means that the nodes in a cluster are closer in cost to their respective cluster centre than to the centres of all other clusters.

2. The number of node groups is simply the number of clusters that are chosen. The selection of the number of clusters is not necessarily arbitrary. Similar to how the number of periods and schedules are chosen using the model time complexity, the number of node groups is also a significant factor that can be calculated in this way.

   (a) As an analogy for calculating model input parameters using the amount of computational resources available, consider the following problem:

      i. The goal is to construct a rectangular prism (edges only) with the largest volume using 100 meters worth of wood planks.

      ii. Let the x-axis (width) represent the number of periods. Let the y-axis (length) represent the number of schedules. Let the z-axis (height) represent the number of node groups.

   (b) The point of the analogy above is to demonstrate the computational resources determining the model input parameters as opposed to the model input parameters determining the computational resources.

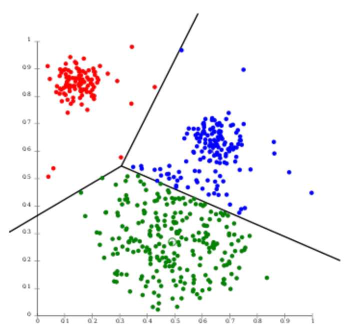

Figure 10: k-means clustering
This plot shows an arbitrary example of two-dimensional k-means clustering ("Cluster analysis," 2004)[12] (the specifics are arbitrary and serve the purpose of explanation only).

1. In the current iteration of the model, the number of nodes in each cluster (cluster size) is approximately equal. Equal cluster sizes are used for the sake of simplicity; however, it is possible to use clusters of unequal sizes by weighting the distribution of budget constraints.
   (a) The distribution of budget constraints is discussed in more detail in the structure portion of stage one.

3.2.1.3 Budget Constraints

1. The number of budget constraints is equal to the number of periods. What this means for the model is that for every period, the total cost of all upgrades must be equal to or less than the budget constraint for that period.
2. The budget constraints depend on factors that are external to the model. These factors are not explored in this thesis. Instead, various ranges of budget constraints are created in proportion to the schedule costs. The selection of budget constraint ranges is rather arbitrary, other than the fact that the ranges must be lenient enough so that some approximate schedule combinations can satisfy some of the budget constraints.

3.2.2 Structure

3.2.2.1 Create Node Group Approximations

1. The method used to represent each node group with an approximation is an algorithm that is sometimes referred to as max-pooling.
2. In essence, the schedules of all nodes in a group are aligned, and for each period, the maximum costs are found.
3. For example: In a group with three nodes, approximating all the first schedules for each group node is done by taking the maximum value for each period. The maximum values for each period represent the first schedule of the node group approximation. This is demonstrated in the following table:

Period	Node 1	Node 2	Node 3	Maximum
1	**$9K**	$8K	$7K	**$9K**
2	$1K	$2K	**$3K**	**$3K**
3	$2K	$3K	**$3K**	**$3K**
4	**$7K**	$5K	$4K	**$7K**

Table 7:
This table demonstrates the data structure used to approximate a group of nodes (the specifics are arbitrary and serve the purpose of explanation only).

1. In Table 7:

   (a) The label 'Period' represents the period of the upgrade schedule.

   (b) The labels with 'Node' represent the individual nodes within the group. The dollar values beneath these labels represent the costs associated with each period of the first schedule for each node.

   (c) The label 'Maximum' represents the group approximation of all the nodes in the first schedule. The dollar values beneath this label represent the maximum value along that row. This is the method used to approximate each schedule for each group.

   (d) The bold font values associate the maximum node cost with the 'Maximum' column result.

2. With the node group approximations, there is a need for an adjustment to budget constraints. The way this is done in the current implementation is to multiply each budget constraint by the ceiling of {the number of nodes divided by the number of node groups}. This makes the constraints compatible with the brute force optimization algorithm.

3.2.2.2 Brute Force Optimization

1. The stage two brute force optimizer evaluates every possible combination of node group schedules compared to the adjusted budget constraints.

   (a) Technically, it is more accurate to say 'permutation with repetitions' instead of 'combination.' Still, 'combination' is used as it sufficiently communicates the concept in this context – just as the word 'or' in common usage implies exclusivity.

2. This is similar to the way that the schedule binaries are determined; however, the base is no longer two, but rather uses the number of schedules per node group as the base.

$$N := \text{Number of nodes}$$

$$n_{\text{s}} := \text{Number of schedules}$$

$$n_{\text{c}} := \text{Number of combinations}$$

$$n_{\text{c}} = n_s^N \tag{4}$$

3. For example: If there are three nodes and three periods, each node would have four possible schedules and 64 possible combinations of schedules. The following table demonstrates the combination of all approximate schedules. Not to be confused with the schedules from the nodes within each group. Each group represents their respective members. In other words, each group is a virtual node constructed using its respective members. The term **virtual node** means that the node does not exist outside this model stage.

Combo	Group 1	Group 2	Group 3	Group 4
1	1	1	1	1
2	1	1	1	2
3	1	1	1	3
4	1	1	1	4
5	1	1	2	1
6	1	1	2	2
7	1	1	2	3
⋮	⋮	⋮	⋮	⋮
64	4	4	4	4

Table 8:
This table demonstrates the data structure containing all possible schedule combinations (the specifics are arbitrary and serve the purpose of explanation only).

4. In Table 8:

   (a) The combination column label is 'Combo.' The column of numbers underneath represents the combination indexes.

   (b) The node group column labels use 'Group ' and their group node index. The column of numbers underneath represents the group schedule indexes.

3.2.3 Output

3.2.3.1 Minimum Cost Schedule Combination

1. The results of the brute force optimizer are an optimal schedule combination; however, this is for the node group approximations as opposed to the exact grid of nodes. Stage three is where the optimal schedule combination is applied to the individual nodes within the node groups.

3.3 Model: Descriptive Layer: Stage Three

1. The purpose of stage three is to (i) unpack the results from stage two and distribute them to each individual node in each node group and (ii) search for the schedule with the minimum total cost per individual node.

3.3.1 Input

1. See Stage One: Output.

3.3.1.2 Minimum Group Cost Schedule Combination

1. See Stage Two: Output.

3.3.2 Structure

3.3.2.1 Unpacking Node Groups

1. The optimal combination of node group schedules is not necessarily useful on its own. For each node group, there exists a schedule with costs per period. These period costs are the new bounds (or constraints) to the nodes within their respective group. These bounds are applied to each node and do not depend on other nodes, which differs from the budget constraints that apply to the sum of all node costs.

2. In essence, the algorithm to create the node group approximations is run in reverse once the optimal group schedule is determined. There is a key difference: in the forward direction, the nodes in the group map to one group schedule; however, in the backward direction, the group schedule maps to any node schedules that are equal to or less than the group schedule.

3. For example:

   (a) In Table 7, the first schedules of every node in their group create the first schedule of the group approximation.

   (b) Focusing on Node 1, the node group approximation schedule costs per period are the new bounds, or constraints, for the schedules in Node 1.

   (c) The first schedule for the node group is [$9K, $3K, $3K, $7K]. Let the following table represent the schedule costs for Node 1:

Schedule	Period 1	Period 2	Period 3	Period 4
1	**$9K**	**$1K**	**$2K**	**$7K**
2	$10K	$5K	$1K	$4K
3	$1K	$8K	$7K	$5K
4	**$4K**	**$1K**	**$2K**	**$2K**
5	$3K	$8K	$3K	$6K
6	**$8K**	**$1K**	**$2K**	**$4K**
7	$9K	$9K	$1K	$4K
8	**$6K**	**$2K**	**$3K**	**$7K**

Table 9:
This table demonstrates the filtering of schedules based on the optimal schedule costs from the optimal group schedule costs (the specifics are arbitrary and serve the purpose of explanation only).

1. In Table 9, the schedules that are in bold font are all the schedules that are less than the bound (first group schedule); specifically, schedules {1, 4, 6, 8}.

3.3.2.2 Brute Force Optimization

1. For each individual node, a search is performed on the list of schedules (sorted from lowest to highest overall cost). Each of these schedules is compared to the node group bounds discussed above. If a schedule is found that satisfies each bound, it is selected as the optimum schedule for that node.

(a) The reason the first schedule that is found to satisfy the bounds is selected as optimal is that the list of schedules is sorted, so any other schedules that satisfy the bounds must have higher overall costs.

2. Once all the optimal schedules for each node are found, the costs and equipment are compiled and returned. In particular:

   (a) Optimized upgrade schedule combination costs.

   (b) Optimized upgrade schedule combination equipment.

Schedule	Period 1	Period 2	Period 3	Period 4	Total
1	\$9K	\$1K	\$2K	\$7K	\$19K
4	**\$4K**	**\$1K**	**\$2K**	**\$2K**	**\$9K**
6	\$8K	\$1K	\$2K	\$4K	\$15K
8	\$6K	\$2K	\$3K	\$7K	\$18K

Table 10:
This table demonstrates the brute force search for the lowest cost schedule (the specifics are arbitrary and serve the purpose of explanation only).

1. In Table 10, {schedule 4} is the lowest cost schedule. Once the lowest cost schedules are found for each node that satisfies the group bounds, these results are returned along with the associated equipment list for each schedule.

3.3.3 Output

1. The optimized upgrade schedule combination costs and equipment are the result of the model. It contains information regarding the equipment to have installed at each node for each period and the detailed costs associated.

2. For example, the schedule costs:

Node	Period 1	Period 2	Period 3	Period 4
1	$9K	$1K	$2K	$7K
2	$8K	$1K	$2K	$4K
3	$6K	$2K	$3K	$7K
4	$3K	$8K	$3K	$6K

Table 11:
This table demonstrates the data structure of the schedule combination costs (the specifics are arbitrary and serve the purpose of explanation only).

1. From the same example in Table 11, the associated schedule equipment:

Node	Period 1	Period 2	Period 3	Period 4
1	105	112	112	115
2	100	100	100	103
3	67	67	67	75
4	30	81	81	85

Table 12:
This table demonstrates the data structure of the schedule combination costs (the specifics are arbitrary and serve the purpose of explanation only).

1. Tables 11 and 12, show the general structure of the model output.

4 Model: Technical Layer

1. The technical layer of the model (this section) is not necessary for understanding the model as a concept; however, it may be useful if the model is transformed into real-world implementation. This section is roughly equivalent to thorough coding documentation that includes a brief algorithmic analysis to demonstrate the scalability of the model. Since this section pertains to application development, detailed contents can be found in **Appendix: Model: Technical Layer**.

5 Model: Rigorous Layer

5.1 Model: Rigorous Layer: Convergence

1. Although the property of convergence is fundamental to the model, the proof of convergence is discussed but not shown in this section because the details involve tens of pages worth of mathematical logic. The tediousness of the proof could otherwise be a source of confusion. For any interested reader: The step-by-step proof can be found in **Appendix: Model: Rigorous Layer**.

2. The proof itself is rather trivial. In plain language, the results of the model approach the exact solution as the resolution of the approximation increases.

3. At the core of the proof is the following:

$$
\begin{aligned}
m, n &\in \mathbb{Z} \\
\text{where,} &\quad m < n \\
\\
A &:= \text{an array of size } n \\
\text{where,} &\quad A \in \mathbb{R} \\
\\
A_0 &= A\,[0 : m] \\
A_1 &= A\,[m : n-1] \\
\\
\text{mean}\,(\max(A)) &\geq \text{mean}\,(\max(A_0) + \max(A_1))
\end{aligned} \tag{5}
$$

4. The array object, A, {contents and where it is split} are both arbitrary. This means that splitting A_0 and A_1, in the same way will also satisify equation 5. As the splits arrays approach the size of one, the maximum resolution has been achieved. The proof is slightly more complex

with objects in three dimensions; however, the recursive logic of nested approximations still holds.

5. For instance:

    The following is an example of one array and the various levels of nesting (various approximation resolutions). The approximation starts at the lowest resolution (Nest Level 4: One group to approximate all 16 elements) and approaches the highest resolution (Nest Level 0: 16 groups to approximate each individual element). Each increase in resolution results in an average that is closer to the true average of the array than the previous resolution. When the average reaches the highest possible resolution, the average is true average of the array. It is at this point that the approximation achieves convergence.

    The following demonstrates the resulting average from having one group represent 16 elements (Nest Level 4) and two groups represent 16 elements (Nest Level 3):

$$
\begin{aligned}
A &:= [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \\
\text{mean}(A) &= 8.5
\end{aligned}
$$

Nest Level 4:

$$
A_{4,0} = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
$$

$$
\text{mean}\left(\max(A_{4,0})\right) = 16
$$

Nest Level 3:

$$
\begin{aligned}
A_{3,0} &= [1,2,3,4,5,6,7,8] \\
A_{3,1} &= [9,10,11,12,13,14,15,16]
\end{aligned}
$$

$$
\begin{aligned}
\text{mean}\left(\max(A)\right) &\geq \text{mean}(\max(A_{3,0})+\max(A_{3,1})) \quad (6) \\
16 &\geq \text{mean}(8+16) \\
16 &\geq 12
\end{aligned}
$$

From above, the mean of maximum from Nest Level 4 is 16 and the mean of maximums from Nest Level 3 is 12. The inequality from equation 6 satisfies equation 5.

The following demonstrates the resulting average from having four groups represent 16 elements (Nest Level 2):

Nest Level 2:

$$\begin{aligned} A_{2,0} &= [1,2,3,4] \\ A_{2,1} &= [5,6,7,8] \\ A_{2,2} &= [9,10,11,12] \\ A_{2,3} &= [12,14,15,16] \end{aligned}$$

$$\begin{aligned} \text{mean}(\max(A_{3,0}){+}\max(A_{3,1})) &\geq \text{mean}\left(\sum_{i=0}^{3} \max\left(A_{2,i}\right)\right) \\ 12 &\geq \text{mean}(4+8+12+16) \\ 12 &\geq 10 \end{aligned} \tag{7}$$

From above, the mean of maximums from Nest Level 3 is 12 and the mean of maximums from Nest Level 2 is 10. The inequality from equation 7 satisfies equation 5.

The following demonstrates the resulting average from having eight groups represent 16 elements (Nest Level 1):

Nest Level 1:

$$\begin{aligned}
A_{1,0} &= [1, 2] \\
A_{1,1} &= [3, 4] \\
A_{1,2} &= [5, 6] \\
A_{1,3} &= [7, 8] \\
A_{1,4} &= [9, 10] \\
A_{1,5} &= [11, 12] \\
A_{1,6} &= [13, 14] \\
A_{1,7} &= [15, 16]
\end{aligned}$$

$$\begin{aligned}
\text{mean}\left(\sum_{i=0}^{3} \max\left(A_{2,i}\right)\right) &\geq \text{mean}\left(\sum_{i=0}^{7} \max\left(A_{1,i}\right)\right) \qquad (8) \\
10 &\geq 9
\end{aligned}$$

From above, the mean of maximums from Nest Level 2 is 10 and the mean of maximums from Nest Level 1 is 9. The inequality from equation 8 satisfies equation 5.

Finally, the following demonstrates the resulting average from having 16 groups represent 16 elements (Nest Level 0):

Nest Level 0:

$$\begin{aligned}
A_{0,0} &= [1] \\
A_{0,1} &= [2] \\
A_{0,2} &= [3] \\
A_{0,3} &= [4] \\
A_{0,4} &= [5] \\
A_{0,5} &= [6] \\
A_{0,6} &= [7] \\
A_{0,7} &= [8] \\
A_{0,0} &= [9] \\
A_{0,1} &= [10] \\
A_{0,2} &= [11] \\
A_{0,3} &= [12] \\
A_{0,4} &= [13] \\
A_{0,5} &= [14] \\
A_{0,6} &= [15] \\
A_{0,15} &= [16]
\end{aligned}$$

$$\begin{aligned}
\operatorname{mean}\left(\sum_{i=0}^{7} \max\left(A_{1,i}\right)\right) &\geq \operatorname{mean}\left(\sum_{i=0}^{15} \max\left(A_{0,i}\right)\right) \\
9 &\geq 8.5
\end{aligned}$$

From above, the mean of maximums from Nest Level 1 is 9 and the mean of maximums from Nest Level 0 is 8.5. The inequality from equation 8 satisfies equation 5.

From all the inequalities of averages above:

$$16 \geq 12 \geq 10 \geq 9 \geq 8.5$$

All the inequality hold for each increase in resolution, demonstrating the property of convergence through approximation. As mentioned at the beginning of this section, the model presented in this thesis uses a three-dimensional array (as opposed to the one-dimensional array in the example above) so the proof convergence requires more steps; however, the basic concept of convergences through approximation is similar.

6 Model: Summary

1. The key takeaways from the model presented in this thesis:
   (a) There are three-stages to the Breen Optimization Model (BOM):
      i. The first stage of the optimizer serves the purpose of processing the initial parameters and the grid demand data. The output is a set of individual grid elements (nodes) that contain schedules for upgrades. These schedules of upgrades have been optimized in the sense that for each possible schedule of upgrades, the equipment that results in the lowest cost is selected. At this stage of optimization, the budget constraints are not applied, so the optimization of each node is independent of all other nodes.
      ii. The second stage of the optimizer serves the purpose of processing the output from the first stage. The first step of processing the set of all the grid node schedules is to approximate the entire dataset by grouping similar nodes together. Once the nodes are represented in groups, every possible combination of schedules is evaluated against the budget constraints. The combination of schedules with the lowest overall cost that also satisfies the budget constraints is the most optimal combination of schedules. This combination of group schedules is the output of this stage.

iii. The third stage of the optimizer serves the purpose of processing the output from the second stage. The first step of processing the results from the previous stage is to unpack the group schedules and use each schedule cost as a filter for each node in each group. For each of the nodes within each group, the schedule with the lowest overall cost among the schedules that satisfy the group cost filter is the optimal schedule. The set of the optimal schedules of all the nodes is the output of the third stage.

(b) The implementation of each algorithm within the model is optimized for scalability to the greatest extent possible within the time constraints of this thesis. The overall time complexity of the model is in polynomial time (polynomial-time approximation scheme).

(c) The model is based on a rigorous foundation. The model is proven to converge to an exact solution as the resolution of the approximation approaches the number of nodes. In other words, any result that the model produces is accompanied by information regarding how optimal the result is.

Part III

Method

Part Contents:

1. Overview

2. Random Data

3. Toy Grid

"The tao that can be told

is not the eternal Tao. "

– Lao Tzu (Stephen Mitchell), *Tao Te Ching* [50]

1 Method: Overview

1. The following sections contain the procedures for (i) evaluating the overall performance of the model under conditions with significant noise (random data) and (ii) demonstrating the model performance in a synthetic use-case (toy grid). Many of the submethods include detailed algorithmic instructions.

2 Method: Random Data

1. This section aims to establish the procedures for determining if the null hypothesis is rejected. This is accomplished with sets of synthetic randomized data used as input to the second stage of the model. The randomized synthetic data are meant to emulate {schedule costs in each period for each node} with a variety of noise types.

2. The method skips to the second stage because the first stage requires real-world (or realistic) parameters. Without such specific parameters, it is impossible to construct results from the first stage. That being said, the first stage optimizer performs a thorough brute force search through all possible schedule upgrades. Additionally, the first stage is unconstrained and is not where convergence is observed. In contrast, the second and third stages are where constraints are applied, and convergence is observed.

3. The convergence of the approximately optimal results with increasing resolution under budget constraints is observable when the synthetic data are randomized to simulate chaotic systems. Although it is possible to have a well-structured set of schedules, it is not guaranteed. Without knowing specific instances of parameters, it is not possible to assess the performance of the model.

   (a) A "well-structured set of schedules" refers to identifiable patterns in the costs within each schedule. The construction of the first stage schedules involves evaluating periods in a computationally irreducible sequence – meaning that the n^{th} state cannot be known without calculating the $(n-1)^{\text{th}}$ state.

   (b) This discreteness of the first stage optimizer can compress the schedules into a branching tree of schedule costs.

4. The following two methods contain two submethods each: (i) unsorted data; and (ii) sorted data. The reason for including the evaluation of sorted random data at all is to check if monotonically increasing schedule costs return a significantly different result than unsorted random data.

2.1 Method: Random Data: Procedural Abstraction

1. With an understanding of the model, the numbered instructions might be sufficient for the understanding of the procedural steps. For more depth, numbered instructions details are also provided following the overview.

2.1.1 Numbered Instructions Overview

1. Set the synthetic grid data.

2. Create stage two approximations at a variety of node groups.

3. Create a range of budget constraints based on the grid approximations.

4. Optimize each synthetic dataset with every combination of period constraints.

5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.

2.1.2 Numbered Instructions Details

Notes:

1. The following are the in-depth procedures for evaluating the synthetic (schedule cost) datasets in each submethod. The pseudocode in the sub-points contains similar instructions in succession. This is meant to avoid the overuse of looping over small lists and to explicitly define distinctly independent objects.
2. Global parameter settings across all submethods:
   (a) These settings are constant in every instance of these procedures.
   (b) The number of nodes: num_nodes = 4
   (c) The number of periods: num_periods = 4
   (d) The number of schedules: num_schedules = 8
   (e) The shape of the synthetic grid:
      i. grid_shape = [num_nodes, num_periods, num_schedules]
   (f) The dimension dictionary:
      i. dim_dict = {"node": 0, "period": 1, "schedule": 2}

1. Set the synthetic grid data.
   (a) synthetic_grid:

i. This data object is constructed from the submethod instantiation of this procedural abstraction. See an instance of these procedures for specific details.

2. Create stage two approximations at a variety of node groups.

   (a) The numbers of approximation node groups: groups = [1, 2, 4]

   (b) synthetic_grid_1 = approximation(synthetic_grid, num_groups=groups[0])

   (c) synthetic_grid_2 = approximation(synthetic_grid, num_groups=groups[1])

   (d) synthetic_grid_4 = approximation(synthetic_grid, num_groups=groups[2])

3. Create a range of budget constraints based on the grid approximations.

   (a) The following process is performed for each synthetic_grid approximation:

      i. The lower bound of the budget constraints is constructed with the minimum period cost for each period.

         A. lower_bound_1 = minimum(synthetic_grid_1, axis=dim_dict[“period”])

         B. lower_bound_2 = minimum(synthetic_grid_2, axis=dim_dict[“period”])

         C. lower_bound_4 = minimum(synthetic_grid_4, axis=dim_dict[“period”])

      ii. The upper bound of the budget constraints is constructed with the minimum period cost for each period.

         A. upper_bound_1 = maximum(synthetic_grid_1, axis=dim_dict[“period”])

         B. upper_bound_2 = maximum(synthetic_grid_2, axis=dim_dict[“period”])

         C. upper_bound_4 = maximum(synthetic_grid_4, axis=dim_dict[“period”])

      iii. The period constraints for each period for each approximation is constructed with a four-point linear space between the upper and lower bounds.

         A. constraints_1_arguments:

            start = lower_bound_1

            stop = upper_bound_1

            number_of_points = 4

         B. constraints_2_arguments:

            start = lower_bound_2

stop = upper_bound_2

number_of_points = 4

C. constraints_4_arguments:

start = lower_bound_4

stop = upper_bound_4

number_of_points = 4

D. constraints_1 = linear_space(constraints_1_arguments)

constraints_1.shape = [4, 4]

E. constraints_2 = linear_space(constraints_2_arguments)

constraints_2.shape = [4, 4]

F. constraints_4 = linear_space(constraints_4_arguments)

constraints_4.shape = [4, 4]

iv. The mesh of constraints that each approximation is evaluated at is constructed by calculating every possible combination of period constraints.

A. mesh_arguments_1:

constraints_1[0]

constraints_1[1]

constraints_1[2]

constraints_1[3]

B. mesh_arguments_2:

constraints_2[0]

constraints_2[1]

constraints_2[2]

constraints_2[3]

C. mesh_arguments_4:

constraints_4[0]

constraints_4[1]

constraints_4[2]

constraints_4[3]

D. mesh_constraints_1 = mesh_grid(mesh_arguments_1)

mesh_constraints_1.shape = [4, 4, 4, 4]

E. mesh_constraints_2 = mesh_grid(mesh_arguments_2)

mesh_constraints_2.shape = [4, 4, 4, 4]

F. mesh_constraints_4 = mesh_grid(mesh_arguments_4)

mesh_constraints_4.shape = [4, 4, 4, 4]

v. The mesh of constraints is reshaped so that each row contains a unique combination of period constraints.

A. mesh_long_1 = reshape(mesh_constraints_1, shape=[64, 4])

B. mesh_long_2 = reshape(mesh_constraints_2, shape=[64, 4])

C. mesh_long_4 = reshape(mesh_constraints_4, shape=[64, 4])

4. Optimize each synthetic dataset with every combination of period constraints.

(a) stage2_results_1 := A list to contain the 64 different results from the application of the 64 different sets of constraints on the optimization of synthetic_grid_1.

(b) stage2_results_2 := A list to contain the 64 different results from the application of the 64 different sets of constraints on the optimization of synthetic_grid_2.

(c) stage2_results_4 := A list to contain the 64 different results from the application of the 64 different sets of constraints on the optimization of synthetic_grid_4.

(d) constraints_combo_1 := A row from mesh_long_1

i. constraints_combo_1.shape = [4]

(e) constraints_combo_2 := A row from mesh_long_2

i. constraints_combo_2.shape = [4]

(f) constraints_combo_4 := A row from mesh_long_4

i. constraints_combo_4.shape = [4]

(g) For each constraints_combo_1 in mesh_long_1:

i. optimization_arguments_1:

            A. dataset = synthetic_grid_1
            B. budget_constraints = constraints_combo_1
        ii. temp = stage2_optimization(optimization_arguments_1)
        iii. stage2_results_1.append(temp)

    (h) For each constraints_combo_2 in mesh_long_2:
        i. optimization_arguments_2:
            A. dataset = synthetic_grid_2
            B. budget_constraints = constraints_combo_2
        ii. temp = stage2_optimization(optimization_arguments_2)
        iii. stage2_results_2.append(temp)

    (i) For each constraints_combo_4 in mesh_long_4:
        i. optimization_arguments_4:
            A. dataset = synthetic_grid_4
            B. budget_constraints = constraints_combo_4
        ii. temp = stage2_optimization(optimization_arguments_4)
        iii. stage2_results_4.append(temp)

5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.
    (a) stage3_results_1 = stage3_optimization(stage2_results_1)
    (b) stage3_results_2 = stage3_optimization(stage2_results_2)
    (c) stage3_results_4 = stage3_optimization(stage2_results_4)

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.
    (a) The average percentage saved:
        i. weighted_mean_arguments:

A. mean_1 = (stage3_results_1 - constraints_1) / constraints_1

B. mean_2 = (stage3_results_2 - constraints_2) / constraints_2

C. mean_4 = (stage3_results_4 - constraints_4) / constraints_4

ii. mean_percent_saved = weighted_mean(weighted_mean_arguments)

A. The weighted mean is used because it is not a guarantee that if the four group approximation satisfies their constraints that the two group approximation will also satisfy the same constraints. Same from two groups to one group. The mean_percent_saved represents the savings for any iteration regardless of the number of groups.

(b) The average rate of convergence:

i. weighted_mean_arguments:

A. mean_4_2 = mean_4 - mean_2

B. mean_2_1 = mean_2 - mean_1

ii. mean_convergence = weighted_mean(weighted_mean_arguments)

2.2 Method: Random Data: Uniform Distribution

1. In this method, the schedule costs are sampled from a uniform distribution.

2. The uniform distribution bounds:

   (a) uniform_min = 0.1

   (b) uniform_max = 1.0

3. The synthetic uniform grid:

   (a) uniform_arguments:

      i. min = uni_min

      ii. max = uni_max

      iii. shape = grid_shape

   (b) grid_uniform = uniform_distribution(uniform_arguments)

2.2.1 Uniform Distribution: Unsorted

The following is an example of where the unsorted uniform distribution dataset is sampled from:

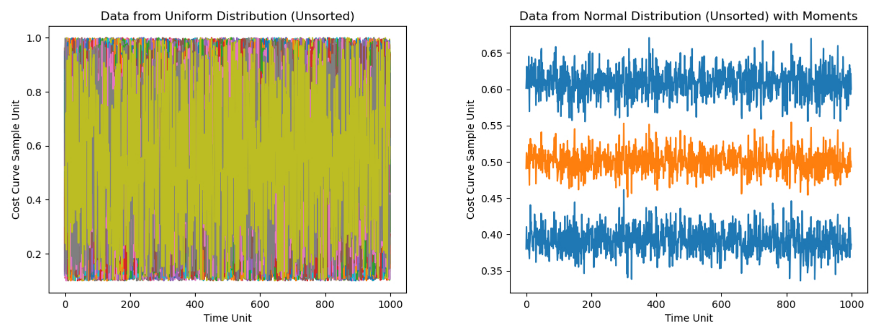

Figure 11:
This is a plot of an example of the random data from a uniform distribution that has not been sorted. On the left: The full view of the sampled data. On the right: The first and second moments of the sampled data (mean and standard deviation).

The plot in Figure 11 is not a plot of the dataset used in this method. The parameter is set in such a way as to demonstrate the structure that the dataset is sampled.

The following are the procedures for evaluating the synthetic data from a uniform distribution that is unsorted:

1. Set the synthetic data with a uniform distribution that is unsorted.

   The synthetic uniform grid: uniform_unsorted = grid_uniform

   - The creation of the grid_uniform is already unsorted.

2. Create stage two approximations at a variety of node groups.
3. Create a range of budget constraints based on the grid approximations.
4. Optimize each uniform unsorted dataset with every combination of period constraints.
5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.

2.2.2 Uniform Distribution: Sorted

The following is an example of where the sorted uniform distribution dataset is sampled from:

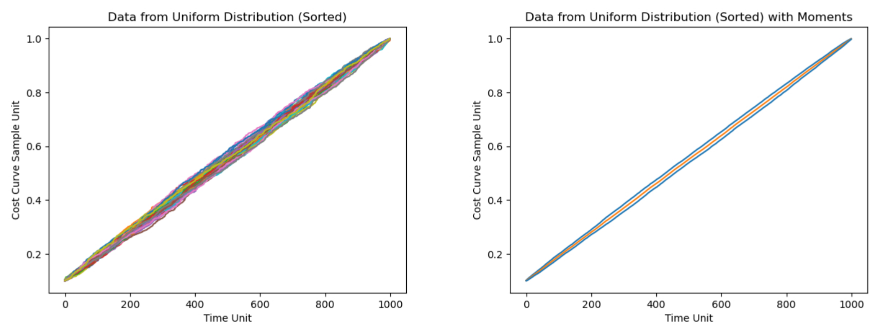

Figure 12:
This is a plot of an example of the random data from a uniform distribution that has been sorted. On the left: The full view of the sampled data. On the right: The first and second moments of the sampled data (mean and standard deviation).

The plot in Figure 12 is not a plot of the dataset used in this method. The parameter is set in such a way as to demonstrate the structure that the dataset is sampled.

The following are the procedures for evaluating the synthetic data from a uniform distribution that is unsorted:

1. Set the synthetic data with a uniform distribution that is sorted.

   The synthetic uniform grid: uniform_sorted = grid_uniform

   - uniform_sorted = sort(grid_uniform, axis=dim_dict["schedule"]

2. Create stage two approximations at a variety of node groups.

3. Create a range of budget constraints based on the grid approximations.

4. Optimize each synthetic dataset with every combination of period constraints.

5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.

2.3 Method: Random Data: Normal Distribution

In this method, the schedule costs are sampled from a normal distribution. The normal distribution is truncated so that all samples are within two standard deviations of the mean.

The synthetic normal grid:

- normal_arguments:
    - mean = 0.5
    - standard_deviation = 0.125
    - shape = grid_shape
- grid_normal = normal_distribution(normal_arguments)

2.3.1 Normal Distribution: Unsorted

The following is an example of where the unsorted normal distribution dataset is sampled from:

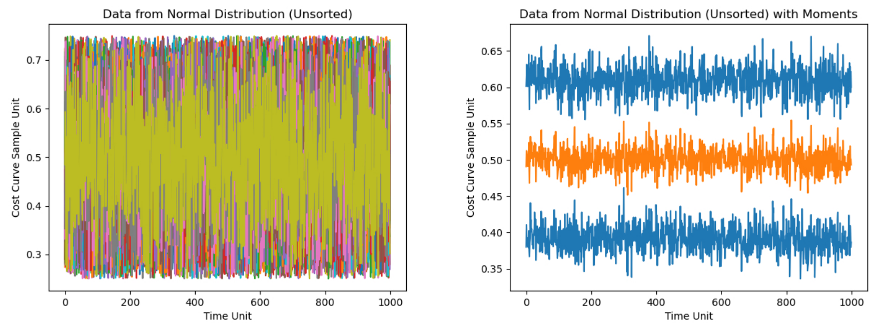

Figure 13:
This is a plot of an example of the random data from a normal distribution that has not been sorted. On the left: The full view of the sampled data. On the right: The first and second moments of the sampled data (mean and standard deviation).

The plot in Figure 13 is not a plot of the dataset used in this method. The parameter is set in such a way as to demonstrate the structure that the dataset is sampled.

The following are the procedures for evaluating the synthetic data from a normal distribution that is unsorted:

1. Set the synthetic data with a normal distribution that is unsorted.

   The synthetic normal grid:

   - normal_unsorted = grid_normal
   - The creation of the grid_normal is already unsorted.

2. Create stage two approximations at a variety of node groups.

3. Create a range of budget constraints based on the grid approximations.

4. Optimize each synthetic dataset with every combination of period constraints.

5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.

2.3.2 Normal Distribution: Sorted

The following is an example of where the sorted normal distribution dataset is sampled from:

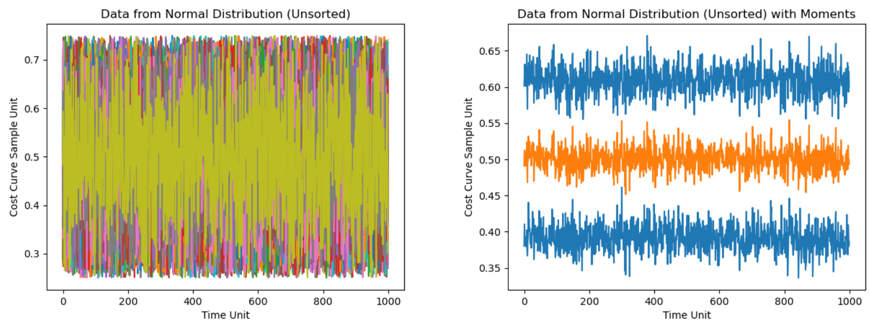

Figure 14:
This is a plot of an example of the random data from a normal distribution that has been sorted. On the left: The full view of the sampled data. On the right: The first and second moments of the sampled data (mean and standard deviation).

The plot in Figure 14 is not a plot of the dataset used in this method. The parameter is set in such a way as to demonstrate the structure that the dataset is sampled.

The following are the procedures for evaluating the synthetic data from a normal distribution that is unsorted:

1. Set the synthetic data with a normal distribution that is sorted.

   The synthetic sorted normal grid:

   - normal_sorted = sort(grid_normal, axis=dim_dict[“schedule”]

2. Create stage two approximations at a variety of node groups.

3. Create a range of budget constraints based on the grid approximations.

4. Optimize each synthetic dataset with every combination of period constraints.

5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.

2.4 Method: Random Data: Sigmoid with Noise

In this method, the schedule costs are sampled from a sigmoid with noise. It is meant to emulate a sharp increase in schedule costs due to increased demand for EV charging.

The way noise is introduced into the sigmoid is by sampling the sigmoid function coefficients from random distributions.

- The sigmoid with noise function is defined as follows:

$$
\begin{aligned}
f(x) &:= \text{Sigmoid function} \\
S(f(x)) &:= \text{Standardization of } f(x) \\
\beta_0 &:= \text{Squish range factor} \\
\beta_1 &:= \text{Offset domain factor} \\
\beta_2 &:= \text{Stretch domain factor} \\
\beta_3 &:= \text{Base range factor}
\end{aligned}
$$

where,

$$\beta_0, \beta_1, \beta_2, \beta_3 \in \mathbb{R}$$

$$f(x) = \beta_0 \left(\left(1 + e^{\dfrac{\beta_1 - x}{\beta_2}} \right)^{-1} + \beta_3 \right) \tag{9}$$

$$S(f(x)) = \frac{f(x)}{\max\left(f(x)\right)} \tag{10}$$

The specific β settings for the sigmoid are designed to introduce random noise, but also ensure that the change in the convexity of the sigmoid occurs within the upgrade window.

- This is to emulate a pressure such as transitioning to 100% EV.

The following are examples of the sigmoid β transformation factors:

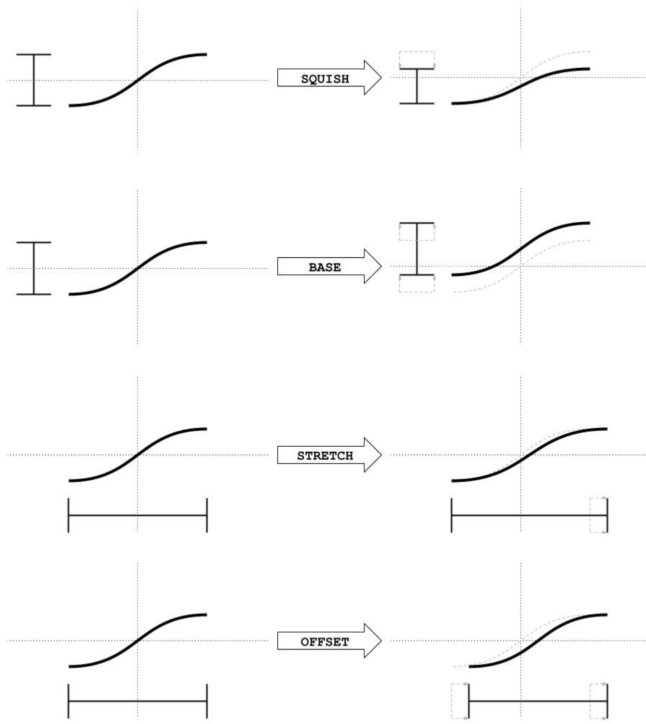

Figure 15:
This graphic demonstrates the transformations of the sigmoid curves. From top to bottom: “SQUISH” is the vertical scaling of the curve; “BASE” is the vertical translation of the curve; “STRETCH” is the horizontal scaling of the curve; “OFFSET” is the horizontal translation of the curve.

2.4.1 Sigmoid with Noise: Uniform

The following is an example of where the sigmoids with uniform noise dataset is sampled:

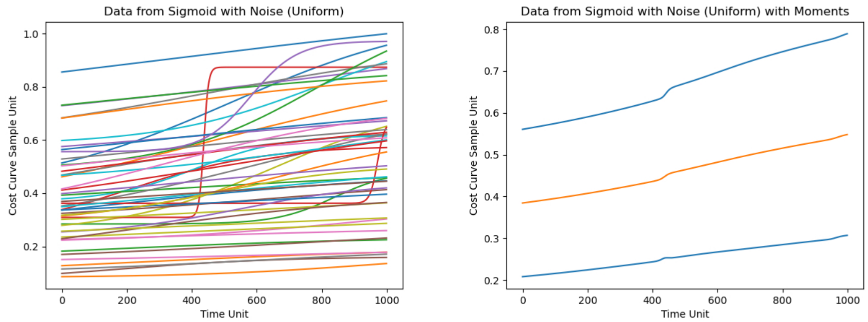

Figure 16:
This is a plot of an example of the sigmoid data with uniform noise. On the left: The full view of the sampled data. On the right: The first and second moments of the sampled data (mean and standard deviation).

The plot in Figure 16 is not a plot of the dataset used in this method. The parameter is set in such a way as to demonstrate the structure that the dataset is sampled.

The following are the procedures for evaluating the synthetic data from a sigmoid distribution that is normal:

1. Set the synthetic data of sigmoids with uniform noise.
   (a) The synthetic sigmoid grid:
      i. grid_length = num_nodes * num_schedules
      ii. offset_min = 0.1
      iii. offset_max = num_periods
      iv. offset_arguments:
         A. min = offset_min
         B. max = offset_max
         C. shape = grid_length
      v. offset_sample = uniform_distribution(offset_arguments)
      vi. stretch_min = 0.1
      vii. stretch_max = num_periods
      viii. stretch_arguments:

        A. min = stretch_min
        B. max = stretch_max
        C. shape = grid_length
    ix. stretch_sample = uniform_distribution(stretch_arguments)
    x. squish_min = 0.1
    xi. squish_max = 1
    xii. squish_arguments:
        A. min = squish_min
        B. max = squish_max
        C. shape = grid_length
    xiii. squish_sample = uniform_distribution(squish_arguments)
    xiv. base_min = 0.5
    xv. base_max = 1.5
    xvi. base_arguments:
        A. min = base_min
        B. max = base_max
        C. shape = grid_length
    xvii. base_sample = uniform_distribution(base_arguments)
    xviii. sigmoid_arguments:
        A. offset_sample
        B. stretch_sample
        C. squish_sample
        D. base_sample
    xix. sigmoid_uniform = sigmoid_function(sigmoid_arguments)

2. Create stage two approximations at a variety of node groups.

3. Create a range of budget constraints based on the grid approximations.

4. Optimize each synthetic dataset with every combination of period constraints.

5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.

2.4.2 Sigmoid with Noise: Normal

The following is an example of where the sigmoids with normal noise dataset is sampled:

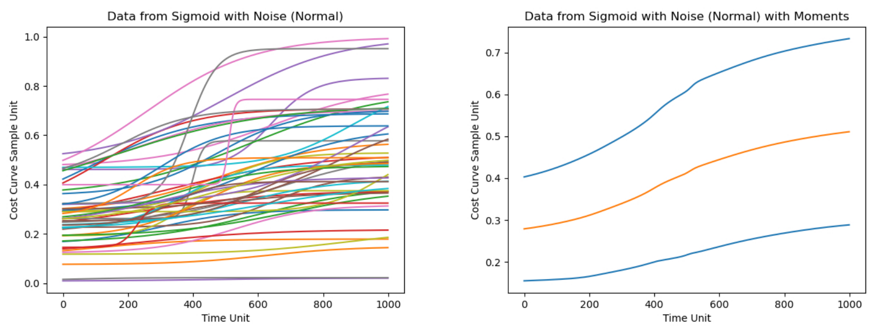

Figure 17:
This is a plot of an example of the sigmoid data with uniform noise. On the left: The full view of the sampled data. On the right: The first and second moments of the sampled data (mean and standard deviation).

The plot in Figure 17 is not a plot of the dataset used in this method. The parameter is set in such a way as to demonstrate the structure that the dataset is sampled.

The following are the procedures for evaluating the synthetic data from a sigmoid distribution that is normal (truncated within two standard deviations):

1. Set the synthetic data of sigmoids with normal noise.

   (a) The synthetic sigmoid grid:

      i. grid_length = num_nodes * num_schedules

ii. offset_mu = num_periods / 2

iii. offset_sigma = num_periods / 4

iv. offset_arguments:

   A. mean = offset_mu

   B. standard_deviation = offset_sigma

   C. shape = grid_length

v. offset_sample = normal_distribution(offset_arguments)

vi. stretch_mu = num_periods / 8

vii. stretch_sigma = num_periods / 16

viii. stretch_arguments:

   A. mean = stretch_mu

   B. standard_deviation = stretch_sigma

   C. shape = grid_length

ix. stretch_sample = normal_distribution(stretch_arguments)

x. squish_mu = 1 / 2

xi. squish_sigma = 1 / 4

xii. squish_arguments:

   A. mean = squish_mu

   B. standard_deviation = squish_sigma

   C. shape = grid_length

xiii. squish_sample = normal_distribution(squish_arguments)

xiv. base_mu = 1

xv. base_sigma = 1 / 4

xvi. base_arguments:

   A. mean = base_mu

   B. standard_deviation = base_sigma

   C. shape = grid_length

xvii. base_sample = normal_distribution(base_arguments)

xviii. sigmoid_arguments:

A. offset_sample

B. stretch_sample

C. squish_sample

D. base_sample

xix. sigmoid_uniform = sigmoid_function(sigmoid_arguments)

2. Create stage two approximations at a variety of node groups.

3. Create a range of budget constraints based on the grid approximations.

4. Optimize each synthetic dataset with every combination of period constraints.

5. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

6. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the average rate of convergence from {4 groups to 2 groups} and {2 groups to 1 group}.

3 Method: Toy Grid

1. The purpose of the toy grid method differs significantly from the random data method. It differs in terms of what aspect of performance is being measured, but also differs in the procedures.

2. More specifically, the purpose of the electrical toy grid use-case is to demonstrate: (i) the behaviour of the model with simulated electrical grid conditions that are not entirely unrealistic (more realistic than entirely random data); and (ii) the propagation of demand projection uncertainty through the model.

3. These specifics require a different set of procedures; in particular, the synthetic data is inserted into the first stage, as opposed to the second stage from the preceding methods. Also, the time series are no longer independent. The average time series projection is synthesized, and the

other two time series are derived from this average. The details of this process are discussed in the following material.

4. As mentioned in the **Model: Descriptive Layer** section, the electrical transformer equipment is modelled only after transformers with aluminum windings. Consequently, the realism of the synthetic data is relatively limited due to the equipment restrictions. In a real-world application, the equipment attributes (such as materials, efficiency, price, installation requirements, and durability) can range significantly.

3.1 Method: Toy Grid: Energy Demands

The toy grid demands are derived from the Independent Electricity System Operator (IESO) forecasted electrical energy demands that are mentioned in the descriptive layer of the model.

- This publicly available forcast from IESO are hourly energy demands in megawatts from 1 Januaray 2020 to 31 December 2040 for all of Ontario.

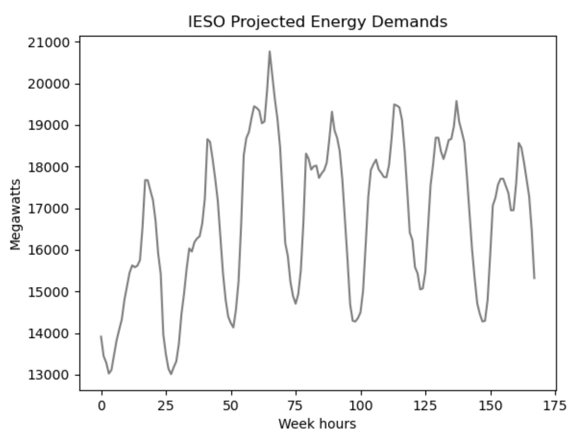

Figure 18:
This is a plot of an excerpt from the forecasted electrical energy demands for Ontario from 2020 to 2040 from the Independent Electricity System Operator (IESO) predictive model (IESO, 2021)[26]. The excerpt covers the first week of January 2020.

Although the forecast of demands brings an element of realism to the synthetic data, the realism due to the derivation of demands is not a significant consideration in the evaluation of the model in this method.

However, since the simulated demands are sampled with noise from the forecast, it is possible to reconstruct the forecast.

- This is to say that it is possible to construct a toy grid with the procedures used in this method that results in the same IESO forecasted electrical energy demands they are derived from – on average, with a normal deviation.

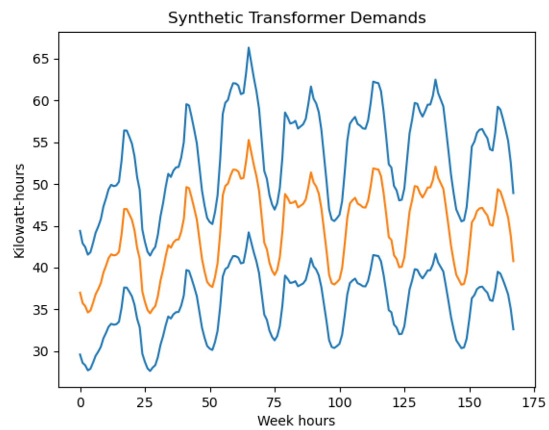

Figure 19:
This is a plot of an excerpt of a week of synthetic toy grid demands for one node derived from the IESO projections. The orange line represents the average of demands for a transformer. The blue line beneath the orange line represents the lower bound of demands, while the blue line above the orange line represents the upper bound of demands.

3.1.1 Numbered Instructions Details:

1. Create three-time series of grid demands that represent the mean of demands per time step and the deviation from the mean of demands per time step, as seen in Figure 19.

2. Optimize schedules for each node in stage one for each dataset and output the three synthetic schedule datasets.

3. Create stage two approximations for each synthetic grid using four-node groups with three nodes per group.

4. Create a range of budget constraints based on the grid approximations.

5. Optimize each synthetic dataset with every combination of period constraints.

6. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

7. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the number of budget constraints that are satisfiable.

3.1.2 Numbered Instructions Details:

The following are the in-depth procedures for evaluating the synthetic (schedule cost) datasets in this method. The pseudocode in the sub-points contains similar instructions in succession. This is meant to avoid the overuse of looping over small lists and to explicitly define distinctly independent objects.

The exact details of how every element is constructed are not included for many reasons. One reason is that this grid is not meant to be perfectly realistic, so the exact details do not develop the ideas in this thesis in any way. Further, although the process is not particularly complicated or advanced, the details are rather tedious – several hundred lines of logic.

Global parameter settings:

- The number of nodes: num_nodes = 12
  - This means that each node group contains three nodes.
- The number of periods: num_periods = 4
- The number of schedules: num_schedules = 8
- The shape of the synthetic grid:
  - grid_shape = [num_nodes, num_periods, num_schedules]

1. Create three-time series of grid demands that represent the mean of demands per time step and the deviation from the mean of demands per time step, as seen in Figure 19.

   (a) The three time series object definitions:

   i. synthetic_grid_mu := The average of demands.
   ii. synthetic_grid_lo := The lower bound of demands.
   iii. synthetic_grid_hi := The upper bound of demands.

(b) The first time series to be constructed is synthetic_grid_mu. This time series is derived from the IESO energy projects shown in Figure 18. This is calculated using the ratio between the overall energy demand of Ontario per year versus the average household energy demand of Ontario per year according to Statistics Canada ("Household energy consumption, Canada and provinces," 2017)[24] in 2015.
   i. household_Gigajoules = 101.0
   ii. Ontario_Gigajoules = 533097802.0
   iii. energy_ratio= household_Gigajoules / Ontario_Gigajoules
   iv. household_demands = energy_ratio * IESO_projections

(c) Once an estimation for household demands is created, the next step is to create the nodes of the grid with random samples of households.
   i. The number of households per node is determined by the following:
      A. normal_arguments:
         mean = 7,
         standard_deviation = 0.1
      B. num_households = floor(normal_distribution(normal_arguments))
         The random sample is truncated at greater than or equal to one.
   ii. The node demands created from this random number of household_demands are then scaled with noise. This process creates four distinct node groups where the nodes within each group are closer together than all other nodes.
   iii. The result of this step is the creation of the grid demands with node groups, but without the inclusion of EV charging.

(d) After the grid demands are constructed, the final step is to add the EV charging per household and conclude the construction of synthetic_grid_mu.
   i. The way that the EV charging energy demands are added is through the use of sampling from a sigmoid distribution. This step is similar to the sigmoids with noise in the preceding method of random data.

ii. When these simulated households transition to EVs, the charging kilowatt-hours are added at a random time in the evening every day from then on.

A. This is meant to emulate the routine charging behaviour of most households.

(e) The remaining time series are derived from the synthetic_grid_mu data.

i. The percentage of deviation here might represent the confidence interval for the energy demand projections, but realistically represent any variation in the deviation of the mean that might be of interest. The arbitrary percentage that is used is 20%:

A. synthetic_grid_hi = synthetic_grid_mu * 1.2

B. synthetic_grid_lo = synthetic_grid_mu * 0.8

2. Optimize schedules for each node in stage one for each dataset and output the three synthetic schedule datasets.

   (a) Each of the synthetic grids is used as input into the first stage of the model so that the optimized schedules can be constructed.

3. Create stage two approximations for each synthetic grid using four-node groups with three nodes per group.

   (a) approximation_hi = approximation(synthetic_grid_hi, num_groups=4)

   (b) approximation_mu = approximation(synthetic_grid_mu, num_groups=4)

   (c) approximation_lo = approximation(synthetic_grid_lo, num_groups=4)

4. Create a range of budget constraints based on the grid approximations.

   (a) This step is virtually identical to the creation of budget constraints in the preceding method for random data.

5. Optimize each synthetic dataset with every combination of period constraints.

   (a) This step is virtually identical to the creation of budget constraints in the preceding method for random data.

6. Unpack the results from stage two and apply the bounds to the individual nodes in each respective group in the stage three optimizer.

    (a) This step is virtually identical to the creation of budget constraints in the preceding method for random data.

7. Analyze the optimization results from each approximation. The first measure of performance is the overall average percentage saved with respect to every set of constraints. The next measure is the number of budget constraints that are satisfiable.

Part IV
Analysis

Part Contents:

1. Overview

2. Random Data

3. Toy Grid

"Give me a place to stand and with a lever I will move the whole world."

– Archimedes[29]

1 Analysis: Overview

1. The purpose of the analysis part of this thesis is to display the results of the method procedures used to analyze the model. This part is relatively limited as compared to the preceding parts since each method returns at most three numerical objects with minimal commentary for context. A more thorough interpretation of the results occurs in the discussion part of the thesis.

2. Each result has a corresponding histogram to display the approximate distribution of data points. This is meant to complement the calculated means and standard deviations since the distributions are not necessarily from normal distributions.

2 Analysis: Random Data

1. The following results are from an accumulation of ten **realizations** of the method procedures for random data.
   (a) The term **realization** refers to an iteration of synthetic data. Each realization contains a unique set of random data (each iteration samples a new realization of synthetic data).

2.1 Analysis: Random Data: Uniform Distribution

2.1.1 Uniform Distribution: Unsorted

2.1.1.1 Uniform Distribution: Unsorted: Percent Savings

1. Average Percent Savings: 29.4% (μ) $\pm$ 7.2% (σ)
2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

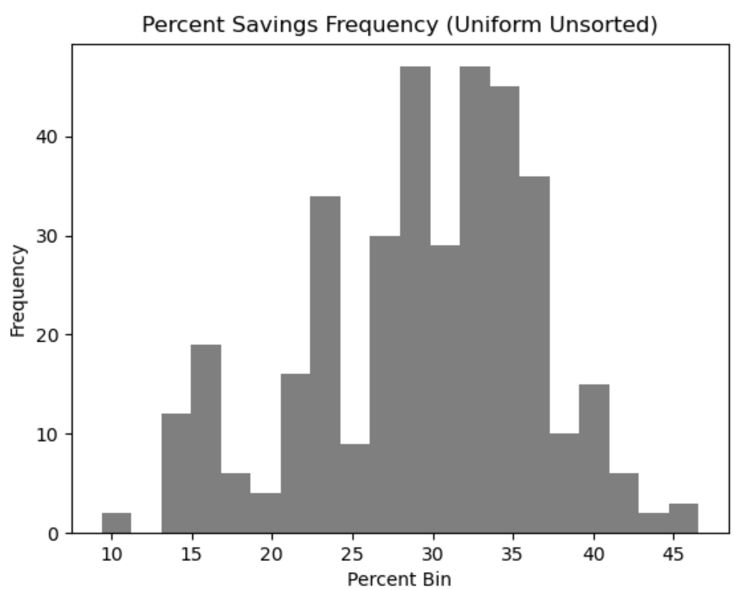

Figure 20:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

2.1.1.2 Uniform Distribution: Unsorted: Percent Convergence

1. Average Percent of Convergence: 3.9% (μ) $\pm$ 3.5% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results:

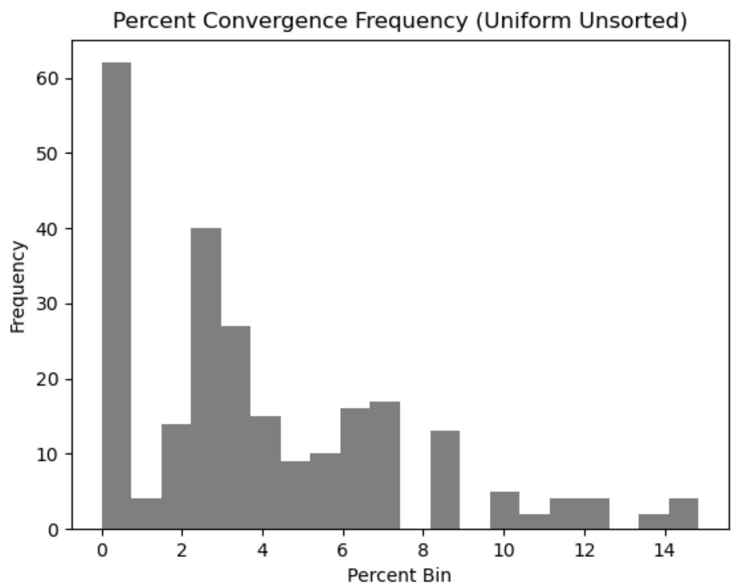

Figure 21:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

2.1.2 Uniform Distribution: Sorted

2.1.2.1 Uniform Distribution: Sorted: Percent Savings

1. Average Percent Savings: 26.1% (μ) $\pm$ 6.5% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

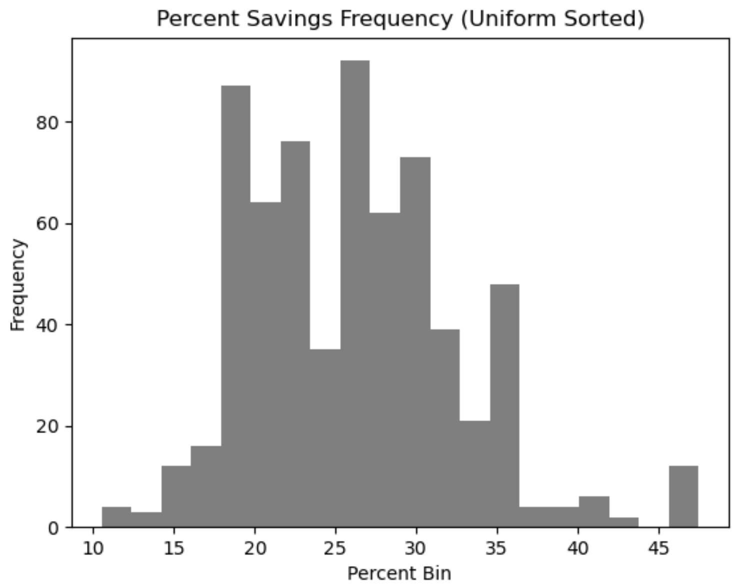

Figure 22:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

2.1.2.2 Uniform Distribution: Sorted: Percent Convergence

1. Average Percent of Convergence: 2.1% (μ) $\pm$ 2.3% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

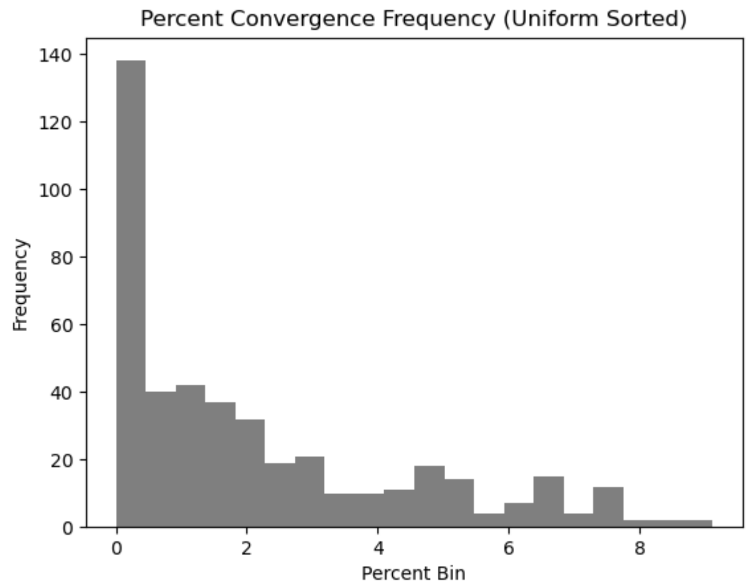

Figure 23:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

2.2 Analysis: Random Data: Normal Distribution

2.2.1 Normal Distribution: Unsorted

2.2.1.1 Normal Distribution: Unsorted: Percent Savings

1. Average Percent Savings: 74.8% (μ) ± 6.6% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

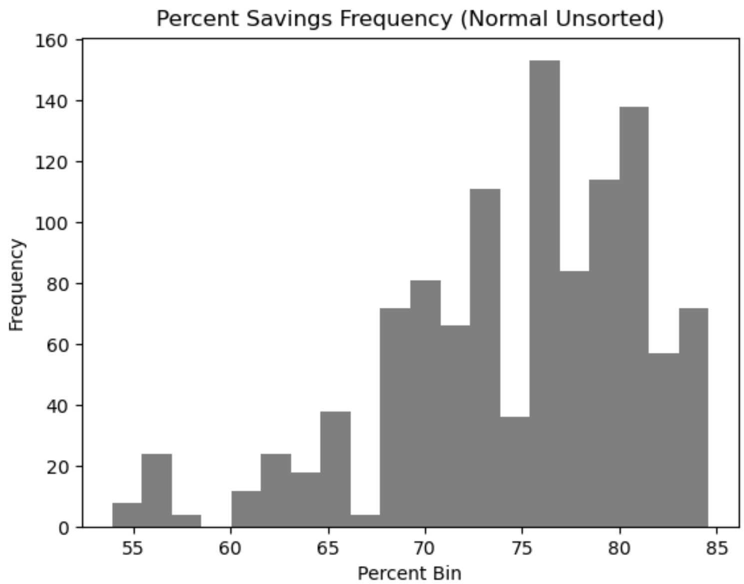

Figure 24:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

2.2.1.2 Normal Distribution: Unsorted: Percent Convergence

1. Average Percent of Convergence: 0.1% (μ) ± 0.4% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

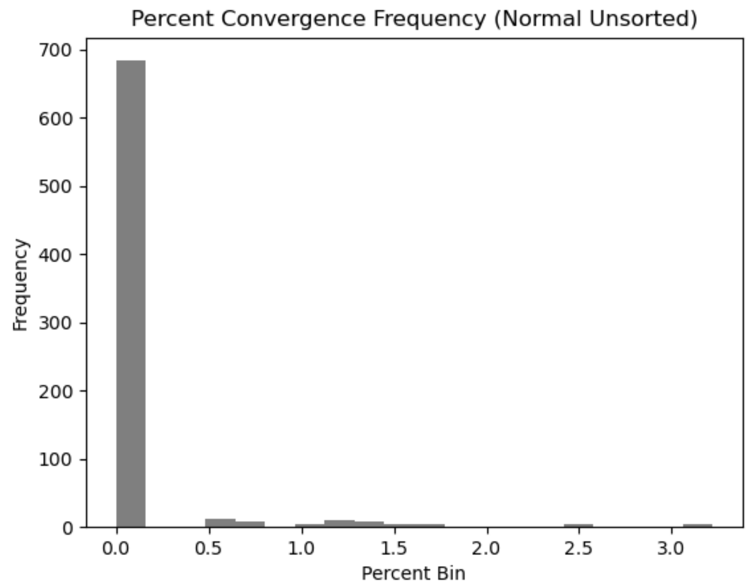

Figure 25:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

2.2.2 Normal Distribution: Sorted

2.2.2.1 Normal Distribution: Sorted: Percent Savings

1. Average Percent Savings: 74.8% (μ) ± 7.2% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

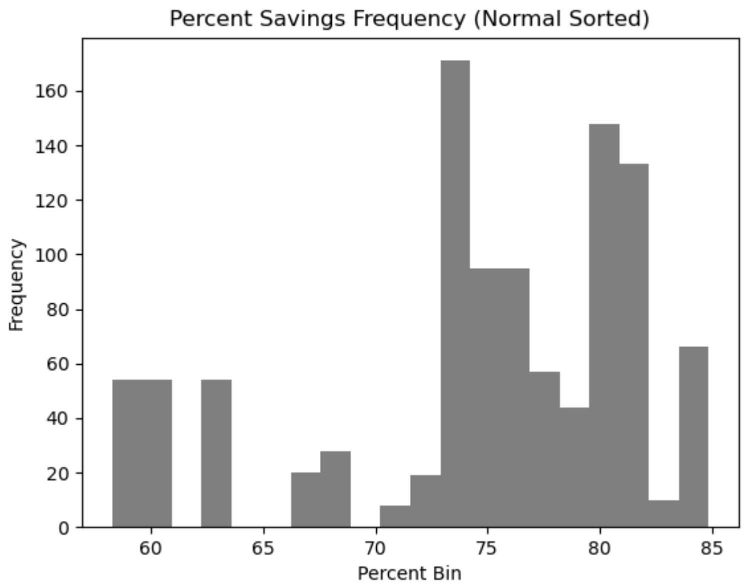

Figure 26:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

2.2.2.2 Normal Distribution: Sorted: Percent Convergence

1. Average Percent of Convergence: 0.1% (μ) ± 0.3% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

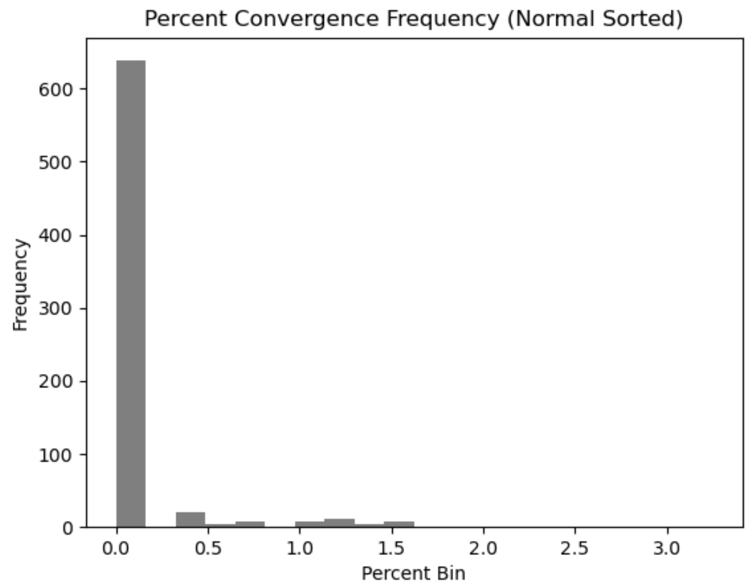

Figure 27:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

2.3 Analysis: Random Data: Sigmoid with Noise

2.3.1 Sigmoid with Noise: Uniform

2.3.1.1 Sigmoid with Noise: Uniform: Percent Savings

1. Average Percent Savings: 45.9% (μ) $\pm$ 14.6% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

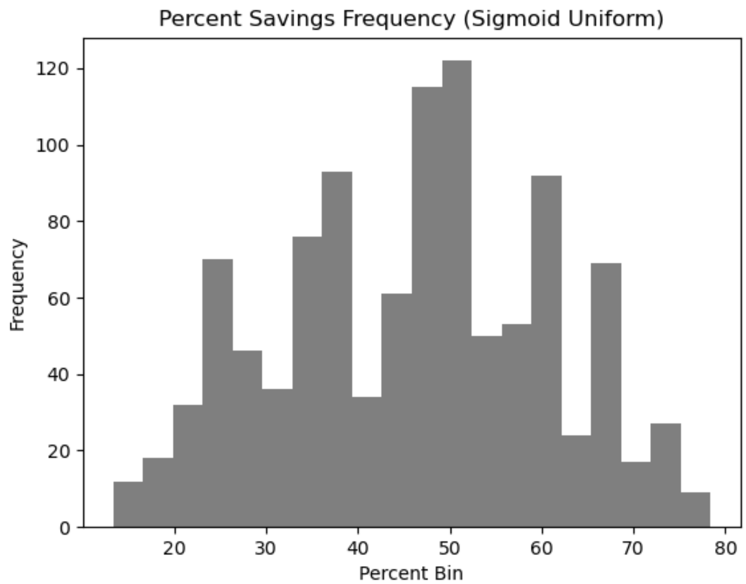

Figure 28:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

2.3.1.2 Sigmoid with Noise: Uniform: Percent Convergence

1. Average Percent of Convergence: 4.2% (μ) ± 5.1% (σ)
2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

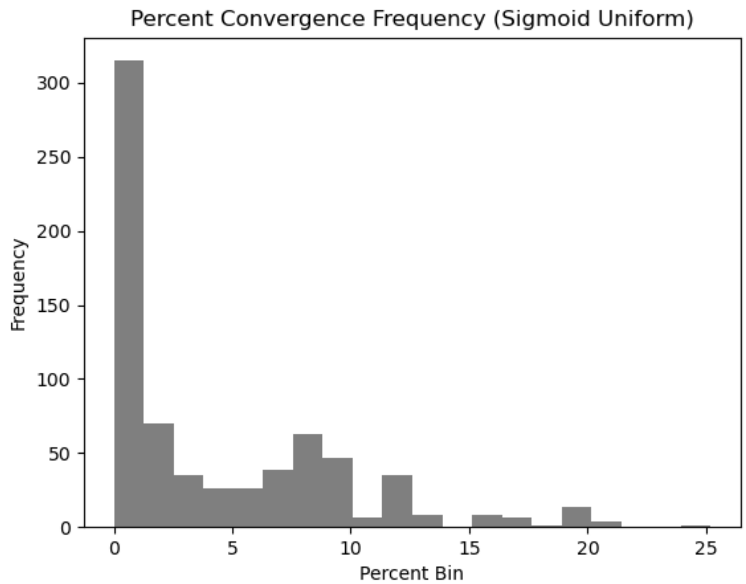

Figure 29:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

2.3.2 Sigmoid with Noise: Normal

2.3.2.1 Sigmoid with Noise: Normal: Percent Savings

1. Average Percent Savings: 45.9% (μ) ± 14.6% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

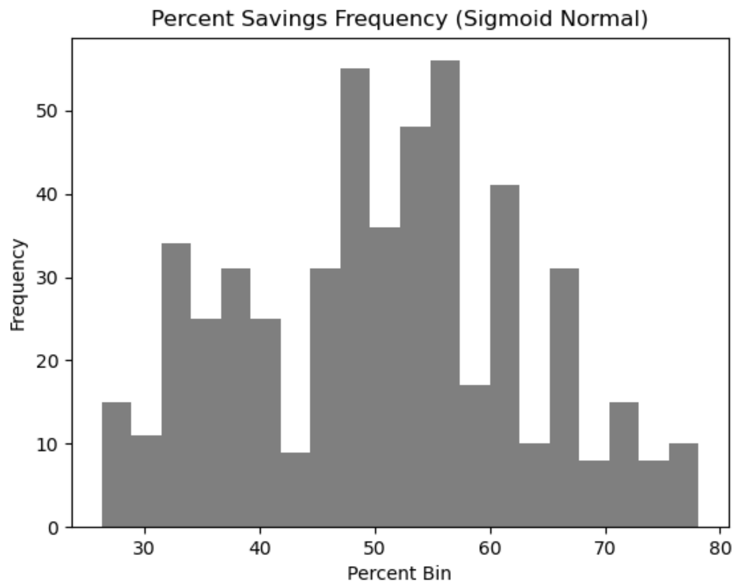

Figure 30:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

2.3.2.2 Sigmoid with Noise: Normal: Percent Convergence

1. Average Percent of Convergence: 4.2% (μ) ± 5.1% (σ)

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

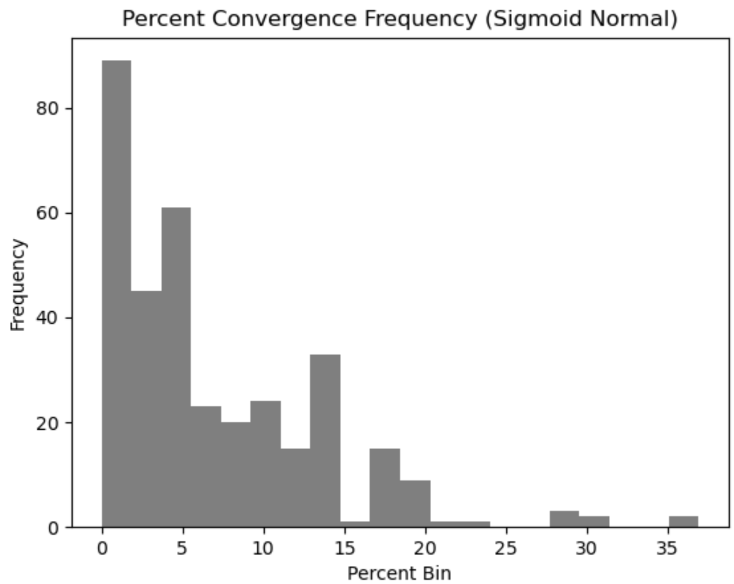

Figure 31:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

3 Analysis: Toy Grid

3.1 Analysis: Toy Grid: Lower Bound of Demands

3.1.1 Lower Bound of Demands: Percent Savings

1. Average Percent Savings: 14.6% (μ) $\pm$ 10.1% (σ)

2. Number of Satisfied Constraints: 256

3. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

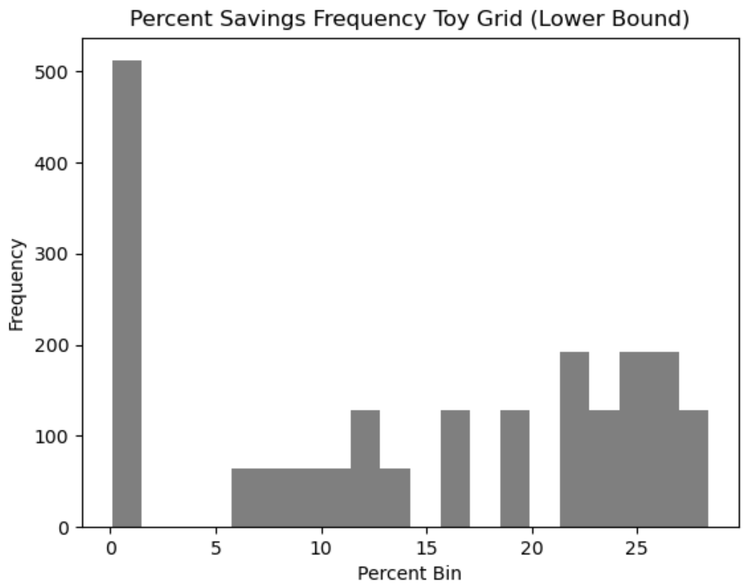

Figure 32:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

3.2 Analysis: Toy Grid: Average of Demands

3.2.1 Average of Demands: Percent Savings

1. Average Percent Savings: 13.9% (μ) ± 10.1% (σ)
2. Number of Satisfied Constraints: 256
3. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

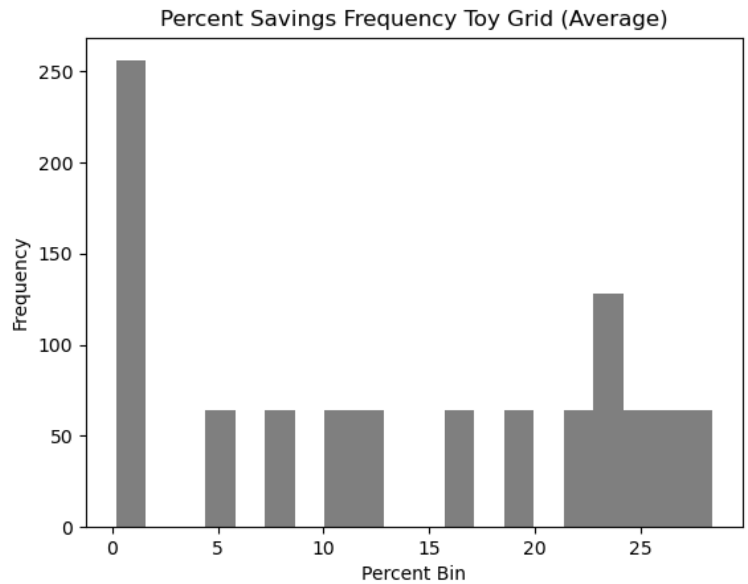

Figure 33:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

3.3 Analysis: Toy Grid: Upper Bound of Demands

3.3.1 Upper Bound of Demands: Percent Savings

1. Average Percent Savings: 14.6% (μ) ± 10.0% (σ)

2. Number of Satisfied Constraints: 192

3. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

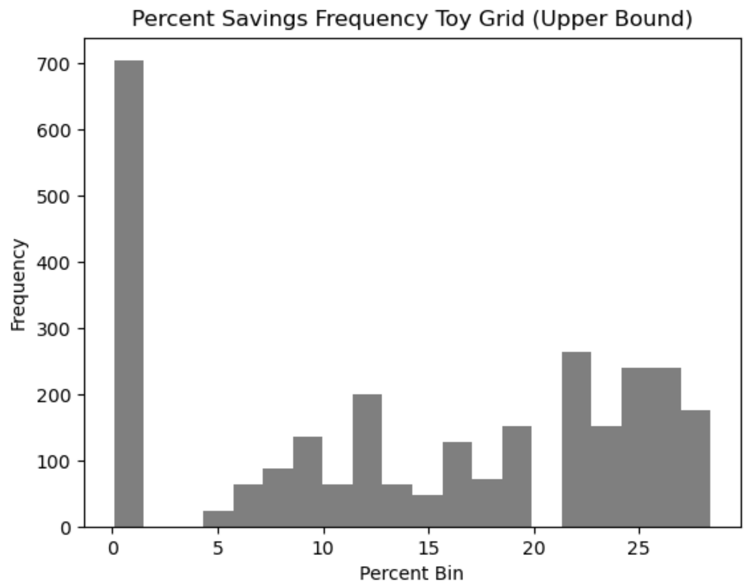

Figure 34:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

Part V
Discussion

Part Contents:

1. Overview
2. Introduction
3. Model
4. Method
5. Analysis

> *"Ultimately, man should not ask what the meaning of his life is, but rather he must recognize that it is he who is asked. In a word, each man is questioned by life; and he can only answer to life by answering for his own life; to life he can only respond by being responsible."*
>
> – Viktor Frankl, *Man's Search for Meaning*[17]

1 Discussion: Overview

1. The purpose of this chapter is to provide a less structured and more reflective commentary regarding the thesis as a whole.

2 Discussion: Introduction

2.1 Discussion: Introduction: General Problem

1. The general problem of climate change is difficult to express in one section of one chapter of one thesis. The topic involves the entire world, all of its inhabitants, and all the dynamics. The way that the **General Problem** section of the **Introduction** of this thesis is presented is meant to demonstrate the logical connection from the global problem of climate change to the Canadian problem of climate change.

2.2 Discussion: Introduction: Specific Problem

1. Similar to the **General Problem**, the **Specific Problem** that is presented in this thesis is, in many ways, an over-simplification of the Canadian problem of climate change; however, the purpose is to demonstrate the connections from the broader problem to the finer aspects of the problem and how it ultimately connects to the thesis problem of upgrade scheduling optimization.

2.3 Discussion: Introduction: Hypothesis

1. It is possible to argue that the evaluation of this null hypothesis is trivial – perhaps even meaningless. However, in the event that the evaluation fails to reject the null hypothesis, it would be clear that other means of optimization should be considered.
   (a) Demonstrating the ineffectiveness of this model would not resolve the problem of upgrade scheduling.
   (b) The resources needed to find and implement an effective optimizer are low compared to the potential return on investment, considering the significant cost of {non-optimized, on-demand} methods of upgrading that are currently in place.
2. The evaluation of the null hypothesis accurately depicts the stage of development of the model – a proof of concept. This does not mean that other aspects of the model are not explored in the discussion. Other aspects are explored, but they are not used to make definitive statements

regarding the capability of the model. This is in contrast to the conclusion of the hypothesis, which is definitive.

3 Discussion: Model

1. The method of approximation that is used in the current version of the model is also used in convolutional neural networks. In convolutional neural networks, this process is called max-pooling. In neural networks, max-pooling is used to identify prevailing properties in relation to a sliding frame of reference over the input (commonly a sliding window across an image) during forward-propagation to enable optimization during back-propagation. However, in the context of this thesis, this method of approximation is used to identify the prevailing properties of clusters (such as transformers with similar energy demands) to enable brute-force optimization.

   (a) The effectiveness of this method under various conditions is discussed in the **Discussion: Analysis** section.

3.1 Discussion: Model: Descriptive Layer

1. There are several differences between the description layer of the model compared to the technical layer. The reason this is done is due to the difficulty in abstracting technical aspects of the model without introducing sources of confusion.

2. For instance:

   (a) In the descriptive layer: The optimization of upgrade schedules in stage one is described in a single method.

   (b) In the technical layer: The optimization of upgrade schedules in stage one is described in seven methods.

3. A similar reason for the differences is that the purpose of this layer is fundamentally different. The descriptive layer is meant to provide a means of conceptualizing the model as it relates

to problems such as electrical infrastructure. Whereas in the technical layer, the purpose is to demonstrate the underlying mechanics of the model.

(a) The **descriptive layer** is to a **user guide** as the **technical layer** is to a **technical manual**.

3.2 Discussion: Model: Technical Layer

1. What is apparent in this layer of the model is that some of the sub-algorithms {referred to as methods} process input in an exponential search space. This size of computational complexity is not necessarily considered scalable in all contexts. There are multiple reasons that the model is not disregarded as a scalable optimizer for upgrade scheduling.

   (a) The model is only in the initial stages of development.

      i. The k-means clustering algorithm is an optimizer that computes results approximately. The core algorithm without helper functions is nearly 1000 times slower than optimized k-means clustering algorithms that include helper functions.

      ii. The potential development of helper functions and algorithm speed-ups is a core reason that the model presented in this thesis is not discarded. There are indications that such algorithm optimizations exist and have been conjectured by Prof. van Veen and Prof. Lewis; however, these methods have not been developed for this version of the model, so they are not discussed in any detail.

   (b) Depending on the size of the input and computational resources, it is possible that the exponential search space is manageable, even in the current implementation of the model.

3.3 Discussion: Model: Rigorous Layer

1. The format of the proof is fairly exotic (in a manner of speaking); however, the argument being made in this format is that it can be communicated, critiqued, and debugged line by line more effectively than other nonstandardized formats.

   (a) This is not to say that other formats are not effective in communicating logic; however, there can be aspects of ambiguity.

(b) The term exotic refers to the proof format not following common conventions used in mathematical analysis.

(c) A more standardized format modelled after tried-and-true programming might be a viable solution to making certain concepts more accessible – at least to those that have obtained the ability to read code.

4 Discussion: Method

4.1 Discussion: Method: Random Data

1. Regarding the method of measuring model performance: the difference between levels of approximations is used as a relative measure of performance, as opposed to measuring performance by the difference to the exact solution – or rather the most optimal solution with the grid at the highest level of resolution (number of groups equal to the number of nodes).

   (a) One reason this is not used is that with a sufficiently large number of groups, the problem becomes intractable – the very issue that necessitates the existence of the PTAS that is this model.

   (b) Another reason is that the exact solution can exist anywhere from the approximation bounds to a zero vector. So measuring the exact solution makes the approximately optimal solution a weaker result since the approximation bounds hold as a solution to any problem that would create the same result.

2. The method to test the above conjecture would involve incrementally varying the deviation parameters of the randomly generated data.

4.1.1 Discussion: Method: Random Data: Uniform Distribution

1. Although the sigmoid is the most realistic distribution among these sub-methods, the uniform distribution is the most robust evaluation of the model.

   (a) This is because input sampled from a uniform distribution does not contain any structure to influence potentially anomalous properties of the model.

(b) In other words, it is possible that certain structures could cause the model to output a rare result that would suggest that the model has a higher level of performance than it might typically have.

4.1.2 Discussion: Method: Random Data: Normal Distribution

1. This method contains more structure than the uniform distribution sub-method; however, it is still fairly robust in terms of capturing a large range of potential stage two schedules. In particular, the conditions in, which the schedule costs are on average equal, but where the deviation decays exponentially.

4.1.3 Discussion: Method: Random Data: Sigmoid with Noise

1. As mentioned above, while the sigmoid method has the most elements of realism, it has the largest potential for bias in the results.

4.2 Discussion: Method: Toy Grid

1. The construction of more realistic grid demands requires the use of objects such as data, statistics, and tools such as models of grid demands. These objects most likely already exist within various (governmental and non-governmental) organizations; however, access is typically restricted for reasons such as privacy, intellectual property, and trade secrecy.

2. For example:

    (a) Equipment efficiencies:

    i. Equipment efficiency measurements are standard protocol in most organizations and are typically required by law to maintain acceptable levels of safety; however, it is not necessarily true for accurate modelling of efficiency curves for each physical piece of equipment. Fortunately, obtaining this data structure is not especially complex. It can be done by direct measurements and creating curves of best fit or, potentially, by simulating efficiency curves using any number of software solutions –, which may be sufficiently accurate for the purpose of upgrade scheduling optimization models.

    (b) Equipment inventories:

   i. Maintaining a thorough inventory is a logistics problem that most organizations actively address as it is a crucial aspect to optimizing the accomplishment of objectives efficiently and effectively. In other words, it is standard practice for nearly all organizations, such as companies in the electrical energy industry. All this to say, the datasets of equipment inventories are already in play and ready for input into a model such as the one presented in this thesis.

(c) Equipment acquisition agreements and catalogues:

   i. There exist instances of well-maintained catalogues of equipment and streamlined acquisition processes, but this is most likely less common among all organizations. As organizations scale in size, the cost of a dedicated acquisitions team, or even department, becomes justifiable. Smaller organizations have less ability to specialize and require members to serve multiple roles. This means the datasets of equipment catalogues are not necessarily ready for input into the model.

(d) Detailed household energy projections:

   i. There exists a vast array of methods for predicting future energy usages, such as statistical learning, neural network learning and other types of machine learning systems. These methods can be validated through processes such as measuring the accuracy of the model on historical data. Although smart grids are becoming more prevalent in major cities, it is not a guarantee that smaller areas will already have predictive models in place. Projecting energy demands is fundamental and could potentially require a significant amount of resources to establish a detailed accounting of energy use behaviours in order to construct a predictive model. These behaviours could include the patterns of appliance usage, electrification of appliances, and transitioning to EV.

(e) Statistical (even causal) models for the behaviour related to energy usage as it depends on government policies and programs:

   i. The models might need to be constructed; however, all the necessary ingredients may already exist. This is because of the utility of tracking data. Not as it pertains to this model, but as it pertains to the interests of various organizations. government organizations may accumulate data for the purpose of monitoring the progress of certain programs. Non-government organizations may accumulate data for the purpose of tracking job tickets. Regardless of the reasoning, it might be possible to create detailed projections at the transformer level without the development of new data collection systems.

5 Discussion: Analysis

1. Aside from identifying the aspect of results from the methods of analysis that are capable of rejecting the null hypothesis, the following discussion is relatively speculative. The subjective points that are made are intended to communicate the model, its properties, and its performance from the perspective of a model developer. The justification for including subjectivity is that it could potentially provide useful insights into the future development and understanding of the model.

5.1 Discussion: Analysis: Random Data

1. The purpose of these methods of analysis is not meant to compare the performance of the model across various sources of data with noise to claim that one is better than another.

2. The point is slightly more subtle: In a real-world application, the source of data is not a matter of selection. This is to say that, in a circumstance where a future version of the model presented in this thesis is implemented, and the inner properties are better understood, it is potentially possible to determine, which optimization model is best suited through analysis of the input data structure.

3. For instance:

    (a) It is possible that there are conditions in, which the Monte Carlo method could significantly outperform the model presented in this thesis if the input data structure is essentially structureless – indistinguishable from a uniform distribution.

    (b) This instance is to illustrate the purpose of analyzing the performance of the model under various conditions and is not a definitive conclusion on model effectiveness in relation to the Monte Carlo method.

5.1.1 Discussion: Analysis: Random Data: Uniform Distribution

5.1.1.1 Uniform Distribution: Unsorted

Uniform Distribution: Unsorted: Percent Savings

1. Average Percent Savings: 29.4% (μ) $\pm$ 7.2% (σ)

   (a) The interpretation of this result is minimal. In essence, it demonstrates that under chaotic conditions, the model is capable of producing optimized results through approximation. If there are not any budget constraints that the model can satisfy, the mean and standard deviation would both be exactly zero.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

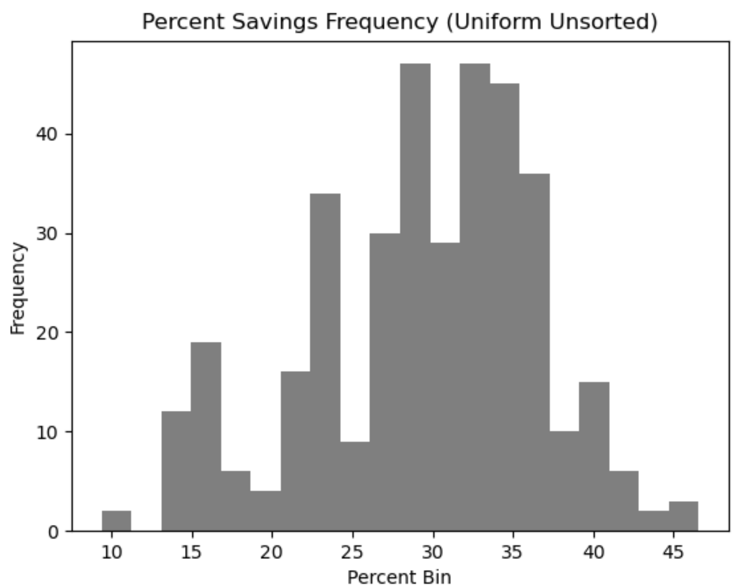

Figure 35:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. Converting datasets into histograms come at the cost of precision; however, it does reveal some semblance of structure in the output despite the input being structureless. This is to say that

there does appear to be a meaningful centre of mass, as opposed to a distribution with a split distribution with the centre of mass having a frequency near or at zero.

Uniform Distribution: Unsorted: Percent Convergence

1. Average Percent of Convergence: 3.9% (μ) $\pm$ 3.5% (σ)

   (a) The interpretation of these values is similar to the average percent of savings; however, it demonstrates a different property model. In particular, under these conditions, there is a nonzero percent of convergence from one approximation group size to a larger approximation group size.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results:

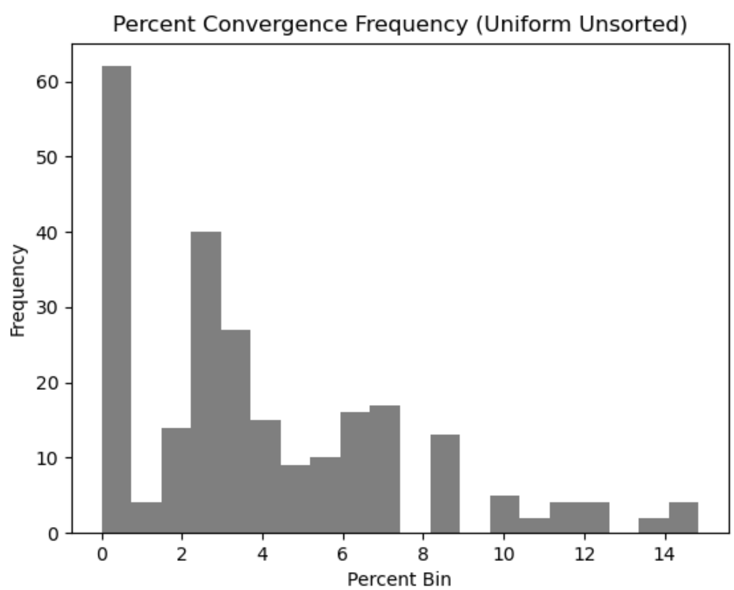

Figure 36:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

1. In Figure 36, the convergence distribution appears to have an exponential decay, where the

most common result is zero convergence and a rapidly tapering-off in frequency as the percent of convergence increases. The exact explanation of this is as complex as the model itself; however, an underlying principle to partially explain this behaviour is in the method of approximation. The use of max-pooling in a discrete problem, such as in this method, creates gaps or steps in the level of optimization that can be achieved in certain circumstances.

2. For instance:

   (a) Approximation of a set of nodes with four groups creates budget constraint bounds per group that are applied to the individual nodes within each respective group. The unpacking of these budget constraint bounds results in the lowest possible schedule costs for each node. In a lower resolution approximation such as two node groups, if the budget constraint bounds are greater than the lowest possible schedule costs from a four group approximation, then the exact same result will be achieved.

3. The apparent structure of this distribution of nonzero convergence values demonstrates that convergences exist with the input being from a uniform distribution.

5.1.1.2 Uniform Distribution: Sorted

Uniform Distribution: Sorted: Percent Savings

1. Average Percent Savings: 26.1% (μ) $\pm$ 6.5% (σ)

   (a) Although the overall results of this method contain a substantial amount of information to unpack, the result in the form of a mean and standard deviation statistic is not particularly remarkable except that it demonstrates that the monotonically increasing (sorted in ascending order) property of the data structure does not appear to significantly change the outcome of optimization process with respect to the unsorted data structure and the corresponding outcome.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

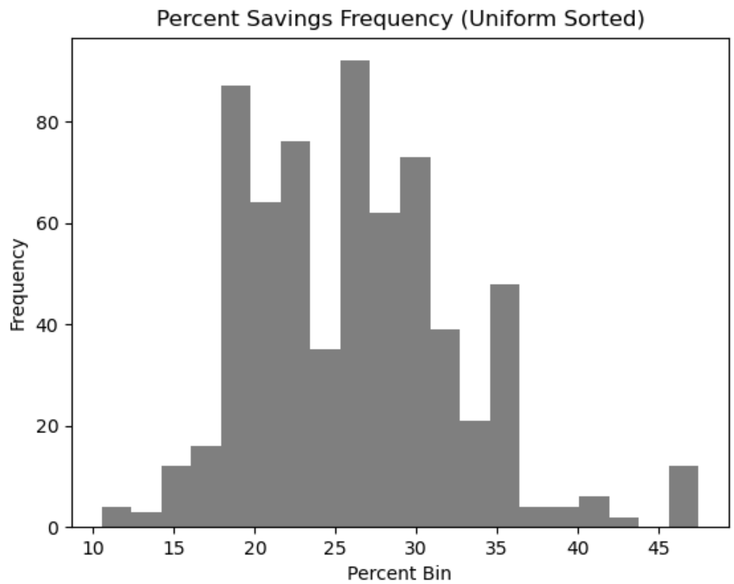

Figure 37:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. In Figure 37, the distribution is nearly indistinguishable from the distribution from Figure 35. Although this statement regarding the two distributions is subjective, a statistical test would also be inherently subjective at this level of understanding of the model and the results that it produces. This is not to take away from statistical rigour or the objective results it produces from data; Rather, it is meant to justify not using such definitive tools of analysis before a well-formulated hypothesis can be established.

   (a) To underline this point, the creation of the dataset is from a random uniform distribution that is sorted in ascending order. If samples are taken from this dataset, the mean and standard deviation can be described with a normal distribution (Central Limit Theorem). However, the data points within the dataset: are at some moments in the algorithm entirely independent, while in other moments, dependent; and perhaps most importantly, under budget constraints that may or may not be satisfiable.

   (b) Even if a statistical test determines that the two distributions are the same, it is not clear that {a sorting transformation has no significant effect on optimization outcomes} is a reasonable conclusion to make.

Uniform Distribution: Sorted: Percent Convergence

1. Average Percent of Convergence: 2.1% (μ) $\pm$ 2.3% (σ)
   (a) The analysis of these values is similar to the commentary regarding average percent savings. These values are both non-zero.
   (b) It is worth mentioning the flaw of using mean and standard deviations when working with percentages, especially when the standard deviation is a larger value than the mean since it suggests negative percentages. The justification for its use in these methods is that it is a (more understandable) common convention, and the lack of normality assumptions is explicitly stated.
      i. Even deeper, the use of variance implies symmetry in the mean. Special context is required to meet this criterion – beyond the existence of symmetry in the problem.
      ii. For instance, it is common practice to use a normal distribution to describe a non-normal distribution that might pass a normality test; however, knowing when this is acceptable is by no means rigorous.
2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

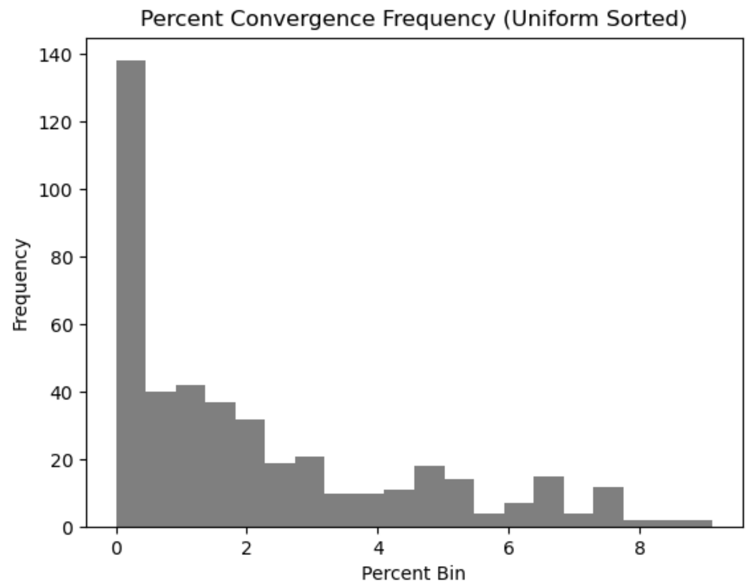

Figure 38:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

1. In Figure 38, the distribution appears to follow the exponential decay as in Figure 36; however, the change in frequencies is less gradual. It is not clear whether or not this is an anomalous behaviour that is inherent to the data structure is sorted. A substantial number of runs of this method might be needed to make this determination. Most importantly, though, there does appear to be a significant amount of nonzero convergence values.

5.1.2 Discussion: Analysis: Random Data: Normal Distribution

5.1.2.1 Normal Distribution: Unsorted

Normal Distribution: Unsorted: Percent Savings

1. Average Percent Savings: 74.8% (μ) $\pm$ 6.6% (σ)

   (a) What is immediately clear here is that the average percent savings are roughly three times larger than the preceding methods. Ultimately, the results demonstrate that there exist nonzero percent savings, which means that the model is capable of producing optimization results from a data structure that is sampled from a normal distribution.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

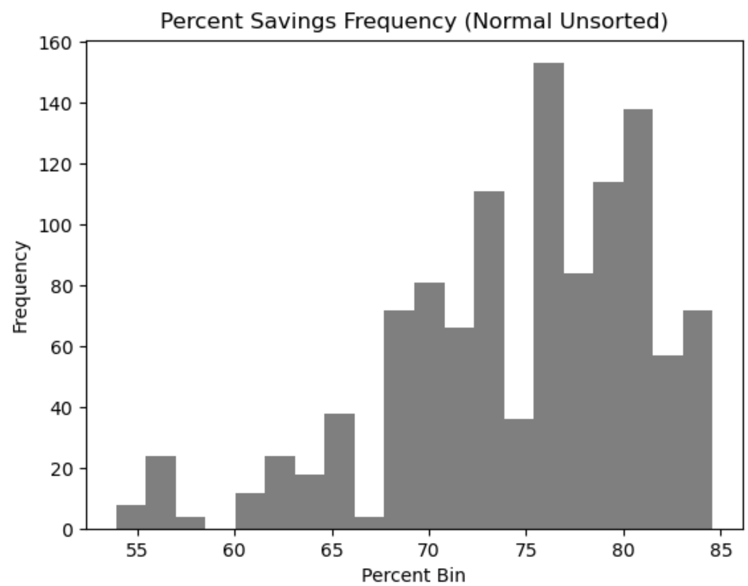

Figure 39:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. In Figure 39, the distribution appears to be roughly normal with a skew. However, this might be a result of the output data points being truncated between 0% and 100%, or, perhaps, a result of the input data points being from a normal distribution centred at $\mu = 0.5$ and truncated at two standard deviations $\sigma = 0.125$.

2. Regardless, the distribution demonstrates that the model is capable of producing optimization results with the input data structure being from a normal distribution.

Normal Distribution: Unsorted: Percent Convergence

1. Average Percent of Convergence: 0.1% (μ) ± 0.4% (σ)

(a) The interpretation of {the average percent of convergence being near-zero with a near-zero deviation} requires nuanced discussion. At first glance, the results appear to indicate that the model breaks under these conditions. More depth is provided beneath Figure 40.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

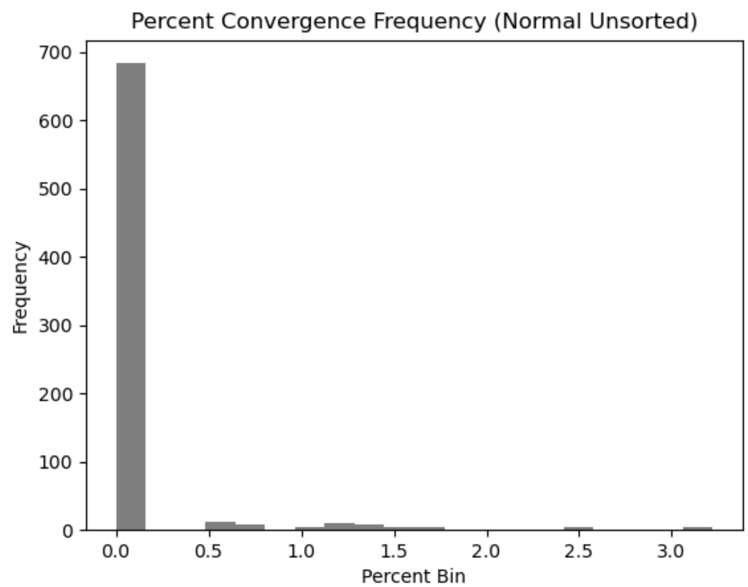

Figure 40:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

1. In Figure 40, the outliers of the result appear to be the nonzero data points. In the context of the significant results from the percent savings, the explanation might be that the data structure from a normal distribution is conducive to optimization at exceptionally low resolution. This would explain the apparent lack of convergence at higher resolutions.

2. In other words, under these conditions, the model appears to have a property of diminishing returns with increases in resolution. This is ultimately an encouraging result for the model

when the input data structure is {from a normal distribution or significantly similar to a normal distribution}.

5.1.2.2 Normal Distribution: Sorted

Normal Distribution: Sorted: Percent Savings

1. Average Percent Savings: 74.8% (μ) ± 7.2% (σ)

   (a) The average percent savings and deviation appear similar to the results from the unsorted-normal method; however, the distribution of data points from this sorted-normal method is significantly different.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

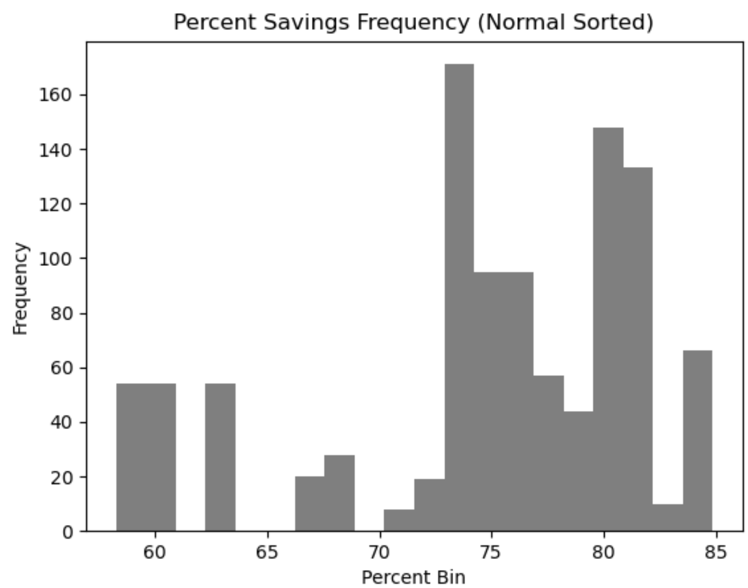

Figure 41:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. The histogram plot in Figure 41 appears to have some structure, but with a substantial amount of noise. It is exceptionally unlikely that the structural difference from previous results is a statistical outlier.

   (a) The reason "a statistical outlier" is "exceptionally unlikely" is because the sample size {for each of the 10 realizations is 128}. This would require that the set of {1,280 random numbers from a normal distribution} not pass a normality test. How large of an outlier this would depend entirely on the method of normality testing.

      i. These methods are not typically discussed in terms of reverse-engineering the bounds of normal anomalies. Further, these methods have elements of subjectivity.

      ii. This is an unavoidable, but manageable part of statistics. Some of the theorems are constructed on the set of extended reals (infinitesimals, infinities, reals) in order to achieve an acceptable level of operational closure.

      iii. The Central Limit Theorem at the foundation of statistics describes the behaviour of an abstract object as the sample size approaches infinity. However, the real-world application of statistical mechanics can only be realized in a finite set of discrete units.

2. The input dataset is not a newly constructed object. It is a sorted copy of the input dataset from the previous method (unsorted-normal). This means that the model is not invariant to the sorting transformation of input data. There is also no indication that the model is not invariant to other transformations.

Normal Distribution: Sorted: Percent Convergence

1. Average Percent of Convergence: 0.1% (μ) $\pm$ 0.3% (σ)

   (a) The interpretation of these values is essentially identical to the values from the unsorted-normal method. The mean and standard deviation are near-zero.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

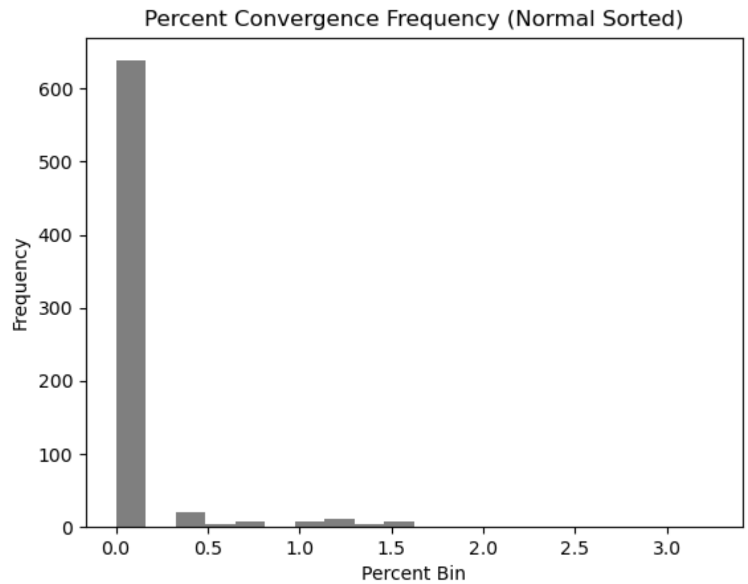

Figure 42:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

1. The histogram in Figure 42 is similar to the histogram in Figure 40, except the frequency of non-zero convergence data points is slightly higher. Ultimately, this distribution suggests that the optimization of outcomes is relatively independent of the input resolution.

5.1.3 Discussion: Analysis: Random Data: Sigmoid with Noise

1. Although the sigmoid input data has more noise than what would likely occur in a real-world application, it does contain strong elements of realism. Specifically, in the context of a system with significant monotonic changes (such as an electrical grid and EV transition), large increases in time series demands cause large increases in the time series of schedule costs. A function that adequately expresses this type of behaviour is the sigmoid function.

5.1.3.1 Sigmoid with Noise: Uniform

Sigmoid with Noise: Uniform: Percent Savings

1. Average Percent Savings: 45.9% (µ) ± 14.6% (σ)

    (a) These are encouraging values that demonstrate that the distribution of data points, regardless of normality, is near the centre of mass and is most certainly not a set of nonzero values.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

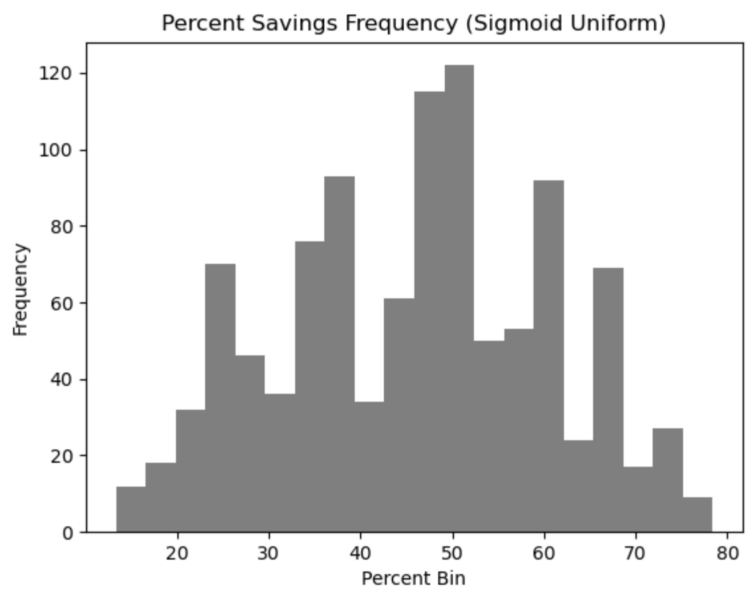

Figure 43:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. The distribution in Figure 43 appears normal; however, there also appears to be an oscillating pattern between higher and lower frequencies along with the domain of percent bins.

2. A (speculative) reason that the distribution in Figure 43 appears to be normal might be due to the relationship between the probability distribution function (PDF) and the cumulative distribution function (CDF) of a normal distribution.

(a) The CDF function of a normal distribution is a sigmoid, just as the dataset of this method is.

(b) The PDF function of a normal distribution is a normal distribution, just as this method distribution of data points appears to be.

(c) Recalling from the method settings, the {average point of inflection for all the sigmoids} is in the middle of the time series upgrade window, just as the distribution in Figure 43 appears to be centred within the bounds.

3. This might be entirely coincidental since (i) the creation of normal distributions only requires sufficiently large random samples, and (ii) the unsorted-uniform method also produces a distribution of data points that appear fairly normal, but do not share the same properties mentioned above regarding the PDF and CDF of a normal distribution.

Sigmoid with Noise: Uniform: Percent Convergence

1. Average Percent of Convergence: 4.2% (μ) $\pm$ 5.1% (σ)

    (a) By the same logic that is expressed in the sorted-uniform method discussion, a standard deviation that is larger than its mean can imply the existence of negative data points, which, in this context, is meaningless for percentages that are strictly greater than or equal to zero.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

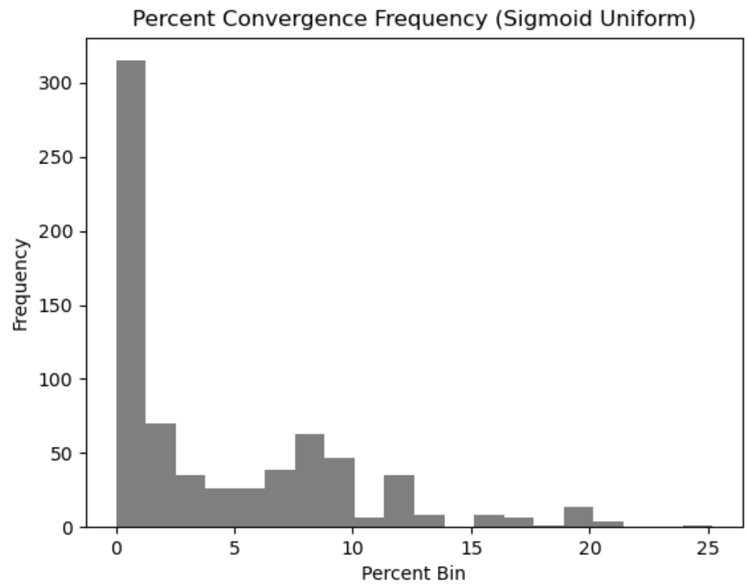

Figure 44:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

1. The distribution in Figure 44 appears to have exponential decay; however, there is possibly a slight oscillation.

5.1.3.2 Sigmoid with Noise: Normal

Sigmoid with Noise: Normal: Percent Savings

1. Average Percent Savings: 50.7% (μ) ± 12.3% (σ)

   (a) The source of noise for the sigmoid methods does not appear to have a significant effect on the average percent savings. At the very least, the model is capable of producing optimal output with input data from sigmoids with normal noise.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

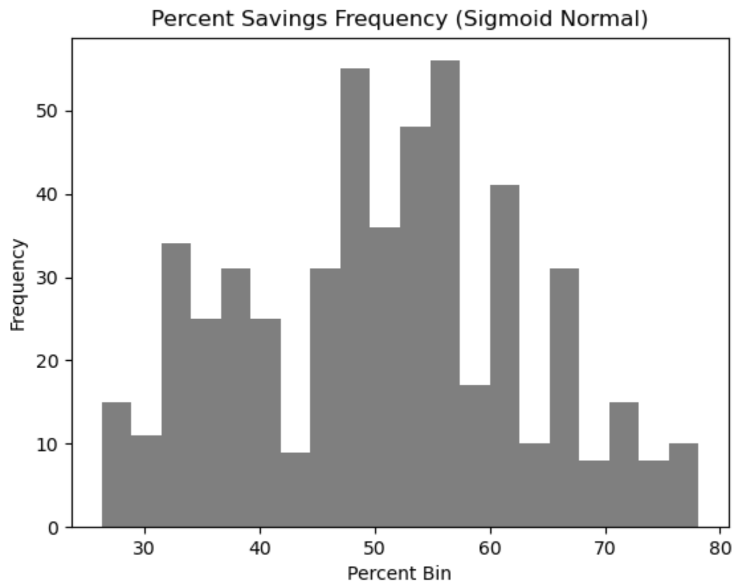

Figure 45:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. The distribution in Figure 45 is remarkably similar to the uniform-sigmoid method distribution; however, the oscillation of frequencies appears to be less consistent.

Sigmoid with Noise: Normal: Percent Convergence

1. Average Percent of Convergence: 6.9% (μ) $\pm$ 6.6% (σ)

   (a) These values appear to be slightly more substantial than the mean and standard deviation from the uniform-sigmoid method, although the difference is not particularly prominent.

2. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

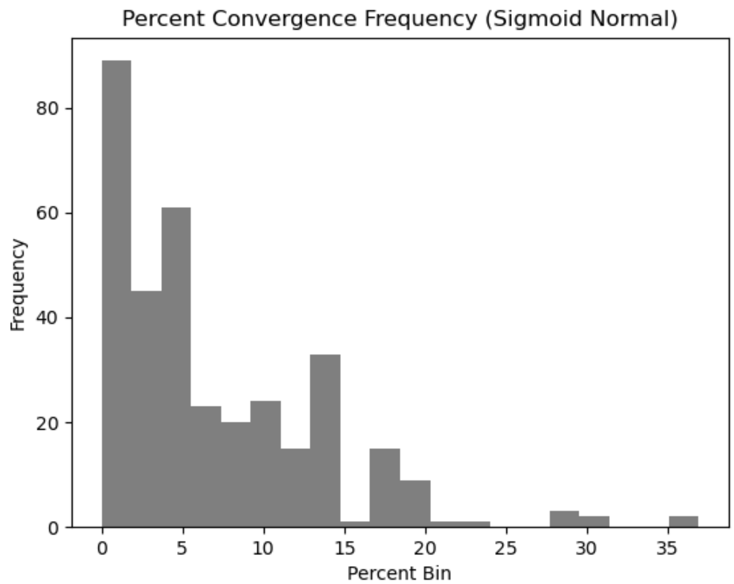

Figure 46:
This is a plot showing the frequency of percent convergence for each set of budget constraints that are satisfiable from a lower resolution to a higher resolution with twice the number of group approximations.

1. The distribution in Figure 46 decay and oscillation attributes appear to be more pronounced than the distribution in Figure 44 from the uniform-sigmoid method.

2. This is not to say that the apparent decay is steeper; rather, the decay in this method is more gradual with less noise.

5.2 Discussion: Analysis: Toy Grid

1. The toy grid method is important for the sake of conceptualization and communicating the workings of the model, but it is also important as it demonstrates the functionality of the model from input to output. This is in contrast to the random-data method, where stage one is skipped.

2. In consideration of the three sub-methods (lower bound of demands, the average of demands,

and upper bound of demands) representing one statistical object, the analysis of the results will accumulate as the discussion progresses through each sub-method.

   (a) In other words, each subsequent sub-method is in the full context of the previous sub-method(s).
   (b) This is in contrast to the random-data methods where the {accumulation of context} is regarding results that are entirely independent and not strictly necessary.
   (c) Whereas, in this method, the results are entirely dependent, and the context is strictly necessary for a thorough analysis.

3. Although the results of the toy grid method do not represent any real-world application, one thing is apparent: if the assumptions regarding the cost attribute dynamics (such as transformer purchase, installation, and removal) hold (i.e. prevailing proportionalities such as some quantity being larger than another; or), then the most robust solution to an impending grid overload is to focus on localized energy storage (battery type or otherwise) and isolated charging stations. This is not to say that the model is invalid. What it means is that there is a larger set of possible equipment to upgrade.
   (a) Localized energy storage enables the levelling-out of instances of demand. This means that the sharp increases in demand due to EV charging are handled by stored energy, and the noncharging hours are used to spread out the demand from the transformer over the largest amount of time. Transformers will need to be upgraded with time still; however, this would resolve the need for exceptionally large transformers that would be needed to accommodate the otherwise un-levelled demand. Similar, perhaps even complementary, ideas are coordinated charging technology that gives chargers time slots to use the grid.
      i. Localized energy storage means at or near customer service entrances, such as battery storage for one household or a transformer for several households.
   (b) Although at-home charging is by far the most convenient charging option, the development of rapid charging technology opens the door to isolated charging stations being a scalable solution. The reason why isolated charging stations are a scalable solution is similar to the reason why electrical substations are scalable.

i. Pad-mounted transformers are simpler to install or remove compared to pole-mounted transformers.

ii. Supporting equipment becomes more economical at larger scales. For example, work to upgrade transformers at an isolated charging station can be done during demand lulls by disconnecting the relevant equipment while keeping the rest of the equipment connected. Conversely, residential transformers can cause a significant disturbance to the local area. For instance, in hot climates, pet owners cannot have their air conditioning off for several hours.

iii. The step-down residential transformer design is under-engineered for the task of handling the electrical demands that come with transitioning to a 100% EV market. Without EVs, a properly rated transformer for a residential area is able to maintain a reasonable percentage of efficiency. With EVs, selecting a properly rated transformer for a residential area is not necessarily possible with any standards of efficiency or standards of safety.

4. A potential compromise between {unregulated at-home charging and charging restricted to station charging} could be that at-home charging would be allowed if (i) battery storage is installed and (ii) the energy demand of each battery storage unit is coordinated. These measures would be to prevent any load spikes beyond transformer capacity. This would also decrease the need for substantially large transformers.

5.2.1 Analysis: Toy Grid: Lower Bound of Demands

5.2.1.1 Lower Bound of Demands: Percent Savings

1. Average Percent Savings: 14.6% (μ) $\pm$ 10.1% (σ)

   (a) This mean and standard deviation of percent savings are not particularly indicative of model performance with pseudo-realistic input, except that it is clear there does exist non-zero percent savings in reference to budget constraints.

2. Number of Satisfied Constraints: 256

   (a) Of all the 256 sets of constraints that are evaluated, the model finds an optimal combination of schedules that satisfy each constraint.

3. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

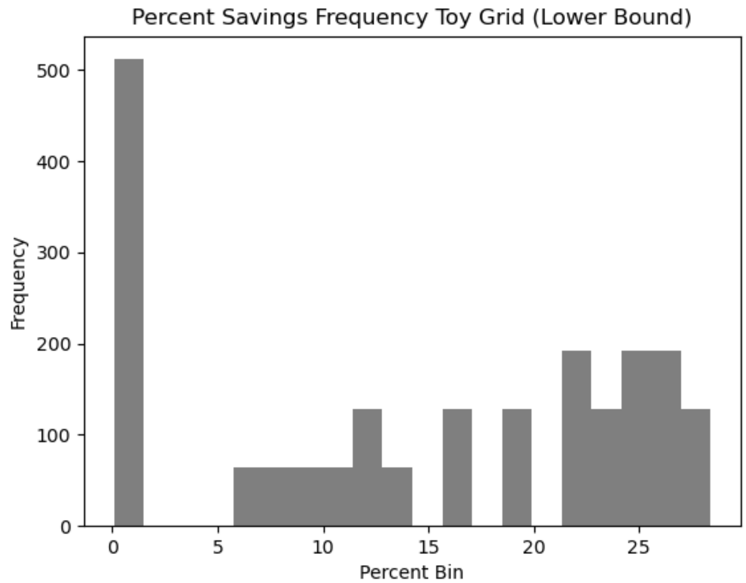

Figure 47:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. In Figure 47, it is not clear what, if any, underlying structure exists except that there is a slight curve in frequencies. To interpret this as a skewed normal distribution is a stretch – especially when taking into account the large frequency at the zero percent bin.

5.2.2 Analysis: Toy Grid: Average of Demands

5.2.2.1 Average of Demands: Percent Savings

1. Average Percent Savings: 13.9% (μ) $\pm$ 10.1% (σ)

    (a) The average of demands contains elements that are strictly larger than the corresponding elements of the lower bound of demands. It is a reasonable expectation that the average percent savings would be less than or equal to the average percent savings of the lower bound of demands; however, far more work and computational resources are needed to achieve such a conclusion since this toy grid method consists of only one instance of demands – as opposed to a sample of demands.

2. Number of Satisfied Constraints: 256

   (a) In this instance, the model satisfies all the constraints; however, it is conceivable that there exist conditions where the average of demands satisfies fewer constraints than the lower bound of demands.

3. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

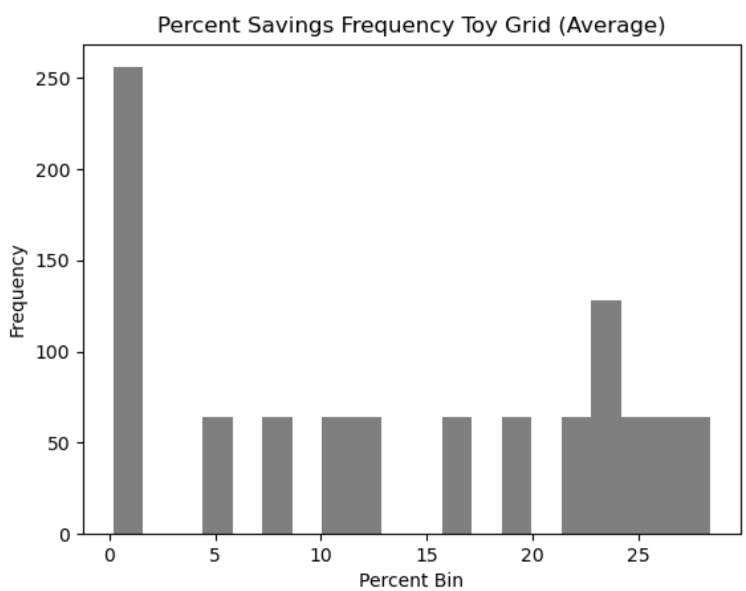

Figure 48:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. The distribution in Figure 48, is noticeably different from the distribution in Figure 47. In particular, the structure of the nonzero bin frequencies is uniform, with the exception of two bins. It is unclear exactly what is causing this behaviour since it deviates substantially from all other percent savings histograms.

2. As peculiar as the distribution in FIgure 48 is, it still demonstrates definitively that the model produces optimal output with an average of demands as input.

5.2.3 Analysis: Toy Grid: Upper Bound of Demands

5.2.3.1 Upper Bound of Demands: Percent Savings

1. Average Percent Savings: 14.6% (μ) $\pm$ 10.0% (σ)
   (a) The average percent savings in this instance is deceptive as it appears to suggest that this method produces better results despite the upper bound of demands being strictly larger than the preceding sub-method inputs.
2. Number of Satisfied Constraints: 192
   (a) Only 192 of the 256 sets of constraints are satisfied, and this is the context that is needed to properly interpret the average percent savings being larger than those from the other sub-methods.
   (b) It is not required that the average percent savings be larger in all circumstances where all the constraints are not satisfied; however, it is possible as a result of a bias due to truncation.
      i. When the model is unable to produce results that satisfy a set of constraints, an error is raised, and the next set of constraints is evaluated.
      ii. In other words, if the model includes the calculations from the unsatisfiable constraints, there are negative percent values. The inclusion of these values reduces the overall percent average.
3. The following is a plot to demonstrate the distribution of results behind the mean and deviation results above:

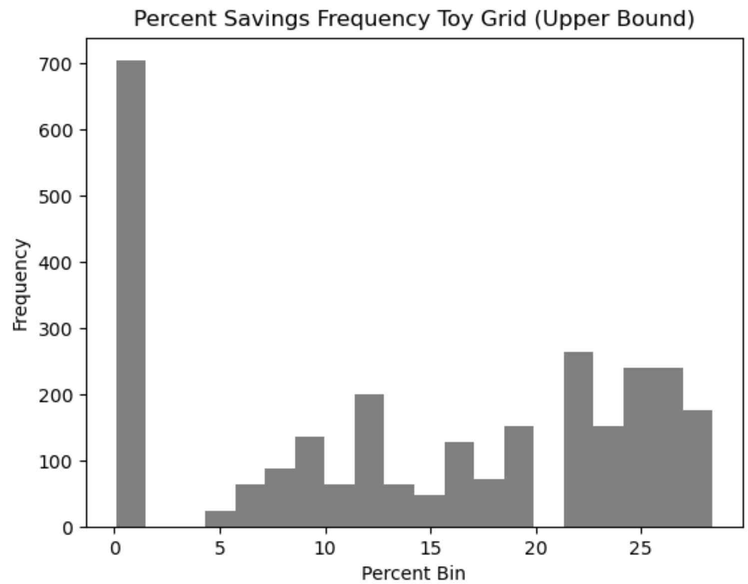

Figure 49:
This is a plot showing the frequency of percent savings for each set of budget constraints that are satisfiable.

1. The distribution in Figure 49 appears to have a structure that is more similar to the lower bound of demands distribution in Figure 47 than it is to the average of demands distribution in Figure 48. This is to say that there is a slight curving without accounting for the zero percent bin.

2. The reason why there is this discrepancy in distributions is unclear. More realizations and variations in the number of groups could potentially demonstrate the aspects that produce these kinds of results. The reason these changes are not implemented is due to time and computational resource constraints.

3. A subtle point regarding the results of this toy grid method as a whole is that model maintains a level of consistency. If the model breaks under certain variations of conditions, it indicates a fundamental flaw in its structure. This is not to say that there are no edge cases that would break the model. Exceptions are an expected aspect of algorithms, especially methods of approximation.

Part VI
Conclusion

Part Contents:

1. Hypothesis Evaluation

2. Model

"ACT III SCENE I. Rome. Before the Capitol; the Senate sitting above.

[...]

CAESAR [To the Soothsayer]: The ides of March are come.

Soothsayer: Ay, Caesar;, but not gone."

– William Shakespeare, *The Life and Death of Julius Caesar*[45]

1 Conclusion: Hypothesis Evaluation

1.1 Conclusion: Hypothesis Evaluation: Null Hypothesis

1. In the context of the results from the methods of analysis, under all input data structure paradigms, it is clear that there exist conditions for, which the model of this thesis produces optimal upgrade schedule combinations with budget constraints.

2. To put it concisely, the results of the model analysis definitively **reject the null hypothesis**.

1.2 Conclusion: Hypothesis Evaluation: Alternative Hypothesis

1. The proof demonstrates that there exist theoretical conditions for, which the model will produce optimal results that converge to the exact solution as the resolution of the approximation increases; however, the exact properties of model input that are required to guarantee optimal results remain unresolved.

2. In other words, the result of the model proof rigorously **confirms the alternative hypothesis**.

2 Conclusion: Model

2.1 Conclusion: Model: Present

1. The status of the model in terms of application-readiness per stage:

   Stage One: The current implementation of the first stage of the model is only partially complete. The basic framework is essentially application-ready; however, the attributes and methods that are needed are unknowable without a specific application to reference. This being said, the framework that is fully developed has been significantly optimized.

   Stage Two: The current implementation of the second stage of the model is essentially application-ready. The input to the second stage is dependent on the first stage; however, the input data structure from the first stage is always the same, which allows the full development of the second stage. This being said, there exists a significant amount of potential improvements in the overall efficiency of the algorithm.

   Stage Three: The current implementation of the third stage of the model is also essentially application-ready for the same reasons as the second stage. The third stage is even closer to completion due to the fact that the algorithm requires only two algorithms that include unpacking optimization bounds and a brute force search optimizer. The efficiency of the algorithm is nearly entirely optimized since the overall time complexity is linear.

2. The results of the hypothesis evaluation appear to indicate that further research and development of the model may be justified.

2.2 Conclusion: Model: Future

1. The next potential steps for the model could include:

   (a) Evaluating the performance of the model could be comparing the results to that of other methods such as Monte Carlo, integer programming, and dynamic programming;

   (b) Determining how sensitive the model results are to changes in projected demands;

      i. For instance, consider a circumstance with 20 years of projected demands, but after ten years, the projected demands unexpectedly change. What are the consequences of the model results not corresponding to the unexpected change in demands?

   (c) Modelling the hyper-parameters for the model, such as the number of groups and how it changes the accuracy of the results to the exact solution;

   (d) Furthermore, extending the model to optimize across multiple objectives instead of just minimizing cost.

      i. For instance, consider a town that aims to minimize the cost of electric grid upgrades but is also planning to grow as a community. How does the model perform when the metrics are not just economical but include metrics such as social satisfaction?

2. If the results of future research and development demonstrate the worthiness of the model, it would be justifiable to begin applying the model to a variation of climate change problems that requires the optimization of upgrade schedules.

3. The problem that this model is abstracted from is the optimization of upgrade scheduling of electrical infrastructure. However, it is conceivable that other problems in climate change can also be solved with a future version of this model, such as:

   (a) The retrofitting buildings with electrical appliances.

   (b) The upgrading of automated EV (and other climate action technology) manufacturing systems.

   (c) The construction of zero-emission energy systems (i.e. solar, wind, battery storage, and nuclear power plants).

4. A potential area to observe regarding model performance is the comparable performance between two similar townships. The concept of the model is constructed on this premise of

approximation through the grouping of similar nodes. However, does optimizing one area that is similar to another area result in a similar solution? In other words, is it sufficient to apply the results from one township to other townships that are substantially similar?

(a) For instance, the study of the attributes of one city in terms of electrical infrastructure requires a certain amount of resources such as surveying, probability of EV transition over time, and energy projections. Once the model returns results, are there underlying principles that can be applied to other cities (as a whole or in part)?

Appendix

Appendix: Model: Overview

1. As mentioned in the **Model: Overview** section, the optimization of budgets is a common problem in civilization and even in nature. It is common in economics, but also in less traditional settings such as individual resource management. Although problems regarding the optimization of budgets come in many forms, they can be converted into a universal form known as **graph problems**.
   (a) For example, governmental and non-governmental organizations have limited resources to work with for any year, or quarter, to accomplish their goals. The way that an organization allocates resources to each department (or sub-organization) can mean the difference between achieving its goals or not achieving its goals.
   (b) A less traditional example would be a two-week adventure across Europe with a fixed budget. If the desired goal is to visit every major city in Europe exactly once and return home, what is the minimum amount of time and money needed to accomplish this goal?
      i. This problem is similar to a famous problem in discrete mathematics and computer sciences, which is known as the Travelling Salesperson Problem (TSP). In brief, in TSP, a salesperson wants to visit each house in a town in the shortest possible path, which returns to the starting point.
   (c) Each example from above (1a, 1b) is capable of being converted into a mathematical structure known as a **graph problem**. Graph problems consist of the following elements:
      i. The **graph** itself is a collection of vertexes (nodes) and edges (paths or node-to-node connections).

A. For instance, a directed graph with nodes (A), (B), and (C) with edges (AC), (CA), and (CB):

$$\begin{array}{ccccc} (A) & & & & (B) \\ \uparrow & & & \nearrow & \\ \downarrow & & \diagup & & \\ (C) & & & & \end{array}$$

B. For instance, an undirected graph with nodes (A), (B), and (C) with edges (AC) and (CB):

$$\begin{array}{ccccc} (A) & & & & (B) \\ | & & & \diagup & \\ | & & \diagup & & \\ (C) & & & & \end{array}$$

C. In many instances, edges have weights associated with them. In TSP, an edge weight represents the path distance between two nodes. The difference between directed and undirected is slightly arbitrary. For instance, the following directed graph with nodes (A), (B), and (C) with edges (AC), (CA), (CB), and (BC) is equivalent to the graph from 1ciB as long (AC) = (CA) and (CB) = (BC) in edge weights:

$$\begin{array}{ccccc} (A) & & & & (B) \\ \uparrow & & & \nearrow & \\ \downarrow & & \swarrow & & \\ (C) & & & & \end{array}$$

ii. The **constraints** define the budget limitations of the problem.

A. In some instances, the budget constraints are not a limiting factor, in, which case these problems are considered unconstrained. In some cases, these problems are described as decision problems. Decision problems are problems, which have yes-no solutions.

B. In other instances, the budget constraints are a limiting factor, in, which case, these problems are considered constrained. Changing the budget constraints may change the solution drastically. In some cases, these problems are described as discrete optimization problems, which have numerical solutions.

iii. The **ruleset** (objectives of the problem) defines the type of problem. In other words, it is the set of rules that make the problem unique in some way. Absolute uniqueness is not necessarily guaranteed, as many problems can be converted into the form of other problems. The term "unique" refers more to an instance of a problem that does not require the need for conversion to be expressed as an instance of a type of problem.

(d) The combination of the elements from 1c (graph, constraints, and ruleset) together constructs a graph problem.

2. The ability to transform a budget optimization problem into graph form is important because it provides the ability to anchor knowledge regarding the algorithms that are available to solve certain problems. These transformations do not guarantee efficient algorithms to solve any problem, but they can guarantee solvability.

   (a) Efficient algorithm: An algorithm, which can solve a problem in a reasonable amount of time. The term *efficient* refers to the ability of an algorithm to solve a problem in a polynomial number of steps. More rigorously, the term *efficient* refers to the ability of an algorithm to solve a problem in polynomial time (described in more detail below).

   (b) Time: In computational theory, time refers to the number of steps (instructions) required to find a solution to a problem. The reason time is used in terms of instructions is based on the fact that basic instructions require a constant unit of time to process input. For example, adding two integers requires a constant amount of time to process regardless of the size of those two integers (within reason – a calculator can only hold so many digits). The same is true for multiplication. Even though multiplying two integers might require more time to process than adding, it still requires a constant amount of time to process regardless of the size of the two integers (again, within reason). The structure that determines the number of steps an algorithm needs to solve a problem is known as its time complexity.

(c) Time complexity: The number of steps (time) required is dependent on the size of the problem; or, rather, the amount of time required is dependent on the **input size** of a problem. The following are examples of algorithm time complexities:

   i. A constant time complexity algorithm: The time required to solve a problem is always the same and so does not depend on the size of the input. An instance of an algorithm with a constant time complexity is finding a specific card in an ordered deck. The time to find the card does not depend on the deck size. The deck size is the input size of the problem. The reason this algorithm has a constant complexity is that the algorithm can skip directly to the location where the card is without having to scan through any of the other cards. A common notation for expressing this complexity is as follows: $\mathcal{T}(n) = c$, where n is the input size and c is the constant number of steps to solve the problem.

      A. To put this example another way: If a hotel staff member needs to deliver food to a hotel guest and the room number of the guest is known, they can travel directly to the room of the guest in order to make the delivery.

   ii. A linear time complexity algorithm: The time required to solve a problem is proportional to the size of the input. An instance of an algorithm with a linear time complexity is finding a specific card in an unordered (shuffled) deck. The reason this is not constant, but linear is that the location of a specific card is unknowable without scanning through cards one at a time. Further, the amount of time it takes to find a specific card depends on the size of the deck. The time complexity is as follows: $\mathcal{T}(n) = n$, where n is the input.

      A. To put this example another way: If a hotel staff member needs to deliver food to a hotel guest and the guest room number is **not** known, they must travel to each room until they find the guest in order to make the delivery.

   iii. A quadratic time complexity algorithm: The time required to solve a problem is proportional to the squared size of the input. An instance of an algorithm with a quadratic time complexity is filling out a square multiplication table (the number of rows equals the number of columns). If the multiplication table has 10 rows, the number of steps to fill out the table is 10 times 10, or (100 steps). If the multiplication table has 20 rows, the number of steps to fill out the table is 20 times 20 (400 steps).

The input size increased by a factor of two (from 10 to 20), but the number of steps increased by a factor of four (two squared). The time complexity is as follows: $\mathcal{T}(n) = n^2$, where n is the input.

iv. A polynomial time complexity algorithm: The time required to solve a problem is proportional to a polynomial function, which is dependent on the input size. In brief, polynomial time algorithms are a class of algorithms with time complexities that can be expressed in the following form: $\mathcal{T}(n) = n^p$, where n is the input and p is a constant value. A more accurate expression of polynomial time complexity involves more terms and is described in more detail in the **Appendix: Computational Complexity** section, but the above description is sufficient for this moment in the thesis. This class of algorithms is considered to be **efficient**.

v. An exponential time complexity algorithm: The time required to solve a problem is proportional to an exponential function, which is dependent on the input size. In brief, exponential time algorithms are a class of algorithms with time complexities that can be expressed in the following form: $\mathcal{T}(n) = p^n$, where n is the input and p is a constant value. A more accurate expression of exponential time complexity involves more terms and is described in more detail in the **Appendix: Computational Complexity** section, but the above description is sufficient for this moment in the thesis. This class of algorithms is not considered to be efficient.

(d) Order of Complexity: The predominant term of computational complexity is known as the order of complexity of an algorithm. Although all the time complexity terms for an algorithm can be significant in certain applications, it can be useful in narrowing down, which algorithms might be more efficient for a particular problem. For instance, some recursive algorithms can be implemented fairly quickly, but at large input scales, they can require more computational resources than might be available. These recursive algorithms are known to have exponential time complexities, whereas their non-recursive counterparts have polynomial time complexities – even though the counterparts take longer to implement. The point is that despite the calculation of exact time complexity being a difficult process at times, the calculation of orders of complexity can be significantly less challenging. As far as expressing orders of complexity, the most common notation is known as BIg-O notation. A set of arbitrary examples illustrates this notation better than a text description:

i. For a time complexity defined as $\mathcal{T}(n) = 1000$:

A. The corresponding order of complexity is $\mathcal{O}(1)$.

B. More compactly: $\mathcal{T}(n) \in \mathcal{O}(1)$ (read as "the time complexity is in Big-O of 1").

ii. For a time complexity defined as $\mathcal{T}(n) = 100n + 1000$:

A. The corresponding order of complexity is in $\mathcal{O}(n)$.

B. More compactly: $\mathcal{T}(n) \in \mathcal{O}(n)$ (read as "the time complexity is in Big-O of n").

iii. For a time complexity defined as $\mathcal{T}(n) = 10n^2 + 100n + 1000$:

A. The corresponding order of complexity is $\mathcal{O}\left(n^2\right)$.

B. More compactly: $\mathcal{T}(n) \in \mathcal{O}\left(n^2\right)$.

iv. For a time complexity defined as $\mathcal{T}(n) = n^3 + 10n^2 + 100n + 1000$:

A. The corresponding order of complexity is $\mathcal{O}\left(n^3\right)$.

B. More compactly: $\mathcal{T}(n) \in \mathcal{O}\left(n^3\right)$.

v. For a time complexity defined as $\mathcal{T}(n) = 2^n + n^3 + 10n^2 + 100n + 1000$:

A. The corresponding order of complexity is $\mathcal{O}(2^n)$.

B. More compactly: $\mathcal{T}(n) \in \mathcal{O}(2^n)$.

3. Universal solvers are methods, which are capable of solving any graph problem. The time complexity of these methods can be inherently inefficient as they are not specifically designed for any particular set of problems, and so they do not make use of context, which could otherwise reduce the time required to find a solution. Modifications to these methods using the context of the problem can substantially reduce the time required to find a solution. Two of the most prevalent universal solvers are the brute force method and the Monte Carlo method.

(a) The brute force method is the least sophisticated universal solver as the process of finding a solution is to evaluate every potential solution from start to finish or until it finds a valid solution. For example:

i. From the NASA archive coverage of the Apollo 11 Mission (Ben Feist, 2019)[4] the trip to the moon lasted approximately 195 hours from lift-off on the 15th of July to splashdown on the 24th of July in 1969. The time spent on the moon was approximately 22 hours (from mission time 102h:46m to 124h:22m). The goal is to find a second (any second) of time within 195 hours of coverage where the NASA craft and crew were physically on the moon.

ii. A brute force method (without any modifications) scans every second of the coverage until it arrives at the beginning of the moon landing at 102h:46m. If each second that is scanned is counted as a step, it would require approximately 369960 steps to find the moment where the "Eagle has landed" (Ben Feist, 2019)[4].

(b) The Monte Carlo method is based on the premise of taking random samples from a problem and evaluating if that sample is a solution. The Monte Carlo method becomes more effective as the number of potential solutions increases. If there is a large sample space (a large number of possible samples) and there is only one solution, the Monte Carlo method can be roughly equivalent to the brute force method.

i. The way in, which the two methods can be roughly equivalent can be observed as follows: In the case of finding an Ace of Diamonds in a shuffled standard 52 card deck: the probability of the Monte Carlo method finding the card is 1 out of 52 on the first random sample; the probability of the brute force method finding the card is 1 out of 52 on the first draw from the top of the deck. If the card is not found on the first step, on the second step: the probability of the Monte Carlo method finding the card is 1 out of 51; the probability of the brute force method finding the card is 1 out of 51. The calculation of probabilities is identical between the two methods. The actual results for each method on any instance are not guaranteed to be the same, but each method converges to the same average as the number of trials approaches infinity.

ii. In the context of the example from 3ai, a Monte Carlo method (without any modifications) samples random seconds of the ~195 hours of coverage. The samples are taken one at a time and without replacement. *Without replacement* meaning that none of the seconds can be sampled more than once (although, in this instance, it does not make a significant difference). After 39 samples (or, rather, 39 steps), the probability of finding a second of coverage where the craft and crew were on the moon is greater than 99%. To put this instance into perspective: On average, 99 times out of 100, Monte Carlo finds the solution after 39 steps, whereas brute force requires 369960.

iii. Conversely, if the goal is to find the second where the "Eagle has landed" instead, the Monte Carlo method requires 77000 steps for the probability of randomly finding that second to become greater than 99%.

4. Polynomial Time Approximation Schemes (PTAS) are methods that find approximate solutions to problems that are otherwise intractable. More precisely, some problems are intractable in the sense that the exact (or most optimal) solution cannot be found efficiently (in polynomial time), whereas an approximate solution can be found efficiently. In particular, PTAS algorithms become more effective as the number of solutions, which can satisfy the constraints of the problem. In other words, the effectiveness of PTAS is inversely proportional to the restrictiveness of problem constraints. These approximate solutions (which do not contain the most optimal solution, known as the global optimum) are, in some instances, referred to as local optima. For example:

   (a) A problem known as **k-means clustering** is a common problem in data analysis where the PTAS is reliably efficient despite the problem requiring exponential time to solve exactly. For some set of data points, the goal of the problem is to find some chosen number of clusters (to find **k** clusters) where the clusters are optimally compact.

      i. More precisely, the global optimum is where {the sum of the squared distances between each data point to its respective cluster centre} for some cluster arrangement is less than or equal to {the sum of the squared distances between each data point to its respective cluster centre} for all other cluster arrangements.

      ii. The most common PTAS for k-means clustering has a time complexity of $\mathcal{T}(n) \in \mathcal{O}\left(n^2\right)$.

5. Although the model presented in this thesis is technically not (in the current iteration) a PTAS, it is an approximation scheme, but serves a similar purpose. The model makes the problem of finding an optimal set of upgrade schedules with budget constraints tractable by reverse-engineering the highest resolution (the most accurate) approximation result that is possible from a predetermined amount of computational resources that are available for some application instance.

   (a) As an analogy: If the original painting of the Mona Lisa is the exact solution, a photographed image is an approximation. The resolution of the camera determines the accuracy of the image. The money (analogous to the computational resources) available determines the resolution of the camera that can be purchased (analogous to the accuracy of the approximation result).

6. Bringing the focus back to the thesis problem: The optimization of upgrade schedules under budget constraints has a super-exponential (greater than exponential) time complexity when the goal is to find the exact solution (or the global optimum). In fact, finding the exact solution can be so intractable that performing the optimization for only one city would require more energy than is available in the observable universe.

    (a) For example:

        i. Consider the following scenario (or parameter settings):

            A. A neighbourhood with 10 electrical transformers and 100 households. For the sake of simplicity, each transformer provides electrical power to its own set of 10 households (10 transformers, 10 households per transformer, 100 households total).

            B. The upgrade time frame (or upgrade window) is from the beginning of 2020 to the beginning of 2040.

            C. The power company has created an accurate prediction of electrical demands, which include the increase in demand due to the increase in EVs. They have also predicted their annual budgets for each year.

            D. Each transformer is properly rated (has enough capacity to cover the electrical demands) for their respective households for at least the first year, from the 1st of January 2020 to the 31st of December 2020. At the start of 2021, the electrical demands for some transformers will be beyond their rated capacity.

            E. The power company must perform upgrades to prevent service interruptions due to overloaded transformers.

        ii. The goal is to find the solution for the lowest total cost of upgrades for the 20-year period, but where the annual costs are equal to or less than their respective annual budget.

A. If the annual budgets (budget constraints) are non-restrictive, finding the exact solution is trivial: calculate all possible upgrade schedules for each individual transformer over the 20 years and select the lowest cost upgrade schedules for each transformer. However, if the budget constraints are restrictive, finding the exact solution does not necessarily allow the selection of the lowest cost upgrade schedules for each transformer since the cumulative costs of each schedule could be beyond at least one of the annual budgets. For instance, if the lowest cost upgrade schedules for each transformer call for an expensive upgrade in the first year, then the sum of these costs could be beyond the budget for the first year.

B. There are 20 years, so there are 20 possible periods of upgrades for each transformer. For each transformer, there are approximately 2^{20} possible schedules (about a million or 10^6).

C. The number of all possible schedule combinations is roughly $\left(10^6\right)^{10} = 10^{60}$ (that is a 10 with 60 zeroes). To put this into perspective: A standard combination lock has four dials with 10 digits (settings) each, which has $10^4 = 10,000$ possible combinations (the technical term is **permutations**, not **combinations**) or codes. In the case of this upgrade schedules example, the size of the problem is roughly equivalent to having a combination lock with 10 dials, each dial having a million settings, and not knowing any of the satisfiable codes.

iii. The difficulty of the problem becomes more obvious when an effort is made to answer the question: if it is not possible to upgrade all transformers in the first year, what is the best way to distribute the upgrades within the 20 year upgrade window? The model presented in this thesis is meant to contribute to answering this question.

Appendix: Model: Technical Layer

1. The purpose of underscores in {variable, function, and other object} names is to differentiate an algorithm object from the concept of an object. For example, 'grid_demands' is an object that refers to a set of data specific to an algorithm, whereas 'grid demands' is a more general concept of a grid having demands regardless of whether the context is within or without an algorithm.

Model: Technical Layer: Stage One

Stage One: Input

1. num_periods:
   (a) The number of upgrade periods in the upgrade window.
   (b) An integer.
2. grid_demands:
   (a) A time series of node demands.
   (b) A 2D array-like object.
      i. Dimension 1: Node index
      ii. Dimension 2: Time index
      iii. For each node index, for each time index, there exists a demand value.
   (c) In the context of the toy electrical grid, grid_demands would contain the projected energy usage for each transformer on the grid, such as kilowatt-hours.
3. equipment_catalogue:
   (a) A dataset of equipment available for purchase.
   (b) A 2D array-like object.
      i. Dimension 1: Equipment index
      ii. Dimension 2: Equipment attributes

(c) In the context of the toy electrical grid, every piece of equipment has a corresponding set of attributes, such as:

i. equipment_rating
ii. equipment_efficiency_curve
iii. equipment_value
iv. cost_to_install
v. cost_to_uninstall

4. equipment_inventory:
   (a) A dataset of equipment that is already purchased.
   (b) A 2D array-like object.
      i. Dimension 1: Equipment index
      ii. Dimension 2: Equipment attributes
   (c) In the context of the toy electrical grid, every piece of equipment has a corresponding set of attributes, such as:
      i. equipment_rating
      ii. equipment_efficiency_curve
      iii. equipment_value
      iv. cost_to_install
      v. cost_to_uninstall
      vi. equipment_location:
         A. The location could be a storage facility or {node_index, if it is in service}

Stage One: Structure

Structure: Attributes

1. grid_demand_bins:
   (a) A 2D array-like object

i. Dimension 1: Node index

ii. Dimension 2: Energy index

iii. For each node index, for each energy index, there exists a frequency count.

(b) In the context of the toy electrical grid, the grid_demands time series is mapped to a histogram of frequencies, grid_demand_bins. This is discussed in more detail in the method set_demand_bins.

2. schedule_binaries:

(a) A 3D array-like object

i. Dimension 1: Node index

ii. Dimension 2: Schedule index

iii. For each node index, for each schedule index, there exists a binary representation of scheduled upgrades.

(b) In the context of the toy electrical grid, each schedule in schedule_binaries contains a one-hot 1D array-like object. For each period in the schedule, there is either a one or a zero. A one indicates that an upgrade is evaluated for that period, and the following periods contain zeroes.

3. optimized_upgrade_schedule_costs:

(a) A 3D array-like object

i. Dimension 1: Node index

ii. Dimension 2: Schedule index

iii. Dimension 3: Period index

iv. For each node index, for each schedule index, for each period index, there exists an equipment cost value.

4. optimized_upgrade_schedule_equipment:

(a) A 3D array-like object

i. Dimension 1: Node index

ii. Dimension 2: Schedule index

iii. Dimension 3: Period index

iv. For each node index, for each schedule index, for each period index, there exists an equipment index.

Structure: Methods

set_demand_bins

1. Before grid_demands is transformed into histograms, the time series is split into upgrade periods using num_periods.
2. For example (i):
   (a) num_periods = 2
   (b) grid_demands = [1, 2, 3, 4, 4, 3, 2, 1]
   (c) grid_demand_splits = [[1, 2, 3, 4],[4, 3, 2, 1]]
3. For example (ii):
   (a) num_periods = 4
   (b) grid_demands = [1, 2, 3, 4, 4, 3, 2, 1]
   (c) grid_demand_splits = [[1, 2], [3, 4], [4, 3], [2, 1]]
4. Maps grid_demands to grid_demand_bins.
   (a) The purpose of this method is to maintain the scalability of the model even for long time series.
   (b) A finite histogram is a set of value counts. In the context of the toy electrical grid, each node has a corresponding histogram.
   (c) Each histogram represents the frequencies of energy bins. For an extreme example, it is possible to have a histogram with only two bins. In this instance, if the maximum energy is 100 kWh and the minimum energy is 20 kWh, the following would be the bin ranges: $\left\{\left[20, \frac{20+100}{2}\right], \left(\frac{20+100}{2}, 100\right]\right\}$. Having only two bins would reduce the precision beyond utility. However, having a thousand bins or more can accurately capture the

overall energy usage in a time series. If a time series has hourly demand data over 20 years, that is 175,200 data points per node as opposed to 1000 data points per node.

(d) This could be further optimized using a sparse tensor data structure.

5. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(n)$

   (a) $n_{\text{demands}} :=$ the number of time indexes multiplied by the number of nodes

   (b) The algorithm loops through each data point, n, and maps it to their respective bins.

   (c) The loop body contains a constant number of instructions, $c_{\text{loop\_body}}$.

   (d) The prevailing order of time units per input size is on the order of the input size, $n = n_{\text{demands}}$.

6. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(n)$

   (a) $n_{\text{demands}} :=$ The number of nodes multiplied by the number of time indexes.

   (b) $c_{\text{demand\_bins}} :=$ The number of nodes multiplied by the number of bins.

   (c) The prevailing order of space units per input is on the order of the input size, $n = n_{\text{demands}} + c_{\text{demand\_bins}}$.

set_equipment_efficiency_bin_edges

1. Maps demand_bin_edges to equipment_efficiency_bin_edges.

   (a) The purpose of this method is to create a tensor containing the efficiency percentage of every element of demand_bin_edges with every piece of equipment.

   (b) This is an example of borrowing from space to reduce time. The efficiency of a piece of equipment across a range of demand values only needs to be computed once per demand value because the efficiency does not depend on, which node it belongs to. In other words, efficiency is computed once, but accessed many times instead of being computed many times. Accessing a data structure is typically much faster than calling a function and running instructions to return a value. There is a balance; however, the number of time-series data points will far exceed the static constant of demand bins.

   (c) This is building up to the computation of the heat cost for each piece of equipment and then the final search for the equipment with the minimum overall cost for each schedule.

2. Algorithm in brief:

Algorithm 1 set_equipment_efficiency_bin_edges

```
s0 = equipment_efficiency_bin_edges
s1 = equipment_catalogue
s2 = demand_bin_edges
for equip_index in equipment:
    t0 = equip_index
    s0[t0] = s1[t0].efficiency_curve(s2)
```

3. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(1)$
   (a) $c_{\text{demand\_bins}}$:= The number of nodes multiplied by the number of bins.
   (b) $c_{\text{efficiency\_curve}}$:= The efficiency_curve function has a constant time complexity.
   (c) $c_{\text{num\_equipments}}$:= The number of pieces of equipment.
   (d) $\mathcal{T}(n) \approx c_{\text{demand\_bins}} \cdot c_{\text{efficiency\_curve}} \cdot c_{\text{num\_equipments}}$
4. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(1)$
   (a) The space complexity is in the same order as the time complexity.
   (b) The efficiency_curve does not require a new allocation of space for every loop, so the space is an additive factor instead of multiplicative factor.
   (c) $\mathcal{S}(\mathcal{T}(n)) \approx c_{\text{demand\_bins}} \cdot c_{\text{num\_equipments}} + c_{\text{efficiency\_curve}}$

set_equipment_supply_bin_edges

1. For each demand, there exists a suppy. For physical infrastructure such as electrical equipment, supply is larger than demand because of inefficiencies in the system.
2. Algorithm in brief:

Algorithm 2 set_equipment_supply_bin_edges

```
s0 = supply_bin_edges
s1 = demand_bin_egdes
s2 = equipment_efficiency_bin_edges
for equip_index in equipment:
    t0 = equip_index
    s0[equip_index] = s1 / s2[equip_index]
```

3. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(1)$

    (a) $c_{\text{num\_equipments}}$:= The number of pieces of equipment.

    (b) $c_{\text{num\_bins}}$:= The number of bin edges.

    (c) $\mathcal{T}(n) \approx c_{\text{num\_bins}} \cdot c_{\text{num\_equipments}}$

    (d) $\mathcal{T}(n) \in \mathcal{O}(1)$

4. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(1)$

    (a) Each loop does not require a new allocation of memory, but does require a data structure to hold all the results.

    (b) $\mathcal{S}(\mathcal{T}(n) \approx c_{\text{num\_bins}} + c_{\text{num\_bins}} \cdot c_{\text{num\_equipments}}$

set_equipment_heat_costs

1. For each demand frequency, there exists a supply.

2. In the context of the toy electrical grid, the calculated supply values are multiplied by the corresponding demand frequency (the number of times a demand value occurs in a dataset) to calculate the total heat loss per demand bin. This heat loss is multiplied by the cost to generate that amount of energy.

3. Algorithm in brief:

Algorithm 3 set_equipment_heat_costs

```
generation_cost := cost to generate a unit of energy.
s0 = heat_cost
s1 = demand_bins
s2 = supply_bin_edges
s3 = generation_cost
for node_index in nodes:
    t0 = node_index
    for equip_index in equipment:
        t1 = equip_index
        s0[t0, t1] = s3 * (s1[t0] * s2[t1])
```

4. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(n)$

(a) $c :=$ The number of pieces of equipment.

(b) $n_{\text{num\_nodes}} :=$ The number of nodes.

(c) $c_{\text{num\_bins}} :=$ The number of bins.

(d) $\mathcal{T}(n) \approx c_{\text{num\_bins}} \cdot c_{\text{num\_equipments}} \cdot n_{\text{num\_nodes}}$

5. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(n)$

(a) Each loop does not require a new allocation of memory, but does require a data structure to hold all the results.

(b) $\mathcal{S}(\mathcal{T}(n)) \approx c_{\text{num\_bins}} \cdot c_{\text{num\_equipments}} \cdot n_{\text{num\_nodes}} + c_{\text{num\_bins}}$

set_schedule_binaries

1. The number of schedules in the current implementation is derived using the number of periods; however, the selection of the number of schedules can be made first. This allows for the selection of a manageable number of schedules. Regardless of how the number of schedules is determined, it is ultimately dependent on the available computational resources.

2. In the context of the toy electrical grid, the number of periods needed to optimize the upgrades schedules across a window of time may be relatively small. For example, an upgrade window of 20 years, four periods (every five years) may provide sufficient resolution to achieve a reasonably optimal solution. Four periods equate to eight possible upgrade schedules. This may not be true for every application of this model.

3. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(2^n)$

(a) $C_{\text{comp\_lim}} :=$ The computational limitation.

(b) $n_{\text{num\_periods}} :=$ The number of periods.

(c) $n = n_{\text{num\_periods}}$

(d) $\mathcal{T}(n) \approx 2^{n-1} \leq C_{\text{comp\_lim}}$

4. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(2^n)$

(a) Although the computation of all the schedule_binaries can be computed one by one without requiring a new allocation of memory, the output is size unavoidably $2^{n-1} \leq C_{\text{comp\_lim}}$ because the schedule_binaries are used in the next stage.

flying_vectors

1. This algorithm is designed for parallel computation and supports scalability. The current implementation of the model uses a TensorFlow function called vectorized_map. This function is specially optimized to enable fast parallel computing accessible to users; however, it does have rather strict requirements.

   (a) This function requires that the instructions be compatible with graphical processing units (GPUs) or tensor processing units (TPUs). This does not allow for certain features found in more traditional programming styles, such as control flow operators (such as if).

   (b) At the time of the function call, a computational graph is generated, so all the instructions operate on the vectorized input object in the exact same flow. This prevents looping (for, while).

   (c) The fly_vectors method exists to bend the rules of the static computational graph and lack of control flow.

   (d) TensorFlow.vectorized_map:

      i. "This method works similar to tf.map_fn, but is optimized to run much faster, possibly with a much larger memory footprint. The speedups are obtained by vectorization (see https://arxiv.org/pdf/1903.04243.pdf)" (TensorFlow, 2022)[48].

2. In a more traditional implementation of this model, a simple loop would be used to iterate through the input data. In the current implementation, the need to iterate still exists and creates programming challenges to reconcile the incompatibility with static computational graphs.

3. This algorithm creates a stack of function calls during runtime (on-the-fly) to resolve this incompatibility. This is because the number of iterations is not necessarily known at the time of implementation, and the dynamic creation of modules at runtime through reflection is not considered a secure solution.

4. The stack of function calls has an initializer function (what would be found before a while loop), numerous body functions (what would be found inside a while loop), and a return function (after a while loop). The number of body functions is determined during the instantiation of this method.

5. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(n \cdot \log(n))$
   (a) $n_{\text{num\_schedules}} :=$ The number of schedules.
   (b) $n_{\text{num\_periods}} :=$ The number of periods.
   (c) $n_{\text{num\_periods}} \approx \log(n_{\text{num\_schedules}})$
   (d) $n = n_{\text{num\_schedules}}$
   (e) $\mathcal{T}(n) \approx n \cdot \log(n)$
6. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(n)$
   (a) The space that is required to compute each function call is proportional to, n.
   (b) This is, in part, thanks to the next method, temporal_recursion_container, where each function frame is destructed (dereferenced and marked for garbage collection) before the next function call. This means that new memory does not need to be allocated to compute the function stack.

temporal_recursion_container

1. Temporal refers to the forward direction of the recursive calls. This means that the function calls do not create control-flow branches. In some recursive functions, once a leaf is reached, the algorithm backtracks to the nearest branch point (or node) and selects the next path. Temporal recursion in this context ascends the computational path until it reaches the leaf node and returns to the frame that initialized the first recursive function call.
2. The container refers to the object that contains each nested recursive call. The container is meant to reduce the memory footprint of the recursive process.
   (a) Despite recursive programming sometimes being rather {simple, compact} solutions, programmers are often taught to avoid recursive programming because of the available resources required at runtime.
   (b) In a temporally recursive function, the frame of each call persists until the leaf node is reached. This means that all objects in each frame are kept regardless of whether or not those objects are needed in subsequent calls. This is where the container is inserted.

(c) The core instructions are wrapped in a function (core function) that is called from inside the container. Once that core function returns to the container, the core function frame is destructed. The container then calls from the next function on the stack – the next container. The core function return objects are handled in a super-frame, so all that remains in each recursive frame are a reference to the core function and a reference to the container stack.

3. The space and time complexity depends on the instructions in the core function. To communicate the complexities of this method independent of the core function: The core function is assumed to have only one instruction. The complexities have the same order of magnitude as a traditional loop. This makes certain forms of recursion, such as temporal recursion, viable programming solutions.

4. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(n)$

   (a) $n_{\text{num\_containers}}$:= The number of container functions on the stack. In the context of the toy electrical grid, the number of containers is equal to the number of upgrade periods.

   (b) $c_{\text{container}}$:= The number of container instructions.

   (c) c_{core} := The number of core function instructions.

   (d) $c_{\text{container}} = 2$

   (e) $c_{\text{core}} = 1$

   (f) $c = c_{\text{core}} + c_{\text{container}}$

   (g) $n = n_{\text{num\_containers}}$

   (h) $\mathcal{T}(n) \approx c \cdot n$

5. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(n)$

   (a) $\mathcal{S}(\mathcal{T}(n)) \approx c \cdot n$

stage_one_optimizer

1. The optimizer in this stage centres around tensor operations such as {multiplication, addition, subtraction, reduce_min, and reduce_max} to satisfy the requirements of the Tensor-

flow.vectorized_map function, but is also an efficient usage of functions that are optimized (typically in a flavour of C) for such computations.

2. The algorithm is rather elaborate, so a summary is presented instead. The following is in regards to a single upgrade schedule (upgrade schedule binary) where each upgrade slice is evaluated. An upgrade slice refers to the time between scheduled upgrades.

   (a) Every upgrade schedule is repesented in slices. For example:

      i. The schedule binary is **1000,** contains one slice. The first slice contains periods one, two, three, and four.

      ii. The schedule binary is **1010,** contains two slices. In the first slice, periods one and two. In the second slice are periods three and four.

      iii. The schedule binary is **1011**, contains three slices. In the first slice are periods one and two. In the second slice is period three. In the third slice is period four.

   (b) Every slice is associated with a 2D square tensor of equipment costs accumulated across all the slice periods.

      i. Before any slices are processed, an initial square is created that contains each possible upgrade from one piece of equipment to every other piece of equipment. The full square is a static tensor used in each period computation. The diagonal of this square excludes the costs to uninstall previous equipment and install new equipment since the diagonal represents the case that the previous equipment is the same as the new equipment – meaning that there is no upgrade. In other words, the diagonal of this square represents the cost of using the equipment.

      ii. The first axis represents every possible piece of equipment that could have been installed in a previous slice.

      iii. The second axis represents every possible piece of equipment that could be installed in the current slice being processed.

   (c) The equipment costs are the combination of costs to:

      i. uninstall current equipment,

      ii. install new equipment, and,

      iii. use the equipment (heat costs (maintenance costs could also be included)).

(d) For the initial slice, the previous equipment is the equipment that is already installed on the grid. This initial equipment filters the equipment costs square so that only the previous equipment is included in the costs. This filtered square is summed together with the diagonal costs multiplied by the remaining number of periods in the slice (the filtered square and diagonal costs are reduced to 1D tensors). The equipment with the lowest overall cost is selected as the optimal equipment for that slice. This optimal equipment is then used as the previous equipment for the next slice calculations. This process continues for all slices in the schedule.

3. The results of this algorithm are returned as follows:

    (a) optimized_upgrade_schedule_costs

    (b) optimized_upgrade_schedule_equipment

4. Time complexity: $\mathcal{T}(n) \in \mathcal{O}(n \cdot \log(n))$

    (a) $C_{\text{comp\_lim}}$:= The computational limitation.

    (b) $n_{\text{num\_schedules}}$:= The number of schedules.

    (c) $n_{\text{num\_periods}}$:= The number of periods.

    (d) $c_{\text{num\_equipments}}$:= The number of pieces of equipment.

    (e) $c = c_{\text{num\_equipments}}$

    (f) $n_{\text{num\_periods}} \approx \log(n_{\text{num\_schedules}})$

    (g) $n = n_{\text{num\_schedules}}$

    (h) $\mathcal{T}(n) \approx c^2 \cdot n \cdot \log(n) \leq C_{\text{comp\_lim}}$

5. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(\log(n))$

    (a) $\mathcal{S}(\mathcal{T}(n)) \approx c \cdot \log(n) \leq C_{\text{comp\_lim}}$

Stage One: Output

1. optimized_upgrade_schedule_costs

2. optimized_upgrade_schedule_equipment

Model: Technical Layer: Stage Two

Stage Two: Input

1. optimized_upgrade_schedule_costs
2. optimized_upgrade_schedule_equipment
3. num_groups:
   (a) The number of groupings of nodes.
   (b) An integer.
4. budget_constraints:
   (a) A 1D array-like object.
      i. Dimension 1: period_index
      ii. For each period, there exists a budget constraint.
   (b) These budget constraints pertains to the grid costs as a whole, as opposed to individual node budget constraints.

Stage Two: Structure

1. The stage two structure is significantly simpler than in stage one. Stage two consists of two significant methods: (i) creating node group approximations and (ii) using brute force to calculate the most optimal combination of the node group schedules.

Structure: Attributes

1. approximated_upgrade_schedule_costs:
   (a) A 3D array-like object

i. Dimension 1: Node group index

ii. Dimension 2: Schedule index

iii. Dimension 3: Period index

iv. For each node group index, for each schedule index, for each period index, there exists an equipment cost value.

2. dim_dict:

   (a) A dictionary of dimensions means to clarify axis arguments. Axis argument errors such as off-by-one drastically change the result of accessing a data structure. Using a dictionary is a logical way to identify, which axis is meant to be accessed.

   (b) dim_dict = {"node": 0, "schedule": 1, "period": 2}

3. min_combo_indexes:

   (a) A 1D array-like object

      i. Dimension 1: Node group index

      ii. For each node group index, there exists a schedule index.

4. min_combo_costs:

   (a) A 2D array-like object

      i. Dimension 1: Node group index

      ii. Dimension 3: Period index

      iii. For each node group index, for each period index, there exists a cost value.

Structure: Methods

set_node_groups

1. The basic premise of creating node groups revolves around a concept known as max-pooling.

2. The choice of the approximation method is rather arbitrary; however, the use of max-pooling argues that the predominant values accurately approximate clusters of nodes.

3. In the context of the toy electrical grid, each group of nodes are compared. Although the algorithm in the current implementation of the model approximates groups of nodes at the period dimension, the schedules of each node group can also be clustered.

4. The number of node clusters (groups of nodes) is determined by the input argument, num_groups. The current implementation of the method assumes that the nodes are already clustered.

   (a) The algorithms for finding clusters of {fixed size and/or fixed number of clusters} is a topic that is well established. The time complexities of these clustering problems are often in NP; however, there exist many Polynomial Time Approximation Schemes (PTAS) that find optimal clusters efficiently.

5. Algorithm in brief:

Algorithm 4 set_node_groups

```
group_node_indexes := The node indexes for each group. List of lists.
s0 = optimal_upgrade_schedule_costs
s1 = approximated_upgrade_schedule_costs
s2 = group_node_indexes
for group_index in groups:
    t0 = group_index
    for schedule_index in schedules:
        t1 = schedule_index
        for period_index in periods:
            t2 = period_index
            s1[t0, t1, t2] = max(s0[s2, t1,t2])
```

6. The space and time complexities do not take into account the multiple threads processing the tensors in parallel.

7. Time complexity: $\mathcal{T}(n_0, n_1) \in \mathcal{O}(n_0 \cdot n_1 \cdot \log(n_1))$

   (a) $c :=$ The number of pieces of equipment.

   (b) $n_{\text{num\_nodes}} :=$ The number of nodes.

   (c) $n_{\text{num\_schedules}} :=$ The number of schedules.

   (d) $n_{\text{num\_periods}} :=$ The number of periods.

   (e) $n_{\text{num\_periods}} \approx \log(n_{\text{num\_schedules}})$

(f) $n_0 = n_{\text{num\_nodes}}$

(g) $n_1 = n_{\text{num\_schedules}}$

(h) $\mathcal{T}(n) \approx n_0 \cdot n_1 \log(n_1)$

8. Space complexity: $\mathcal{S}(\mathcal{T}(n_0, n_1)) \in \mathcal{O}(n_0 \cdot n_1 \cdot \log(n_1))$

   (a) $\mathcal{S}(\mathcal{T}(n)) \approx n_0 \cdot n_1 \cdot \log(n_1)$

stage_two_optimizer

1. This brute force optimizer is exactly that – every schedule combination between node groups is evaluated.

2. The schedule combinations are constructed in a similar way to how the schedule binaries are constructed. However, instead of base two (one-hot), the base is the number of schedules, schedule_combos.

3. In essence, for each combination of group schedules, the period costs are summed and compared to the budget constraints. If one or more period sums is greater than their respective period budget (over-budget), then the next combo is evaluated. Otherwise, if all budget constraints are satisfied and the total sum is less than the previous minimum combo (min_combo), the new combo is stored as the new min_combo.

   (a) The budget constraints in this method are not the original input budget constraints. Because of the node groups, the budget constraints must be adjusted, adjusted_budget_constraints.

      i. temporary_variable = ceiling(num_nodes / num_groups)

      ii. adjusted_budget_constraints = budget_constraints *temporary_variable

4. Algorithm in brief:

Algorithm 5 stage_two_optimizer

```
group_node_indexes := The node indexes for each group. List of lists.
s0 = approximated_upgrade_schedule_costs
s1 = min_combo_indexes
s2 = min_combo_cost
s3 = period_totals
s4 = dim_dict
s5 = adjusted_budget_constraints
for combo_indexes in schedule_combos:
    t0 = combo_indexes
    t1 = s0[:, t0, :]
    s3 = sum(t1, axis=s4["period"])
    t2 = sum(s3)
    if (s3 < s5) && (t2 < s2):
        s1 = t0
        s2 = t2
```

5. The space and time complexities do not take into account the multiple threads processing the tensors in parallel.

6. Time complexity: $\mathcal{T}(n_0, n_1) \in \mathcal{O}(n_1^{n_0} \cdot \log(n_1))$

   (a) c := The number of pieces of equipment.

   (b) $n_{\text{num\_groups}}$:= The number of node groups.

   (c) $n_{\text{num\_schedules}}$:= The number of schedules.

   (d) $n_{\text{num\_periods}}$:= The number of periods.

   (e) $n_{\text{num\_periods}} \approx \log(n_{\text{num\_schedules}})$

   (f) $n_0 = n_{\text{num\_groups}}$

   (g) $n_1 = n_{\text{num\_schedules}}$

   (h) $\mathcal{T}(n) \approx n_1^{n_0} \cdot \log(n_1)$

7. Space complexity: $\mathcal{S}(\mathcal{T}(n_0, n_1)) \in \mathcal{O}(n_0 \cdot n_1 \cdot \log(n_1))$

   (a) $\mathcal{S}(\mathcal{T}(n)) \approx n_1^{n_0} \cdot \log(n_1)$

Stage Two: Output

1. min_combo_indexes

2. min_combo_costs

Model: Technical Layer: Stage Three

Stage Three: Input

1. min_combo_indexes

2. min_combo_costs

3. optimal_upgrade_schedule_costs

4. optimal_upgrade_schedule_equipment

Stage Three: Structure

Structure: Attributes

1. node_bounds:

    (a) A 2D array-like object

        i. Dimension 1: Node group index

        ii. Dimension 2: Period index

        iii. For each node group index, for each period index, there exists a budget bound.

2. optimized_upgrade_schedule_combo_costs

    (a) A 2D array-like object

        i. Dimension 1: Node index

        ii. Dimension 2: Period index

        iii. For each node index, for each period index, there exists a cost value.

3. optimized_upgrade_schedule_combo_equipment

   (a) A 2D array-like object

      i. Dimension 1: Node index

      ii. Dimension 2: Period index

      iii. For each node index, for each period index, there exists an equipment index.

Structure: Methods

set_node_bounds

1. The node_bounds are technically constraints; however, to avoid confusion with budget_constraints and adjusted_budget_constraints, the word bound is used. The node_bounds also differ as they are not dependent on other nodes; they are instead sets of bounds for each individual node index.

2. In essence, the optimized node group schedules set the limits of the individual node contained in each node group.

3. Time complexity: $\mathcal{T}(n_0, n_1) \in \mathcal{O}(n_0 \cdot n_1)$

   (a) $n_{\text{num\_nodes}} :=$ The number of nodes.

   (b) $n_{\text{num\_periods}} :=$ The number of periods.

   (c) $n_0 = n_{\text{num\_nodes}}$

   (d) $n_1 = n_{\text{num\_periods}}$

   (e) $\mathcal{T}(n_0, n_1) \approx n_0 \cdot n_1$

4. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(n)$

   (a) $n = n_{\text{num\_periods}}$

   (b) $\mathcal{S}(\mathcal{T}(n)) \approx n$

stage_three_optimizer

1. This algorithm is a brute force search through each node and each schedule. The schedules are sorted by minimum total cost in ascending order. The first schedule to satisfy its respective node bound is selected as the optimal schedule.

2. Algorithm in brief:

Algorithm 6 stage_three_optimizer

```
s0 = node_bounds
s1 = optimal_upgrade_schedule_costs
s2 = optimal_upgrade_schedule_equipment
s3 = dim_dict
s4 = optimized_upgrade_schedule_combo_costs
s5 = optimized_upgrade_schedule_combo_equipment
for node_index in nodes:
    t0 = node_indexes
    t3 = s0[t0]
    for schedule_index in schedules_sorted:
        t1 = schedule_index
        t2 = s1[t0, t1, :]
        all(leq(t2, t3, axis=s3["period"])):
            s4[t0] = t2
            s5[t0] = s2[t0, t1, :]
            break
```

3. Time complexity: $\mathcal{T}(n_0, n_1) \in \mathcal{O}(n_0 \cdot n_1)$

   (a) $n_{\text{num\_nodes}} :=$ The number of nodes.

   (b) $n_{\text{num\_schedules}} :=$ The number of schedules.

   (c) $n_{\text{num\_periods}} :=$ The number of periods.

   (d) $n_{\text{num\_periods}} \approx \log(n_{\text{num\_schedules}})$

   (e) $n_0 = n_{\text{num\_nodes}}$

   (f) $n_1 = n_{\text{num\_schedules}}$

   (g) $\mathcal{T}(n) \approx n_0 \cdot n_1 \cdot \log(n_1)$

4. Space complexity: $\mathcal{S}(\mathcal{T}(n)) \in \mathcal{O}(n)$

   (a) $n = n_{\text{num\_periods}}$

(b) $\mathcal{S}(\mathcal{T}(n)) \approx n$

Stage Three: Output

1. optimized_upgrade_schedule_combo_costs
2. optimized_upgrade_schedule_combo_equipment

Appendix: Model: Rigorous Layer

The following proof borrows notation from programming, specifically Python. The point is to move towards a more standardized mathematical notation that explicitly handles conflicts such as ambiguity of {functions, operators, and variables}. Another reason is to avoid large one-line expressions that are considered to be clever coding in computer science. Clever code is avoided for several reasons. It is not always reusable and more difficult to debug and maintain.

Notation:

Everything is an object (**obj**) including expressions

"$\mathrm{obj}_1 := \mathrm{obj}_2$" means $\mathrm{obj}_{\mathbf{1}}$ is defined by $\mathrm{obj}_{\mathbf{2}}$

$\mathrm{obj}_1 \,\vert\, \mathrm{obj}_2$	$:=$	obj_1 is {associated with or has the attribute} obj_2
$\mathrm{obj}_1 == \mathrm{obj}_2$	$:=$	obj_1 is equal in value to obj_2
$\mathrm{obj}_1 = \mathrm{obj}_2$	$:=$	obj_1 is obj_2
$\mathrm{obj}_1 = \mathrm{obj}_2$	$\vert$	modification of obj_1 identically modifies obj_2
$\mathrm{obj}_1 = \mathrm{obj}_2$	$\vert$	modification of obj_2 identically modifies obj_1
$\mathrm{obj}_1 = \mathrm{obj}_2$	$\vert$	if obj_1,obj_2 are numerical objects then $\mathrm{obj}_{\mathbf{1}} \boldsymbol{=} \mathrm{obj}_{\mathbf{2}}$ is an equation
$\begin{array}{r}\mathrm{obj}_1\\ \quad\mathrm{obj}_2\end{array}$	$:=$	evaluating obj_2 in the context of evaluating obj_1
$\mathrm{len}(\mathrm{obj})$	$:=$	number of elements

$$
\begin{aligned}
\mathrm{sym} &:= \text{a mathematical symbol} \\
i_{(\mathrm{sym})} &:= \text{reassignable int for counting}
\end{aligned}
$$

$$
\begin{aligned}
x, y &:= \text{obj} \\
xy &\neq x \cdot y \\
& \quad\ \text{multiplication is explicit}
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{V}_{(\mathrm{sym})} &:= \text{a re-assignable object} \\
\mathbb{A}_{(\mathrm{sym})} &:= \text{a re-assignable array} \\
\mathbb{T}_{(\mathrm{sym})} &:= \text{a re-assignable set}
\end{aligned}
$$

$$
\begin{aligned}
\Lambda(\mathbb{V}_1) &:= \text{function evaluated at } \mathbb{V}_1 \\
\Lambda(\mathbb{V}_1) &\mid \Lambda(\mathbb{V}_1) \text{ will } \textbf{return } \mathbb{V}_2 \\
\Lambda(\mathbb{V}_1) &\mid \mathbb{V}_2 = \Lambda(\mathbb{V}_1)
\end{aligned}
$$

Preamble:

$$N := \text{largest node index} \tag{11}$$

$$\mathbb{T}_N := \text{set of node indexes} \tag{12}$$

$$\mathbb{T}_N = [1, N] \subset \mathbb{Z} \tag{13}$$

$$n := \text{node index} \tag{14}$$

$$n \mid n \in \mathbb{T}_N \tag{15}$$

$$P := \text{largest period index} \tag{16}$$

$$\mathbb{T}_P := \text{set of period indexes} \tag{17}$$

$$\mathbb{T}_P = [1, P] \subset \mathbb{Z} \tag{18}$$

$$p := \text{period index} \tag{19}$$

$$p \mid p \in \mathbb{T}_P \tag{20}$$

$$S := \text{largest schedule index} \tag{21}$$

$$S = 2^{P-1} \tag{22}$$

$$\mathbb{T}_S := [1, S] \subset \mathbb{Z} \tag{23}$$

$$s := \text{schedule index} \tag{24}$$

$$s \mid s \in \mathbb{T}_S \tag{25}$$

$$T := \text{largest node schedule combination index} \tag{26}$$

$$T = S^N \tag{27}$$

$$\mathbb{T}_T := \text{set of node schedule combination indexes} \tag{28}$$

$$\mathbb{T}_T = [1, T] \subset \mathbb{Z} \tag{29}$$

$$t := \text{node schedule combination index} \tag{30}$$

$$t \mid t \in \mathbb{T}_T \tag{31}$$

The object $\mathbb{T}_N$ is the set of node indexes. In the context of the Toy Grid, a node index is an identification number which refers to a specific location where a transformer is installed.

The object $\mathbb{T}_P$ is the set of period indexes. In the context of the Toy Grid, a period index is an identification number which refers to a specific duration of time in the upgrade window where an upgrade can occur. Specifically, consider an upgrade window across 20 years with an upgrade period each year: the first year would be the first period (period 1); the second year would be the second period (period 2); the third year would be the third period (period 3); and further.

The object $\mathbb{T}_S$ is the set of schedule indexes. In the context of the Toy Grid, a schedule index is an identification number which refers to an arrangement of upgrades spread across all periods. Specifically, a schedule which spans 20 periods contains upgrade information (upgrade equipment or upgrade cost) for each period.

The object $\mathbb{T}_T$ is the set of node schedule combination indexes. In the context of the Toy Grid, a schedule combination index is an identification number which refers to a combination of schedules spread across all nodes. Specifically, the first combination consists of the first schedule from each node. The second combination consists of the first schedule from each node except for the last node, which contributes to its second schedule. The third combination consists of the first schedule from each node except for the last node, which contributes to its third schedule. Each subsequent combination increments the schedule indexes in a similar way that numbers are incremented.

- For instance, consider counting up from zero in base 10 number systems with four digits. The first number would be represented by 0000. The second number would be represented with 0001. This pattern continues until 0009, where the next number is represented by 0010 followed by 0011, 0012, 0013, 0014, 0015, ...,0019,0020, 0021, ...
- Each digit in the above example is analogous to a node index and the digit value analogous to a schedule index.

$$
\begin{aligned}
\mathbb{T}_C[t] &:= t^{\text{th}} \text{ combination of node schedules} && (32)\\
\mathbb{T}_C[t,n] &:= \text{the schedule of the } n^{\text{th}} \text{ node of } \mathbb{T}_C[t] && (33)\\
\mathbb{T}_C &\mid \mathbb{T}_C.\text{shape}[0] = T && (34)\\
\mathbb{T}_C &\mid \mathbb{T}_C.\text{shape}[1] = N && (35)
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{T}_C \mid \quad & \forall t_0 \in \mathbb{T}_T && (36)\\
& \quad \forall t_1 \in \mathbb{T}_T \setminus t_0\\
& \quad\quad \mathbb{T}_C[t_0, 0:N] \neq \mathbb{T}_C[t_1, 0:N]
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{T}_C \mid \quad & \forall t \in \mathbb{T}_T && (37)\\
& \quad \forall n \in \mathbb{T}_N\\
& \quad\quad \exists! s \in \mathbb{T}_S\\
& \quad\quad\quad \mathbb{T}_C[t,n] = s
\end{aligned}
$$

$$
\begin{aligned}
d[n] &:= \text{array of demands for the } n^{\text{th}} \text{ node} && (38)\\
d[n,p] &:= p^{\text{th}} \text{ demand element of } d[n] && (39)\\
d &\mid d.\text{shape}[0] = N && (40)\\
d &\mid d.\text{shape}[1] = P && (41)
\end{aligned}
$$

The object $\mathbb{T}_C$ is the two-dimensional array that contains the schedule information for each node for each combination of nodes. A schedule combination refers to a set of schedules from all nodes. The specific upgrade information (whether or not an upgrade occurs in a period of a schedule) can only be accessed by specifying a specific combination and a specific node.

The object d is the two-dimensional array that contains demand information for each period of each node. In the context of the Toy Grid, a node index and period index refers to the demand of a transformer across an entire year (one period that is a year in duration).

$$u_N[n] \quad := \quad \text{all upgrade costs for } n^{\text{th}} \text{ node} \tag{42}$$

$$u_N[n,s] \quad := \quad s^{\text{th}} \text{ upgrade schedule costs } u_0[n] \tag{43}$$

$$u_N[n,s,p] \quad := \quad p^{\text{th}} \text{ upgrade period cost of } u_0[n,s] \tag{44}$$

$$u_N \quad | \quad u.\text{shape}[0] = N \tag{45}$$

$$u_N \quad | \quad u.\text{shape}[1] = S \tag{46}$$

$$u_N \quad | \quad u.\text{shape}[2] = P \tag{47}$$

$$\begin{aligned} u_N \quad | \quad & \forall n \in \mathbb{T}_N \\ & \quad \forall p \in \mathbb{T}_P \\ & \quad\quad U : d[n,p] \rightarrow \{u_0[n,s,p]\}_{\forall s \in \mathbb{T}_S} \end{aligned} \tag{48}$$

$$\begin{aligned} u_N \quad | \quad & \forall n_0 \in \mathbb{T}_N \\ & \quad \forall n_1 \in \mathbb{T}_N \setminus n_0 \\ & \quad\quad d[n_0] > d[n_1] \Rightarrow \{u(d[n_0,s]) > u(d[n_1,s])\}_{\forall s \in \mathbb{T}_S} \\ & \quad\quad d[n_0] == d[n_1] \Rightarrow \{u(d[n_0,s]) == u(d[n_1,s])\}_{\forall s \in \mathbb{T}_S} \end{aligned} \tag{49}$$

$$\begin{aligned} u_N \quad | \quad & \forall s_0 \in \mathbb{T}_S \\ & \quad \forall s_1 \in \mathbb{T}_S \setminus s_0 \\ & \quad\quad \forall n_0 \in \mathbb{T}_N \\ & \quad\quad\quad \forall n_1 \in \mathbb{T}_N \setminus n_0 \\ & \quad\quad\quad\quad u[n_0,s_0] \neq u[n_1,s_1] \end{aligned} \tag{50}$$

The object u_N is the three-dimensional array that contains the upgrade cost information for each period of each schedule of each node. The cost for each schedule is calculated using the demands and upgrade cost functions.

- Equation 48 refers to the mapping from demand to upgrade costs for each schedule of each node.

- Equation 49 refers to the property that if the demands from one node are greater than the demands of another node, the node with the greater demand also has a greater upgrade cost. If the demands between nodes are equal, their upgrade costs are also equal.
- Equation 50 refers to the property that the upgrade cost from one schedule from one node is not necessarily equal to the upgrade cost from another schedule from another node.

$$\mathbf{node\_group} \quad := \quad \text{a set of node indexes} \tag{51}$$

$$\mathbf{node\_group} \quad | \quad \mathbf{node\_group} \subseteq \mathbb{T}_N \tag{52}$$

$$m \quad := \quad \text{array of } \mathbf{node\_group} \text{ sets index} \tag{53}$$

$$m \quad | \quad m \in \mathbb{T}_N \tag{54}$$

$$u_m[n] \quad := \quad \text{all upgrade costs for } n^{\text{th}} \ \mathbf{node\_group} \tag{55}$$

$$u_m[n, s] \quad := \quad s^{\text{th}} \text{ upgrade schedule costs } u_m[n] \tag{56}$$

$$u_m[n, s, p] \quad := \quad p^{\text{th}} \text{ upgrade period cost of } u_m[n, s] \tag{57}$$

$$u_m \quad | \quad u_m.\text{shape}[0] = m \tag{58}$$

$$u_m \quad | \quad u_m.\text{shape}[1] = S \tag{59}$$

$$u_m \quad | \quad u_m.\text{shape}[2] = P \tag{60}$$

The object **node_group** is a set of nodes that are grouped together.

The object m is the number of groups of nodes as well as the index for how nested.

The object u_m is the maximum cost for each period of each schedule of each group in the **node_group** object.

$$\mathbb{G}_1 := 1^{\text{st}} \text{ array of node\_group set} \tag{61}$$

$$\mathbb{G}_1[1] = \{n\}_{\forall n \in \mathbb{T}_N} \tag{62}$$

$$\mathbb{G}_1 \mid \mathbb{G}_1.\text{shape}[0] = 1 \tag{63}$$

$$\mathbb{G}_1 \mid \mathbb{G}_1.\text{shape}[1] = n \tag{64}$$

$$\mathbb{G}_1 \mid g_1 = 1 \tag{65}$$

$$\begin{aligned} u_1 = \; & \forall p \in \mathbb{T}_P \\ & \quad \forall s \in \mathbb{T}_S \\ & \quad\quad \forall i \in [1, g_1] \subset \mathbb{Z} \\ & \quad\quad\quad \max \\ & \quad\quad\quad\quad \forall n \in \mathbb{G}_1[i] \\ & \quad\quad\quad\quad\quad u_N[n, s, p] \end{aligned} \tag{66}$$

The object $\mathbb{G}_1$ is an array that has only one group that contains all nodes. In other words, all nodes exist in one group, which is the lowest possible approximation resolution.

The object u_1 is the maximum cost for each period of each schedule of each member in their respective group.

- Equation 66 explained in words, line-by-line:
  1. For each period index in the set of all periods:
  2. For each schedule index in the set of all schedules:
  3. For each group index in the set of all groups:
  4. Find the maximum of the following:
  5. For each node index in the group (specified by the group index):
  6. The upgrade cost associated with this period (period index) of this schedule (schedule index) of this node (node index).

$$\mathbb{G}_2 \quad := \quad 2^{\text{nd}} \text{ array of node\_group sets} \tag{67}$$

$$\mathbb{G}_2 \quad | \quad \mathbb{G}_2.\text{shape}[0] = 2 \tag{68}$$

$$\mathbb{G}_2 \quad | \quad \mathbb{G}_2.\text{shape}[1] \geq 1 \tag{69}$$

$$\mathbb{G}_2 \quad | \quad g_2 = 2 \tag{70}$$

$$\mathbb{G}_2 \quad | \quad \bigcup_{i=1}^{g_2} \mathbb{G}_2[i] == \mathbb{G}_1 \tag{71}$$

$$\mathbb{G}_2 \quad | \quad \bigcap_{i=1}^{g_2} \mathbb{G}_2[i] == \emptyset \tag{72}$$

$$\begin{aligned} \mathbb{G}_2 \quad | \quad & \forall i \in [1, g_2] \\ & \quad \mathbb{G}_2[i] \subseteq \mathbb{T}_N \end{aligned} \tag{73}$$

$$\begin{aligned} u_2 \quad = \quad & \forall s \in \mathbb{T}_S \\ & \quad \forall p \in \mathbb{T}_P \\ & \quad\quad \forall i \in [1, g_2] \subset \mathbb{Z} \\ & \quad\quad\quad \max \\ & \quad\quad\quad\quad \forall n \in \mathbb{G}_2[i] \\ & \quad\quad\quad\quad\quad u_N[n, s, p] \end{aligned} \tag{74}$$

$$\begin{aligned} u_1, u_2 \quad | \quad & \forall n_1 \in [1, g_1] \subset \mathbb{Z} \\ & \quad \forall n_2 \in [1, g_2] \subset \mathbb{Z} \\ & \quad\quad \forall s \in \mathbb{T}_S \\ & \quad\quad\quad \forall p \in \mathbb{T}_P \\ & \quad\quad\quad\quad u_1[n_1, s, p] \geq u_2[n_2, s, p] \end{aligned} \tag{75}$$

$$\vdots$$

The object $\mathbb{G}_2$ is an array that has two groups of nodes. The union of each group is equivalent to the object $\mathbb{G}_1$. The intersection of all groups is an empty set since each group does not share nodes.

The object u_2 is the maximum cost for each period of each schedule of each member in their respective group.

- Equation 74 is logically identical to Equation 66, except the groupings of nodes have changed from one group to two distinct groups.

Objects u_1 and u_2 are connected by inequalities when aligned by schedule, period, and (respective) group indexes. Equation 75 describes this connection, demonstrating the result of increasing approximation resolution (more groups means a higher resolution).

- Equation 75 explained in words, line-by-line:

  1. For each group index (n_1) associated with $\mathbb{G}_1$:
  2. For each group index (n_2) associated with $\mathbb{G}_2$:
  3. For each schedule index (s) of all schedules:
  4. For each period index (p) of all periods:
  5. At this schedule (s) and this period index (p), the upgrade cost associated with the n_1 group of $\mathbb{G}_1$ is greater than or equal to the upgrade cost associated with the n_2 group of $\mathbb{G}_2$.

$$\mathbb{G}_m \quad := m^{\text{th}} \text{ array of node\_group sets} \tag{76}$$

$$\mathbb{G}_m \quad | \quad \mathbb{G}_m.\text{shape}[0] = m \tag{77}$$

$$\mathbb{G}_m \quad | \quad \mathbb{G}_m.\text{shape}[1] \geq 1 \tag{78}$$

$$\mathbb{G}_m \quad | \quad g_m := m \tag{79}$$

$$\mathbb{G}_m \quad | \quad \bigcup_{i=1}^{g_m} \mathbb{G}_m[i] == \mathbb{G}_{m-1} \tag{80}$$

$$\mathbb{G}_m \quad | \quad \bigcap_{i=1}^{g_m} \mathbb{G}_m[i] == \emptyset \tag{81}$$

$$\begin{aligned} \mathbb{G}_m \quad | \quad & \forall i_m \in [1, m] \subset \mathbb{Z} \\ & \quad \exists! i_{m-1} \in [1, m-1] \subset \mathbb{Z} \\ & \quad\quad \mathbb{G}_m[i_m] \subseteq \mathbb{G}_{m-1}[i_{m-1}] \end{aligned} \tag{82}$$

$$\begin{aligned} u_m \quad = & \forall s \in \mathbb{T}_S \\ & \quad \forall p \in \mathbb{T}_P \\ & \quad\quad \forall i \in [1, g_m] \subset \mathbb{Z} \\ & \quad\quad\quad \max \\ & \quad\quad\quad\quad \forall n \in \mathbb{G}_m[i] \\ & \quad\quad\quad\quad\quad u_N[n, s, p] \end{aligned} \tag{83}$$

$$\begin{aligned} u_m, u_{m-1} \quad | \quad & \forall n_{m-1} \in [1, g_{m-1}] \subset \mathbb{Z} \\ & \quad \forall n_m \in [1, g_m] \subset \mathbb{Z} \\ & \quad\quad \forall s \in \mathbb{T}_S \\ & \quad\quad\quad \forall p \in \mathbb{T}_P \\ & \quad\quad\quad\quad u_{m-1}[n_{m-1}, s, p] \geq u_m[n_m, s, p] \end{aligned} \tag{84}$$

$$\vdots$$

The object $\mathbb{G}_m$ is an array that has m number of groups of nodes. The union of each group is equivalent to the object $\mathbb{G}_1$. The intersection of all groups is an empty set since each group does not share nodes.

The object u_m is the maximum cost for each period of each schedule of each member in their respective group. The object m representing the number of groups in object u_m is arbitrary in terms of its definition; however, the object m still corresponds to the level of approximation resolution. In other words, for any number of groups represented by an object m, the associated object u_m has a higher approximation resolution than that of u_{m-1}.

- Equation 83 is logically identical to Equation 74, except the groupings of nodes have changed from two distinct groups to m distinct groups.

Objects u_m and u_{m-1} are connected by inequalities when aligned by schedule, period, and (respective) group indexes. Equation 84 describes this connection, demonstrating the result of increasing approximation resolution (more groups means a higher resolution).

- Equation 84 is logically identical to Equation 75, except for the level of approximation resolution represented by objects u_m and u_{m-1}.

$$\mathbb{G}_N \quad := \quad N^{\text{th}} \text{ array of node\_group sets} \tag{85}$$

$$\mathbb{G}_N \quad | \quad \mathbb{G}_N.\text{shape}[0] == N \tag{86}$$

$$\mathbb{G}_N \quad | \quad g_N := N \tag{87}$$

$$\mathbb{G}_N \quad | \quad \forall i_N \in [1, N] \subset \mathbb{Z} : \quad \mathbb{G}_N[i_N] = i_N \tag{88}$$

The object $\mathbb{G}_N$ is an array that has N number of groups of nodes. The union of each group is equivalent to the object $\mathbb{G}_N$. The intersection of all groups is an empty set since each group does not share nodes. The object $\mathbb{G}_N$ has the highest possible resolution. In other words, since each node has its own group, it is not an approximation but is exact.

$$\boldsymbol{\beta} \quad := \quad \text{budget constraints} \tag{89}$$

$$\beta[p] \quad := \quad p^{\text{th}} \text{ period budget constraint} \tag{90}$$

$$\beta \quad | \quad \beta.\text{shape} == P \tag{91}$$

$$\$_m[t] \quad := \quad t^{\text{th}} \text{ combination of node schedules period costs} \tag{92}$$

$$\$_m[t,p] \quad := \quad p^{\text{th}} \text{ period costs of } \$_m[t] \tag{93}$$

$$\$_m \quad | \quad \$_m.\text{shape}[0] == T \tag{94}$$

The object $\boldsymbol{\beta}$ is the set of budget constraints for each period, which applies to all nodes.

The object $\$_m$ is the sum of costs per period associated with each schedule combination at the resolution level with m groups.

$$\mathit{\Theta}_m \quad := \quad \text{schedule combination with minimum total cost} \tag{95}$$

$$\mathit{\Theta}_m \quad | \quad \mathit{\Theta}_m.\text{shape}[0] == 1 \tag{96}$$

$$\mathit{\Theta}_m \quad | \quad \exists t_{\Theta} \quad \forall t \in \mathbb{T}_T \setminus t_{\Theta} \quad \sum_{\forall p \in \mathbb{T}_P} \$_m[t_{\Theta}, p] \leq \sum_{\forall p \in \mathbb{T}_P} \$_m[t, p] \tag{97}$$

$$\mathit{\Theta}_m \quad = \quad \sum_{\forall p \in \mathbb{T}_P} \$_m[t_{\Theta}, p] \tag{98}$$

$$\vartheta_m[p] \quad := \quad p^{\text{th}} \text{ period cost of } \$_m[t_{\Theta}] \tag{99}$$

$$\vartheta_m[p] \quad = \quad \$_m[t_{\Theta}, p] \tag{100}$$

$$\vartheta_m \quad | \quad \sum_{\forall p \in \mathbb{T}_P} \vartheta_m[p] = \mathit{\Theta}_m \tag{101}$$

$$\vartheta_m \quad | \quad \vartheta_m.\text{shape}[0] == P \tag{102}$$

The object $\mathit{\Theta}_m$ is the sum of period costs of the schedule combination from object $\$_m$ with the minimum total cost. The object $\mathit{\Theta}_m$ represents the total cost of the approximate solution.

The object ϑ_m is the schedule combination from object $\$_m$ with the minimum total cost. The object ϑ_m represents the period costs of the approximate solution.

Proof:

Statement:

$$
\begin{aligned}
\mathbb{V}_1 \quad := \quad & \forall \beta \in \mathbb{Q}^P \\
& \quad \forall m \in \mathbb{T}_N \\
& \quad\quad \exists t_m \in \mathbb{T}_C \\
& \quad\quad\quad \forall p \in \mathbb{T}_P \\
& \quad\quad\quad\quad \$_m[t_m, p] \leq \beta[p]
\end{aligned} \tag{103}
$$

$$
\begin{aligned}
\mathbb{V}_2 \quad := \quad & \forall m \in \mathbb{T}_N \\
& \quad \exists \Theta_m
\end{aligned} \tag{104}
$$

$$
\mathbb{V}_3 \quad := \quad \mathbb{V}_1 \wedge \mathbb{V}_2 \tag{105}
$$

$$
\begin{aligned}
\mathbb{V}_4 \quad := \quad & \forall m_0 \in \mathbb{T}_N \\
& \quad \forall m_1 \in \mathbb{T}_N \setminus [1, m_0] \\
& \quad\quad m_1 > m_0 \Rightarrow \Theta_{m_0} \leq \Theta_{m_1}
\end{aligned} \tag{106}
$$

$$
\mathbb{V}_5 \quad := \quad \mathbb{V}_3 \Rightarrow \mathbb{V}_4 \tag{107}
$$

$$
\mathbb{V}_{\alpha_0} \quad := \quad \mathbb{V}_5 \Rightarrow \lim_{m \to N} \Theta_m = \Theta_N \tag{108}
$$

The object $\mathbb{V}_1$ states that there exists at least one schedule combination at each level of resolution approximation that satisfies the budget constraints.

The object $\mathbb{V}_2$ states that there exists a schedule combination with a minimum total cost at each level of resolution approximation.

The object $\mathbb{V}_3$ is the conjunction of $\mathbb{V}_1$ and $\mathbb{V}_2$.

The object $\mathbb{V}_4$ states that for each subsequent increase in the level of resolution approximation, the associated total cost of the minimum schedule combination is less than or equal to the total cost of the minimum schedule combination from the immediately previous level of resolution approximation.

The object $\mathbb{V}_5$ states that if $\mathbb{V}_3$ is true, then $\mathbb{V}_4$ is true.

The object $\mathbb{V}_{\alpha_0}$ states that if $\mathbb{V}_5$ is true, then as the level of resolution approximation increases, the difference between the total cost of the approximate solution and the total cost of the exact solution approaches zero.

Base Argument:

$$
\begin{aligned}
\Lambda_1(t,p,i) \quad := \quad & n = \mathbb{G}_1[i,0] \\
& s = \mathbb{T}_C[t,n] \\
& a_1 = \operatorname{len}\left(\mathbb{G}_1[i,:]\right) \\
& \text{return } a_1 \cdot u_1[n,s,p]
\end{aligned} \tag{109}
$$

$$
\begin{aligned}
\$_1 \quad = \quad & \forall t \in \mathbb{T}_T \\
& \quad \forall p \in \mathbb{T}_P \\
& \quad\quad \text{sum} \\
& \quad\quad\quad \forall i \in [1] \subset \mathbb{Z} \\
& \quad\quad\quad\quad \Lambda_1(t,p,i)
\end{aligned} \tag{110}
$$

$$
\begin{aligned}
\Lambda_2(t,p,i) \quad := \quad & n = \mathbb{G}_m[i,0] \\
& s = \mathbb{T}_C[t,n] \\
& a_2 = \operatorname{len}\left(\mathbb{G}_2[i,:]\right) \\
& \text{return } a_2 \cdot u_2[n,s,p]
\end{aligned} \tag{111}
$$

$$
\begin{aligned}
\$_2 \quad = \quad & \forall t \in \mathbb{T}_T \\
& \quad \forall p \in \mathbb{T}_P \\
& \quad\quad \text{sum} \\
& \quad\quad\quad \forall i \in [1,2] \subset \mathbb{Z} \\
& \quad\quad\quad\quad \Lambda_2(t,p,i)
\end{aligned} \tag{112}
$$

The object Λ_1 function returns the combination costs for all the nodes within the specified group index per specified schedule index per specified period index at the resolution where $m = 1$.

The object $\$_1$ is the sum of costs per period for each schedule combination at the resolution where $m = 1$.

The object Λ_2 function returns the combination costs for all the nodes within the specified group index per specified schedule index per specified period index at the resolution where $m = 2$.

The object $\$_2$ is the sum of costs per period for each schedule combination at the resolution where $m = 2$.

$$\begin{aligned} \mathbb{V}_6 \quad := \quad & \forall n_1 \in [1, g_1] \subset \mathbb{Z} \\ & \quad \forall n_2 \in [1, g_2] \subset \mathbb{Z} \\ & \quad\quad \forall s \in \mathbb{T}_S \\ & \quad\quad\quad \forall p \in \mathbb{T}_P \\ & \quad\quad\quad\quad u_1[n_1, s, p] \geq u_2[n_2, s, p] \end{aligned} \tag{113}$$

$$\begin{aligned} \mathbb{V}_7 \quad := \quad & \forall i_1 \in [1] \subset \mathbb{Z} \\ & \quad \exists! i_2 \in [1, 2] \subset \mathbb{Z} \\ & \quad\quad \mathbb{G}_1[i_1] \subseteq \mathbb{G}_2[i_2] \\ & \quad\quad \operatorname{len}(\mathbb{G}_1[i_1, :]) \geq \operatorname{len}(\mathbb{G}_2[i_2, :]) \end{aligned} \tag{114}$$

$$\mathbb{V}_8 \quad := \quad \mathbb{V}_6 \wedge \mathbb{V}_7 \tag{115}$$

$$\begin{aligned} \mathbb{V}_9 \quad &:= \quad \forall t \in \mathbb{T}_T \\ &\qquad \forall p \in \mathbb{T}_P \\ &\qquad\quad \$_1[t,p] \geq \$_2[t,p] \end{aligned} \tag{116}$$

$$\mathbb{V}_{10} \quad := \quad \mathbb{V}_8 \Rightarrow \mathbb{V}_9 \tag{117}$$

$$\mathbb{V}_{\alpha_1} \quad := \quad \mathbb{V}_{10} \Rightarrow \mathit{\Theta}_1 \geq \mathit{\Theta}_2 \tag{118}$$

The above set of states is the initial instance of the proof statement.

The object $\mathbb{V}_{\alpha_1}$ states that if $\mathbb{V}_{10}$ is true, then as the level of resolution approximation increases from $m = 1$ to $m = 2$, the total cost of the approximate solution at $m = 1$ must be greater than or equal to the total cost of the approximate solution at $m = 2$.

Recursive Argument:

$$\begin{aligned}\Lambda_{m-1}(t,p,i) \quad &:= \quad n = \mathbb{G}_{m-1}[i,0] \\ & \quad\quad s = \mathbb{T}_C[t,n] \\ & \quad\quad a_{m-1} = \text{len}\left(\mathbb{G}_{m-1}[i,:]\right) \\ & \quad\quad \text{return } a_{m-1} \cdot u_{m-1}[n,s,p]\end{aligned} \tag{119}$$

$$\begin{aligned}\$_{m-1} \quad &= \quad \forall t \in \mathbb{T}_T \\ & \quad\quad\quad \forall p \in \mathbb{T}_P \\ & \quad\quad\quad\quad \text{sum} \\ & \quad\quad\quad\quad\quad \forall i \in [1, m-1] \subset \mathbb{Z} \\ & \quad\quad\quad\quad\quad\quad \Lambda_{m-1}(t,p,i)\end{aligned} \tag{120}$$

$$
\begin{aligned}
\Lambda_m(t,p,i) \quad &:= \quad n = \mathbb{G}_m[i,0] \\
&\qquad s = \mathbb{T}_C[t,n] \\
&\qquad a_m = \mathrm{len}\left(\mathbb{G}_m[i,:]\right) \\
&\qquad \text{return } a_m \cdot u_m[n,s,p]
\end{aligned} \tag{121}
$$

$$
\begin{aligned}
\$_m \quad &= \quad \forall t \in \mathbb{T}_T \\
&\qquad\quad \forall p \in \mathbb{T}_P \\
&\qquad\qquad \text{sum} \\
&\qquad\qquad\quad \forall i \in [1,m] \subset \mathbb{Z} \\
&\qquad\qquad\qquad \Lambda_m(t,p,i)
\end{aligned} \tag{122}
$$

The object Λ_{m-1} function returns the combination costs for all the nodes within the specified group index per specified schedule index per specified period index at the resolution of $m-1$.

The object $\$_{m-1}$ is the sum of costs per period for each schedule combination at the resolution of $m-1$.

The object Λ_m function returns the combination costs for all the nodes within the specified group index per specified schedule index per specified period index at the resolution of m.

The object $\$_m$ is the sum of costs per period for each schedule combination at the resolution of m.

$$
\begin{aligned}
\mathbb{V}_{11} \quad := \quad & \forall n_{m-1} \in [1, g_{m-1}] \subset \mathbb{Z} \\
& \quad \forall n_m \in [1, g_m] \subset \mathbb{Z} \\
& \quad\quad \forall s \in \mathbb{T}_S \\
& \quad\quad\quad \forall p \in \mathbb{T}_P \\
& \quad\quad\quad\quad u_{m-1}[n_{m-1}, s, p] \geq u_m[n_m, s, p]
\end{aligned}
\tag{123}
$$

$$
\begin{aligned}
\mathbb{V}_{12} \quad := \quad & \forall i_{m-1} \in [1, m-1] \subset \mathbb{Z} \\
& \quad \exists! i_m \in [1, m] \subset \mathbb{Z} \\
& \quad\quad \mathbb{G}_{m-1}[i_{m-1}] \subseteq \mathbb{G}_m[i_m] \\
& \quad\quad\quad \operatorname{len}\left(\mathbb{G}_{m-1}[i_{m-1}, :]\right) \geq \operatorname{len}\left(\mathbb{G}_m[i_m, :]\right)
\end{aligned}
\tag{124}
$$

$$
\mathbb{V}_{13} \quad := \quad \mathbb{V}_1 \wedge \mathbb{V}_2 \tag{125}
$$

$$
\begin{aligned}
\mathbb{V}_{14} \quad := \quad & \forall t \in \mathbb{T}_T \\
& \quad \forall p \in \mathbb{T}_P \\
& \quad\quad \$_{m-1}[t, p] \geq \$_m[t, p]
\end{aligned}
\tag{126}
$$

$$
\mathbb{V}_{15} \quad := \quad \mathbb{V}_{13} \Rightarrow \mathbb{V}_{14} \tag{127}
$$

$$
\mathbb{V}_{\alpha_2} \quad := \quad \mathbb{V}_{15} \Rightarrow \Theta_{m-1} \geq \Theta_m \tag{128}
$$

The above set of states is an arbitrary instance of the proof statement.

The object $\mathbb{V}_{\alpha_2}$ states that if $\mathbb{V}_{15}$ is true, then as the level of resolution approximation increases from $m-1$ to m, the total cost of the approximate solution at $m-1$ must be greater than or equal to the total cost of the approximate solution at m.

Conclusion:

$$\mathbb{V}_A \quad := \quad \mathbb{V}_{\alpha_1} \wedge \mathbb{V}_{\alpha_2} \tag{129}$$

$$\mathbb{V}_\Omega \quad := \quad \mathbb{V}_A \Rightarrow \lim_{m \to N} \Theta_m = \Theta_N \tag{130}$$

The object $\mathbb{V}_A$ is the conjunction of $\mathbb{V}_{\alpha_1}$ and $\mathbb{V}_{\alpha_2}$. The approximate solutions from $\mathbb{V}_A$ decrease in cost monotonically until the maximum resolution is reached.

The object $\mathbb{V}_\Omega$ states that if $\mathbb{V}_A$ is true, then as the level of resolution approximation increases, the difference between the total cost of the approximate solution and the total cost of the exact solution approaches zero.

Appendix: Computational Complexity

1. Orders of complexity (Big-O or Big Omicron)

   (a) Consider the following orders of time complexity:

      i. $\mathcal{T}(n) \in \mathcal{O}(1)$

      ii. $\mathcal{T}(n) \in \mathcal{O}(n)$

      iii. $\mathcal{T}(n) \in \mathcal{O}(n^2)$

      iv. $\mathcal{T}(n) \in \mathcal{O}(2^n)$

   (b) The following figure shows a sliding window of input sizes and the resulting computational steps for the above orders of time complexities:

   (c) Each order of complexities is only mathematically guaranteed for sufficiently large input sizes. For instance, although all the orders of complexity from **1-a** are from most efficient (**1-a-i**) to least efficient (**1-a-iv**), the top-left plot from Figure 50 demonstrates how the short term behaviour of algorithms can be in the reverse order of efficiency. Specifically, the long-term (large input size) behaviour of $\mathcal{O}(1)$ is the most efficient an algorithm can be as it is independent of the input size; however, the short-term (small input size) behaviour of $\mathcal{O}(1)$ can require a larger number of steps than even the largest orders of complexity such as $\mathcal{O}(2^n)$.

   (d) In Figure 50, as the input sizes increase the orders of complexity tend towards being well ordered. At the input size of 100, all the orders of complexities are in order from most efficient to least efficient, as follows:

      i. $\mathcal{O}(1)$ below $\mathcal{O}(n)$

      ii. $\mathcal{O}(n)$ below $\mathcal{O}(n^2)$

      iii. $\mathcal{O}(n^2)$ below $\mathcal{O}(2^n)$

2. Components of Complexity:

   (a) **The Problem**:

      i. Decision problems have solutions that are in a yes-or-no format (true-or-false, 0-or-1).

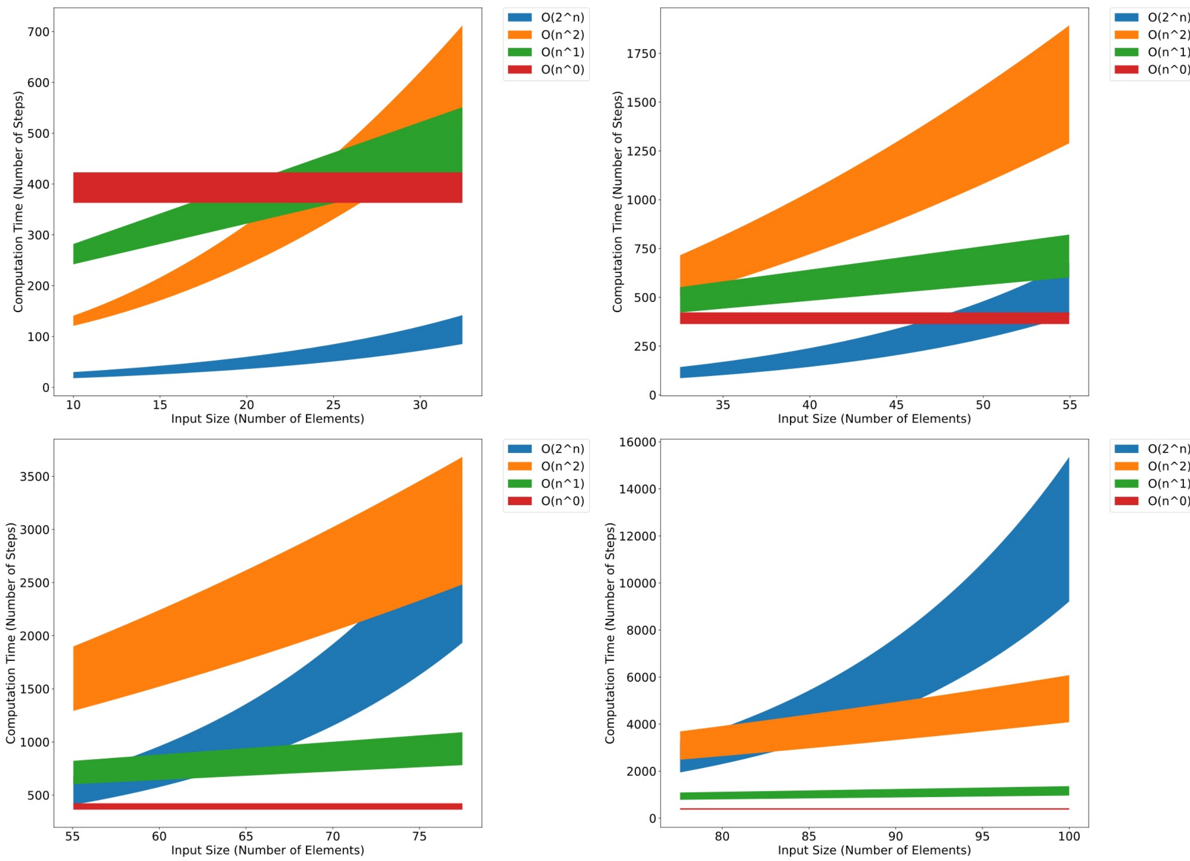

Figure 50:

The above plots are an arbitrary set of functions with various orders of complexity. The top-left plot represents a domain of smaller input sizes and the resulting number of computational steps to solve a problem. The important aspect of this plot is how the orders of complexities are not yet in order from largest to smallest(least efficient to most efficient). The top-right plot represents a domain of slightly larger input sizes and the resulting number of computational steps to solve a problem. The bottom-left plot represents a domain of a slightly larger input size domain, and the bottom-right plot has the largest input size domain. Instead of the plots of the functions being lines, areas are plotted to demonstrate the behaviour of orders of complexity with ranges of scalars.

ii. Optimization problems have solutions that are in a numerical format.

iii. Maze example: Find shorest path through a maze from entrance to exit.

(b) **The Solver**:

i. The (algorithmic) system capable of solving instances of **The Problem** with computation.

ii. Maze example: **The Solver** searches through potential paths, step-by-step, until the shortest path from entrance to exit is found.

(c) **The Oracle**:

i. The (all-knowing) system that knows the solution to **The Problem** without computation.

ii. Maze example: **The Oracle** perfectly guesses the shortest path from entrance to exit.

(d) **The Verifier**:

i. The (algorithmic) system that determines if a solution (from **The Solver** or **The Oracle**) is valid.

ii. Maze example: **The Verifier** verifies that a path from entrance to exit is the shortest possible path.

(e) **The Translator**:

i. The (algorithmic) system that converts **The Problem** into the form of another problem.

ii. Maze example: **The Translator** translates **The Problem** (of finding the shortest path through a maze) into a circuit-satisfiability problem (or any other translatable form).

3. Complexity hierarchy:

(a) **P**:

i. Decision problems with yes-no solutions.

ii. Solvers solve these problems in polynomial time.

iii. The Oracle solves these problems in polynomial time.

iv. Verifiers verify solutions to these problems in polynomial time.

v. Translator translate these problems to similar P problems in polynomial time.

(b) **NP**:

i. Decision problems with yes-no solutions.

ii. Solvers solve these problems in polynomial time or greater (such as exponential time).

iii. The Oracle solve these problems in polynomial time.

iv. Verifiers verify solutions to these problems in polynomial time.

v. Translator translate these problems to similar **NP** (as well as **P**) problems in polynomial time.

(c) **NP-Complete**:

i. Decision problems with yes-no solutions.

ii. Solver in polynomial time or greater (such as exponential time). At least as hard as the hardest NP problem.

iii. The Oracle solve these problems in polynomial time.

iv. Verifiers verify solutions to these problems in polynomial time.

v. Translator translate these problems to other **NP-Complete** (as well as **P** and **NP**) problems in polynomial time.

(d) **NP-Hard**:

i. Optimization problems with numerical solutions.

ii. Solver in polynomial time or greater (such as exponential time).

iii. The Oracle solve these problems in polynomial time.

iv. Verifiers verify solutions to these problems in polynomial time.

v. Translator translate these problems to other **NP-Hard** (as well as **P**, **NP** , and **NP-Complete**) problems in polynomial time.

Appendix: Reflections

1. The way that this thesis discussion is constructed is unconventional. Typically, discussions are focused on results; however, the decision to limit the discussion to only results is arbitrary. The self-reflective analysis of the structure of a document as a whole is arguably just as important.

2. For example, in the case of software development (software in general, from scripts to operating systems – as opposed to only industry software), the documentation of everything regarding a project is crucial in maintaining continuity. Documentation does not only include descriptions of each class, method, and attribute. Although this level of documentation might be all that is presented in the code files, documentation extends far beyond this level. Specifically, the software design, flow charts, the reasons for design decisions, communication between designers and developers, and any others, are all instances of useful information in the development and maintenance of software that occurs outside of coding documentation.

3. How the above example relates to academic and scientific work is that there lacks a discussion regarding the conventions and traditions. It is unclear why research would be harmed by this kind of reflection, which is a neutral attribute at worst. Without a justification for a lesser form of communication aside from aesthetics, the world is better off with this unspoken dogma being rejected.

References

[1] Aesop. (1484, March 26). The internet classics archive | Aesop's fables by Aesop. Retrieved from https://classics.mit.edu/Aesop/fab.3.3.html 4.4

[2] Al Jazeera. (2022, April 2). 'Who's talking climate change now?' energy producers say. Retrieved from https://www.aljazeera.com/news/2022/4/2/whos-talking-climate-change-now-energy-producers-say 1a

[3] Battle, J. (2010, September 22). The Iraq war -- Part I: The U.S. prepares for conflict, 2001. Retrieved from https://nsarchive2.gwu.edu/NSAEBB/NSAEBB326/index.htm 7(a)i, 7(a)ii, 7(a)iii, 7(a)iv

[4] Ben Feist. (2019). Apollo 11 in real time. Retrieved July 4, 2022, from https://apolloinrealtime.org/11/ 3(a)i, 3(a)ii

[5] BlackPast. (2019, September 23). (1964) Malcolm X's Speech at the Founding Rally of the Organization of Afro-American Unity. Retrieved from https://www.blackpast.org/african-american-history/speeches-african-american-history/1964-malcolm-x-s-speech-founding-rally-organization-afro-american-unity/ (document)

[6] Brown, A. (2020, June 7). Powerful petrochemical lobbying group advanced anti-protest legislation amid pandemic. Retrieved from https://theintercept.com/2020/06/07/pipeline-petrochemical-lobbying-group-anti-protest-law/ 7(e)i

[7] Brown, A., Parrish, W., & Speri, A. (2017, May 27). Leaked documents reveal counterterrorism tactics used at standing rock to "Defeat pipeline insurgencies". Retrieved from https://theintercept.com/2017/05/27/leaked-documents-reveal-security-firms-counterterrorism-tactics-at-standing-rock-to-defeat-pipeline-insurgencies/ 7(d)i

[8] Canada Energy Regulator. (2021, January 29). CER – NEB web experience survey. Retrieved from https://www.cer-rec.gc.ca/en/data-analysis/energy-markets/provincial-territorial-energy-profiles/ 1(a)ii, 1(a)iii, 1(a)iv

[9] Canadian Electricity Association. (2016). Adapting to Climate Change: State of Play and Recommendations for the Electricity Sector in Canada. Retrieved from https://www.rncan.gc.ca/sites/www.nrcan.gc.ca/files/energy/energy-resources/Adapting_to_Climate_Change_State_of_Play_and_Recommendations_for_the_Electricity_Sector_in_Canada.pdf 5

[10] Chattopadhyay, D., Bazilian, M. D., & Chattopadhyay, M. (2019, February 5). Climate change impacts on power systems. Retrieved from https://archive-yaleglobal.yale.edu/content/climate-change-impacts-power-systems 3

[11] Chomsky, N. (2005). Three factors in language design. Linguistic Inquiry, 36(1), 1-22. doi:10.1162/0024389052993655 3b

[12] Cluster analysis. (2004, May 21). Retrieved from https://en.wikipedia.org/wiki/Cluster_analysis 10

[13] Crawford, N. C., Lutz, C., & Saleh, Z. (2021, June). Costs of War: Iraqi civilians. Retrieved from https://watson.brown.edu/costsofwar/costs/human/civilians/iraqi 7(a)v

[14] David Keith, Kenton Heidel, & Robert Cherry. (2010). Capturing CO2 from the atmosphere: Rationale and Process Design Considerations. Geo-Engineering Climate Change: An Environmental Necessity or Pandora's Box?. Retrieved from https://keith.seas.harvard.edu/publications/capturing-co2-atmosphere-rationale-and-process-design-considerations 3(a)i, 3(a)ii

[15] Environment and Climate Change Canada. (2021, July 26). Greenhouse gas sources and sinks: Executive summary 2021. Retrieved from https://www.canada.ca/en/environment-climate-change/services/climate-change/greenhouse-gas-emissions/sources-sinks-executive-summary-2021.html 4

[16] FACT SHEET: Establishing the fight against corruption as a core U.S. national security interest. (2021, June 3). Retrieved from https://www.whitehouse.gov/briefing-room/statements-releases/2021/06/03/fact-sheet-establishing-the-fight-against-corruption-as-a-core-u-s-national-security-interest/ 2(e)i

[17] Frankl, V. E. (2014). Man's search for meaning. Beacon Press. 3.3.1

[18] The global expansion of authoritarian rule. (2022, February 3). Retrieved from https://freedomhouse.org/report/freedom-world/2022/global-expansion-authoritarian-rule 1(b)iA, 1(b)iB

[19] Google OR-Tools. (2020, April 29). Python reference: Algorithms: Knapsack Solver. Retrieved August 24, 2022, from https://developers.google.com/optimization/reference/python/algorithms/pywrapknapsack_solver?hl=en 2(c)ii

[20] Google OR-Tools. (2021, August 11). Integer optimization. Retrieved August 24, 2022, from https://developers.google.com/optimization/mip?hl=en 2(c)iv, 2(d)ii

[21] Google Scholar. (November, 2004).Retrieved June 10, 2022, from https://scholar.google.com

[22] Grainger Canada. (2022). Grainger-Canada: Industrial supply, safety equipment and fasteners - Grainger, Canada. Retrieved July 8, 2022, from https://www.grainger.ca/en

[23] Hopkins, Francesca (University of California, Riverside). (2018). Inland Deserts Summary Report (SUM-CCCA4-2018-008). California's Fourth Climate Change Assessment. 1b

[24] Household energy consumption, Canada and provinces. (2017, December 1). Retrieved from https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=2510006001 2a, 1b

[25] IEA. (2021). Electricity production – Electricity information: Overview – Analysis, IEA, Paris. Retrieved from https://www.iea.org/reports/electricity-information-overview/electricity-production 3d

[26] IESO. (2021, February 9). Planning and Forecasting: Annual Planning Outlook. Retrieved from https://www.ieso.ca/en/Sector-Participants/Planning-and-Forecasting/Annual-Planning-Outlook 7, 18

[27] Influence Map. (2019, March). Big oil's real agenda on climate change. Retrieved from https://influencemap.org/report/How-Big-Oil-Continues-to-Oppose-the-Paris-Agreement-38212275958aa21196dae3b76220bddc 7(c)i

[28] Jalali, R., Hills, Z., El-Khatib, K., Pazzi, R. W., & Hoornweg, D. (2016). Evaluating the Impact of Electric Vehicles on the Smart Grid. Paper presented at IARIA, The Fifth International

Conference on Advances in Vehicular Systems, Technologies and Applications. Retrieved from https://www.thinkmind.org/articles/vehicular_2016_2_30_30022.pdf 7

[29] John Tzetzes (12th century AD). (n.d.). Book of Histories (Chiliades) 2, 129-130 (Francis R. Walton, Trans.). 3.1.2

[30] Li, H., Yeo, J. H., Bornsheuer, A. L., & Overbye, T. J. (2021). The creation and validation of load time series for synthetic electric power systems. IEEE Transactions on Power Systems, 36(2), 961-969. doi:10.1109/tpwrs.2020.3018936 2

[31] Luz, S. (2022, April 4). The evidence is clear: The time for action is now. We can halve emissions by 2030. — IPCC. Retrieved from https://www.ipcc.ch/2022/04/04/ipcc-ar6-wgiii-pressrelease/ 2(a)i

[32] Ma, X., & Lv, W. (2019). Joint optimization of production and maintenance using Monte Carlo method and Metaheuristic algorithms. Mathematical Problems in Engineering, 2019, 1-22. doi:10.1155/2019/3670495 2(a)iv, 2(b)iv

[33] Metropolis, N. (1987). The beginning of the Monte Carlo method. Los Alamos Science (1987 Special Issue dedicated to Stanislaw Ulam), 128. Retrieved from https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-88-9067 2(a)ii

[34] Most, D. (2019, June 27). Are ICE detention centers concentration camps? Retrieved from https://www.bu.edu/articles/2019/are-ice-detention-centers-concentration-camps/ 1(b)iiA, 1(b)iiiA

[35] NASA Scientific Visualization Studio. (2022, January 12). SVS: Atmospheric carbon dioxide concentrations. Retrieved from https://svs.gsfc.nasa.gov/4962 3b

[36] National Intelligence Estimate: Climate change and international responses increasing challenges to U.S. national security through 2040 (NIC-NIE-2021-10030-A). (2021). Retrieved from DIRECTOR of NATIONAL INTELLIGENCE, National Intelligence Council website: https://www.dni.gov/files/ODNI/documents/assessments/NIE_Climate_Change_and_National_Security.pdf 1(b)ivA, 1(c)i, 1(c)ii, 1

[37] Natural Resources Canada. (n.d.). Fuel consumption ratings search tool: 2022 Ford Escape Hybrid. Retrieved June 7, 2022, fromhttps://fcr-ccc.nrcan-rncan.gc.ca/en#VehicleReport/25438 1(b)iiiD

[38] Natural Resources Canada. (n.d.). Fuel consumption ratings search tool: 2022 Hyundai IONIQ Electric. Retrieved June 7, 2022, fromhttps://fcr-ccc.nrcan-rncan.gc.ca/en#VehicleReport/25616 1(b)ii

[39] Natural Resources Canada. (n.d.). Fuel consumption ratings search tool: 2022 Hyundai IONIQ Blue. Retrieved June 7, 2022, from https://fcr-ccc.nrcan-rncan.gc.ca/en#VehicleReport/26218 1(b)i

[40] Oak Ridge National Laboratory. (2012, September 26). Conversion tables. Retrieved from https://cdiac.ess-dive.lbl.gov/pns/convert.html 3b

[41] Oliveira, B. B., Carravilla, M. A., & Oliveira, J. F. (2022). A diversity-based genetic algorithm for scenario generation. European Journal of Operational Research, 299(3), 1128-1141. doi:10.1016/j.ejor.2021.09.047

[42] Optimizing electric grids and charging infrastructure for mass electric vehicles penetration. (2020, September 21). Retrieved from https://www.nrcan.gc.ca/science-data/funding-partnerships/digital-accelerator/current-artificial-intelligence/optimizing-electric-grids-and-charging-infrastructure-mass-electric-vehicles-penetration/22971 2a

[43] Pörtner, H. O., Roberts, D. C., Tignor, M., Poloczanska, E. S., Mintenbeck, K., Alegría, A., . . . Rama, B. (2021). AR6 climate change 2022: Impacts, adaptation, and vulnerability. UN IPCC. 1(b)i, 1, 1a, 1b, 1a

[44] Service Canada. (2022, March 29). 2030 emissions reduction plan: Clean air, strong economy. Retrieved from https://www.canada.ca/en/services/environment/weather/climatechange/climate-plan/climate-plan-overview/emissions-reduction-2030.html 1

[45] Shakespeare, W. (1998). Julius Caesar: Third series. D. Daniell (Ed.). A&C Black. 5.2.3.1

[46] Skea, J., Calvo Buendia, E., Masson-Delmotte, V., Pörtner, H. O., Roberts, D. C., Zhai, P., . . . Shukla, P. R. (2019). IPCC special report on climate change, desertification, land degradation,

sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems: Summary for policymakers. UN IPCC. 1(b)ii

[47] Stop Ecocide International. (2020, November). Legal definition of ecocide drafted by independent expert panel. Retrieved June 10, 2022, from https://www.stopecocide.earth/legal-definition 2(c)i

[48] TensorFlow. (2022, February 3). Tf.vectorized_map. Retrieved from https://www.tensorflow.org/api_docs/python/tf/vectorized_map 1(d)i

[49] Transport Canada. (2021, June 29). Building a green economy: Government of Canada to require 100% of car and passenger truck sales be zero-emission by 2035 in Canada. Retrieved from https://www.canada.ca/en/transport-canada/news/2021/06/building-a-green-economy-government-of-canada-to-require-100-of-car-and-passenger-truck-sales-be-zero-emission-by-2035-in-canada.html 1

[50] Tzu, L. (1988). Tao Te Ching (S. Mitchell, Trans.). HarperCollins. 3.3.2.2

[51] U.S. Department of State. (2021, September 23). Senior state department official on security implications of the climate crisis in advance of Secretary Blinken's participation in the UN Security Council open debate on climate and security. Retrieved from https://www.state.gov/senior-state-department-official-on-security-implications-of-the-climate-crisis-in-advance-of-secretary-blinkens-participation-in-the-un-security-council-open-debate-on-climate-and-security/ 1

[52] U.S. Energy Information Administration. (2021, November 4). Frequently asked questions (FAQs). Retrieved from https://www.eia.gov/tools/faqs/faq.php?id=74&t=11 1(a)i

[53] United Nations. (1951). Convention on the Prevention and Punishment of the Crime of Genocide. 7(b)iiE

[54] Wampler, R. A. (2015, December 2). U.S. climate change policy in the 1980s. Retrieved from https://nsarchive2.gwu.edu/NSAEBB/NSAEBB536-Reagan-Bush-Recognized-Need-for-U.S.-Leadership-on-Climate-Change-in-1980s/ 2(b)i

[55] Wampler, R. A. (2018, September 24). The U.S. and climate change: Washington's see-saw on global leadership. Retrieved from https://nsarchive.gwu.edu/briefing-book/environmental-diplomacy/2018-09-24/us-climate-change-washingtons-see-saw-global-leadership 2(c)i, 2(d)i

[56] "Transcript for NMT 3: Justice Case." Harvard Law School Library. (2020). Nuremsberg Trials Project. Retrieved June 10, 2022, from https://nuremberg.law.harvard.edu/transcripts/3-transcript-for-nmt-3-justice-case

2(b)i, 2(b)ii