The Pennsylvania State University
The Graduate School

LONG TERM PLANNING UNDER UNCERTAINTY: MODELS AND ALGORITHMS

A Dissertation in
Industrial Engineering
by
Vijay Kumar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2021

The dissertation of Vijay Kumar was reviewed and approved by the following:

Mort D Webster
Professor of Energy Engineering
Dissertation Advisor
Chair of Committee

Uday V Shanbhag
Professor of Industrial and Manufacturing Engineering

Eunhye Song
Assistant Professor of Industrial and Manufacturing Engineering

Karen Fisher-Vanden
Professor of Environmental and Resource Economics

Steven Landry
Professor of Industrial and Manufacturing Engineering
Head of the Department of Industrial and Manufacturing Engineering

Abstract

An increase in renewable energy portfolio in the electric grid to reduce emissions from the energy sector has changed grid requirements in terms of flexibility, reliability, and resiliency. Power system planning should consider all these essential characteristics of the grid to ensure a reliable grid in the future. This thesis aims to develop tools and techniques to address power system planning with all the characteristics in a computationally tractable framework.

The methods developed are inspired by a framework called approximate dynamic programming or reinforcement learning. Approximate Dynamic Programming(ADP) is a Monte Carlo-based simulation technique that uses low order approximation of the objective function and uses dynamic programming principles to obtain future policies for planning under uncertainty. We exploit the stage-wise and problem decomposition framework present to do adaptive system planning under uncertainty with hourly resolution in simulating operations of the power system. We develop variations of the hourly power system operations, out of which one of them is integrated into long-term planning.

Chapter 2 focuses on developing hourly power system operations models, including integrality constraints to represent specific units and other operational constraints such as uptime, downtime, ramping, and reserve constraints, along with transmission constraints. To ensure computational tractability for large systems like Western Electricity Coordinating Council (WECC), we decompose into two separate models with operational constraints and transmission constraints. This approximation was required to integrate the power system model into other physical system models such as water, land, and economy in a computationally tractable framework. We compare this proposed model with different variations of the models used in the power system to estimate the impacts of water stress in the future on the power system and other parts of the economy.

Chapter 3 introduces the approximate dynamic programming framework for future power system planning. ADP is a sampling-based optimization technique, and like other techniques within these classes of methods, it suffers from the 'explore vs. exploit' problem. It needs to balance the trade-off between exploring new areas of the search space to improve estimates where the variance could be significant or exploiting the current approximation to obtain better policies. We propose a new algorithm called Q-Importance Sampling (QIS), where importance sampling is defining the sampling policy than weighing costs from some other policy. The disproportionate sampling characteristic in importance sampling addresses the explore vs. exploit problem as the approximation improves.

Chapter 4 integrates the hourly power system operations model in Chapter 2 without integrality and operational constraints into long term system planning model based on

approximate dynamic programming developed in Chapter 3. We compare the proposed model with other multi-stage stochastic optimization methods such as progressive hedging and stochastic dual dynamic programming for solution quality and computational effort.

Table of Contents

List of Figures

List of Tables

List of Symbols

GEP	Generation Expansion Planning
WECC	Western Inter-Connection
ADP	Approximate Dynamic Programming
RL	Reinforcement Learning
SDDP	Stochastic Dual Dynamic Programming
PH	Progressive Hedging
PSM	Power System Model
ED	Economic Dispatch
UC	Unit Commitment
OPF	Optimal Power Flow
UC+OPF	Unit Commitment + Optimal Power Flow
WBM	Water Balance Model
REM	Regional Economic Model
GCM	Global Circulation Models
CA	California
ROW	Rest of Western Inter-Connection
ROUS	Rest of United States
γ	Discount Factor
λ	Learning Rate

QIS	Q-Importance Sampling
QIS-RE	Q-Importance Sampling with Reduced number of evaluations.
SP	Stochastic Programming equivalent

Acknowledgments

This work would not have been complete without the assistance from multiple people, and I would like to thank everyone for their support. Firstly, I would like to thank my advisor, Mort Webster, for guiding me in the last five years as a researcher and person. Working with him on multiple projects has been a privilege, and you have shaped me from an amateur to a professional researcher. I would like to specially mention Karen Fisher-Vanden for her unwavering support for me during this journey. It has been a pleasure working with her. I want to thank her for introducing me to the world of multi-sector dynamics and PCHES. I am also grateful to my committee members, Uday V. Shanbhag and Eunhye Song, for helpful suggestions and comments about my work. The discussions have been thought-provoking and have vastly improved this work and my line of thought.

A special thank you to my fellow graduate students: Jesse, Kshitij, Brayam, Sourabh, Sushant, Aravind, and Ashish, for discussions that have improved this work and their friendships. Prawal, Puja, Sai, Disha, and Sushil: you have made this journey enjoyable, and I will miss hanging out doing potlucks or having chai. A special thank you to Neha, Karthik, Vignesh, Abhijeet, and Rajat. I missed you in State College for the last three years. Finally, I would like to give a huge thank you shoutout to all my undergraduate friends for supporting me in graduate school and the pandemic through hours of zoom sessions. Special mention to Aditya, Kanishka, Tushar, Mayur and Prashant.

To my parents Ramnath and Parvathi Kumar, who had faith in my abilities and supported me during the dissertation. I cannot thank them enough for sending goodies parcels from India, which always made me happy for the entire month. My sister, Sucheta, and brother-in-law Harish, who helped me all along since I stepped onto United States soil. They helped me settle down here and made me feel at home, and I will be eternally grateful for that. My nephew Vishnu and niece Meera for all the fun times on Whatsapp calls.

This dissertation would not be here if it were not for one person: Pratibha Mahale. Throughout this journey, she has been with me and has helped me through countless ups and downs personally, academically, and mentally. Her patience through my outbursts is admirable, and I cannot thank her enough. I am fortunate enough to have her in my life.

Computational experiments for this research were performed on the Pennsylvania State University's Computational and Data Sciences Roar supercomputer. This research is supported by the U.S. Department of Energy, Office of Science, Biological and Environmental Research Program, Earth and Environmental Systems Modeling, MultiSector Dynamics,

Contract No. DE-SC0016162. The opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Energy.

Dedication

This thesis is dedicated to all my family and friends who have supported me during graduate school.

Chapter 1
Introduction

Power systems are rapidly changing because of an increased focus on renewable generation for a cleaner environment and advancement in a range of supply and end-use technologies. The growing use of policies to combat climate change by incentivizing renewable generation has raised concerns about the flexibility of the power system to accommodate these uncertain and intermittent sources of energy.

Another trend affecting the planning and operation of power systems is the increasingly interconnected nature with other physical systems such as land and water and the vulnerability of the energy system to an extreme event because of climate change. An example of such impacts could occur under increasing surface temperatures in the Western United States, which lead to increased water temperatures in rivers and lakes, which in turn could make power plants that require water for cooling unavailable [1–4].

Historically, power systems planners would seek to achieve a system that was low cost and high reliability. More recently, this list has expanded to also include resilience against extreme events and low carbon emissions. The additional objectives motivate the search for a future electricity generation portfolio that could meet a reasonable balance in these desired objectives. The mathematical formulation of this design problem as a constrained optimization problem is commonly referred to as the generation expansion planning problem (GEP).

Power system investments, such as electric power generating plants and high voltage transmission lines, are multi-decadal decisions because of the cost and lifetime of these capital investments. Therefore, current investment decisions will have an effect on future decades and any new capital should be evaluated for the range of conditions in which that capital will be expected to operate. The longer time horizon of the impacts of these investments increases the value of considering uncertainty in those future conditions before making an

irreversible investment. Many of the relevant uncertain trends will evolve over time so that a structure where decisions are adapted based on the previous decision will enable better near-term solutions. Problems with this structure are often referred to as multi-stage adaptive stochastic optimization problems.

To date, generation expansion studies that consider impacts from water stress or other extreme events have relied on models with low temporal and spatial complexity or have not explicitly represented uncertainty. Low complexity in these models could potentially overestimate system flexibility relative to actual while failing to consider uncertainty in planning could lead to sub-optimal investments in the future. The purpose of this dissertation is twofold: First, I develop a power systems operations model with a high spatial and temporal complexity to investigate the impacts from water stress and compare the results across models that vary in the level of detail. Second, I develop improved methods for solving the adaptive generation expansion planning problem (GEP) under uncertainty that can tractably solve larger-scale problems. The rest of this chapter outlines the new characteristics of the power system, the elements of a power system model, and a summary of the principal contributions.

1.1 Power System Characteristics

Traditionally, the objectives for power systems planners have been low cost and high reliability. Low carbon emissions and resilience against extreme events were two additional objectives added by system planners to address concerns about climate change. We explain these two new characteristics below.

1.1.1 Low Carbon

The electric sector is one of the largest contributors to CO_2 emissions, accounting for about 35% of the total emissions in the United States [5]. A push for decarbonization of the electric sector has gained pace over the last decade to achieve a reduction in total emissions to meet climate objectives set by countries or states. There are three major drivers to reduce emissions from the power sector; – a) government policy initiatives, b) research in renewable sources to make them cost-effective, and c) development of new technologies or improvements in existing ones to make them cost-effective or reduce emissions.

The objective of the government policy initiatives is to incentivize electric utilities and plant owners to invest in renewable sources of energy. Some of these initiatives are achieved by introducing renewable energy portfolio standards (RES) or adding a CO_2 cost to every

technology. Research in solar and wind technologies has drastically reduced the generation cost from these resources, making them viable for adding to the grid. Some studies show the decrease in solar generation cost to be as much as 70% in the last decade and predict that it will reduce even further in the future [6].

Another technology that could assist with increasing renewables in the grid, thereby reducing emissions, is battery storage. Battery storage can 'store' electricity when the supply of energy in the grid is high and demand is low, and 'supply' energy when demand is high. Initially, this technology was cost-prohibitive, but from 2015-2018 the generation cost has decreased by 70% [7] with a further projected decrease [8], increasing the possibility of being a viable option for investment. Research in improving the efficiency of fossil-based plants, which reduces emissions, has also increased in the last decade. Some of the proposed improvements to fossil-based plants are as follows; - a) converting coal operated plants to plants that can operate coal and natural gas and b) converting single cycle gas plants to combined cycle gas plants, and c) using carbon capture and sequestration. Technology improvements in fossil-based fuel plants to reduce emissions are necessary because these plants can accommodate the variability and uncertainty of renewables and help balance the grid.

A GEP model for planning a future with low carbon is sensitive to the technologies and government policy initiatives represented in the model. Including all possible future technologies in a GEP model can give better insights into future grid investment than using a reduced subset of technologies. However, the computational complexity increases with more number of variables in the model.

1.1.2 Resilience

Resiliency in the power sector is defined as the ability of the power system to accommodate and recover from extreme events. Extreme events can be of two types. (i) The first type is high impact, low-frequency events such as earthquakes, tsunamis, hurricanes, wildfires, cyber and physical attacks. Examples of this type of extreme event are hurricanes Katrina and Sandy in the US or the Sumatra and Andaman earthquake in southern Asia. (ii) The second type is gradual events, with the potential to have an extreme impact such as global warming, rising sea levels, and variable precipitation patterns. An example of such an event is the freezing temperatures during the Texas blackouts in February 2021. Any extreme event can disrupt lives, cause damage to the power system generation and transmission system, and have serious economic consequences.

Another aspect of resiliency is the interconnected nature of the electric sector with other systems, and extreme events in one of the other systems can propagate to impact the electric grid. The energy-water-land and energy-natural gas are some of the examples of interconnected systems. One such gradual extreme event in the water system is increasing air and surface temperatures, thereby increasing water temperatures which has a cascading effect on fossil-based power plants that require water for cooling. With rising water temperatures, especially in Southern and Western United States, these plants cannot operate, thus causing an unscheduled generator outage. This outage could increase electricity prices and, in extreme cases, result in unmet demand. Power system managers could benefit from considering the impacts of events from within the power sector and other systems when deciding future investments into the grid. This would make the grid more reliable and resilient to extreme events from within the grid and from other systems. Investigating the impacts of extreme events from other systems would require building power system models with plausible detail to consider the interconnected behavior of systems.

1.2 Power System Model Elements

This section gives an overview of three elements in power system models to meet the characteristics described above, along with low cost and reliability. This list is not extensive but gives an overview of the elements that are the focus of this dissertation.

1.2.1 Uncertainty

Uncertainty is ubiquitous in our daily decision making and determining the future generation portfolio is no different. Uncertainty in power systems investment planning is broadly of two types: short-term uncertainty and long-term uncertainty. Short-term uncertainties include variable demand and renewable energy profiles based on human consumption and climate-induced factors. Uncertainty in the short term requires power system operations models at hourly temporal detail to represent the variability within hours. Long-term uncertainties include fuel prices, technological investment costs, uncertainty in policies that may be used to reduce emissions, and water availability for cooling power plants, to name a few.

Long-term and short-term uncertainty implies that no one resource plan will be the best under all possible future conditions. However, it is essential to build plans that are robust to all possible scenarios, perform well on an expected value basis, and do not have feasibility issues to meet demand. Power system operators and managers should consider the uncertainty

in future conditions to plan investments. This type of analysis will provide planners with far more confidence while making decisions such that the system will be resilient to hedge against all possible futures cost-effectively.

Including uncertainty in GEP is not straightforward and can be computationally expensive. This is because an investment and dispatch problem is solved together for every future scenario in GEP. It would not be possible to solve GEP for an exhaustive list of all scenarios. Hence, this entails the selection of uncertainties to be represented in the model, which requires technical expertise.

1.2.2 Flexibility

The power grid needs to maintain sufficient flexibility in the system depending upon the generation portfolio to maintain an uninterrupted supply of energy. Flexibility is defined as the ability to which the power system can modify electricity production in response to changes from the supply and demand side. Historically, when the objectives of the grid were low cost and reliability only, flexibility in the system from the supply side included changes in pumped storage hydro generation, coal and gas plants, or cycling between on and off for the dispatchable sources of energy such as coal and slow gas units. These units could accommodate the variability because of lower quantities of renewable energy in the grid.

However, with uptake in renewables to meet the objectives of low carbon in the grid, there is higher variability in generation. The increase in renewables has changed the requirements of the power grid. Some of the changes are a) Steeper ramps if the renewable output is decreasing and demand is increasing, b) lower high demand hours where peak demands are very short in duration, and c) ability of dispatchable generators to operate at low levels during times of high renewable energy but can rise quickly if needed.

Conventional power plants of coal and nuclear are not able to accommodate these changes while remaining profitable. Hence, managers are looking at other types of technology investments such as energy storage or quick-start gas units in the future to meet demand as we incorporate more renewable energy into the grid. Simulating power plant operations to represent flexibility in GEP would require operations models with great temporal detail and include operational constraints such as ramping, uptime, and downtime. These power plant operation models are inherently computationally intensive, and integrating them into GEP to determine future investments to represent flexibility would be even harder.

1.2.3 Complexity

Decisions in the electricity grid are made at different temporal scales. Some of the decisions are as follows; - a) balancing the reactive flow at a minute or smaller intervals, b) ten minute, thirty minute, hourly, and daily resource adequacy decisions, and c) yearly or decade-long planning decisions. The decisions are also affected by the spatial constraints present in the system because of the transmission lines in the grid. The fundamental laws of power flow govern these lines. At a high level, these laws dictate that power cannot be sent arbitrarily from one location to another, like transporting physical goods. Instead, the topology of the grid and the characteristics of the transmission lines determines power flow across each line. The other factor that affects decisions is the operational constraints of power plants. Power plants cannot arbitrarily turn on or off and change generation levels. Depending upon the power plant characteristics, the plant will have a specified ramp rate, uptime, downtime, and the reserve type it can provide to the system.

Adding each of the spatial, operational, and temporal constraints in a power system model would increase its complexity. It is necessary to evaluate whether the specified level of complexity is required for the question to address and its computational feasibility. For example, when doing reactive flow balancing, it would not be necessary to add operational constraints of generators because these constraints affect generator decisions at higher time steps. Similarly, when doing long-term planning using a GEP model, it would be computationally difficult to add minute-level reactive power balancing to calculate dispatch within GEP. Also, it is unclear whether this level of complexity would provide any enhancement to the model and the decisions. However, adding operational constraints of generators at hourly time steps to simulate dispatch in GEP changes the capacity plan compared to disregarding these constraints. Technical expertise and experience with the system are required when evaluating the tradeoffs between increasing model complexity and computational effort. Although the easy access of high-performance computing facilities has alleviated some computational concerns, some problems are still intractable, especially when considering uncertainty.

1.3 Generation Expansion Planning Literature Review

The Generation Expansion Planning model (GEP) aims to efficiently plan new generation investments to meet demand at minimum cost that answers four questions. The four questions are a) which technologies to build, b) how much of each technology to build, c) when during

the planning horizon to build, d) and where these plants are located. There are two approaches used within GEP; the first involves a centralized framework to determine investments with the objective of cost minimization, and the other is a decentralized approach considering the different market players to determine investments. This thesis focuses on the centralized approach, where a vertically integrated utility determines the new investments to minimize cost.

GEP has been an active area of research since the 1950s when linear programming was applied to the French electric power industry [9]. The complexity associated with GEP increased after the mid-1960s due to several factors. Some of them being an increase in the number of available technologies, a focus on reliable electricity supply, and an uncertain load profile. As a result, many non-linear relationships between the objective function and decision variables were observed. This motivated development of methods apart from linear programming such as nonlinear programming [10], mixed-integer programming [11], dynamic programming [12], and decomposition methods [13, 14]. Operations simplifications were made to GEP, such as ignoring of spinning reserve, or an assumption of a constant heat rate helped to keep the problems computationally tractable.

As the portfolio of available generation technologies grew and reliability concerns became more stringent for power system planners, GEP was modeled as a constrained discrete dynamic optimization problem [15, 16]. Finding satisfactory solutions to such problems requires complete enumeration of candidate expansion plans [17]. However, an exhaustive search is infeasible because potential solutions grow explosively with the problem size.

This was followed by an era of faster computers and improved solver capabilities for mixed integer programming, specifically the branch and bound method. This led to MIP models which consider a) generation and transmission decisions together [18], b) added detailed objective functions to determine investments [19] and c) used multi-objective programming with competing objectives to determine the new generation portfolio [20,21]. Another proposed method for GEP was to develop heuristics or meta-heuristics such as expert systems [16], particle swarm optimization [22] and genetic algorithms [23]. However, one of the downsides of these heuristics is that they get stuck in local optimums without any theoretical guarantee of converging to a global optimum.

All the models described above are deterministic. However, planning over the long term inevitably gives rise to uncertainties, as described in Section 1.2.1. An easy extension to include uncertainty in GEP through deterministic models has been through scenario analysis [24, 25] and probabilistic analysis [26, 27]. These methods have been derived from

deterministic models and adapted to consider uncertainties by providing a deterministic optimal solution to several distinct scenarios. The solutions are then combined in a way that is inherently subjective and is dependent on the modeler. These decisions are further evaluated with more rigorous testing before being considered a feasible candidate. Both scenario analysis and probabilistic analysis are beneficial and can give good insights, and the results can be easily communicated to an audience without technical expertise. There are two significant drawbacks of these methods: Even a good scenario selection scheme can potentially ignore some useful scenarios/uncertainties, especially within a multi-dimensional uncertainty space [14, 28]. This can result in sub-optimal decisions and, in an extreme situation, fail to accommodate the evolution of the uncertainties. Another drawback is that the methods to combine decisions from scenarios are abstract and sometimes result in inferior decision making [29].

Another approach to include uncertainty in GEP is stochastic programming. This method attempts to combine scenarios into a single, probabilistically weighted mathematical program with constraints that force a single decision at a given stage. Initially, a mixed-integer programming approach with recourse was implemented for GEP and solved using decomposition methods such as Dantzig Wolfe [13], or Benders Decomposition [14]. The short-term uncertainty of wind and solar generation in long-term planning has been investigated using two-stage stochastic programming models [30, 31]. Although two-stage stochastic studies are more common in the literature, some multi-stage stochastic studies have also been implemented [32–35]. Approximations are made to obtain decisions in multi-stage stochastic GEP such as a rolling horizon approach [34] or a partially adaptive stochastic program [33] because of the high computational effort associated with multi-stage stochastic models.

Another optimization approach applied to GEP with uncertainty is robust optimization. This approach, unlike stochastic programming, does not require probability distributions of the specified uncertainties and can be used with different objectives such as regret minimization or worst-case optimization. Initially, robust optimization was applied to single-stage capacity planning under uncertainty [36, 37]. Nonetheless, with advancements in robust optimization [38] and computing power, multi-stage robust optimization studies have become common [39–42]. Some of these studies include operational details of the power system [41] or consider generation and transmission decisions together [42,43]. Although robust optimization provides better solutions than deterministic approaches, it can be very conservative as it plans for the worst-case scenario. This has led to the development of hybrid methods which combine stochastic programming and robust optimization methods [43, 44]. These hybrid methods are

computationally less intensive than stochastic programming but give more flexible solutions than robust optimization alone.

As discussed above, there has been an evolution of models for GEP from deterministic to stochastic. However, another factor present in investment planning is the flexibility requirement of the grid. This requirement has become more pronounced with the increase in renewables and their intermittent nature as discussed in Section 1.2.2. So far, the value of flexibility in investment planning has been studied with deterministic GEP models [45–49] and a stochastic model with a small number of representative days [50]. These studies incorporate operational constraints of ramping, uptime, downtime, and reserve requirements in GEP and show that adding this complexity would significantly change the generation portfolio in the future. It is also shown that ignoring these constraints overstates the flexibility and can result in generation portfolios that cause load shedding and be infeasible to operate [47, 49]. The addition of unit commitment constraints to simulate an entire year of dispatch within multi-stage stochastic GEP is computationally intractable with the current methods in the literature. Approximations such as using representative days or weeks to simulate an entire year of dispatch are usually employed to keep the multi-stage stochastic GEP problem with unit commitment constraints tractable.

This literature review is not an exhaustive list of all the work done in Generation Expansion Planning. For a more comprehensive review, please see [51–55]. The literature cited here is to give an evolution of GEP since its initial development and to motivate the work in this dissertation.

1.4 Contributions of this Dissertation

This dissertation aims to develop models and methods to address investment planning under different uncertainties so that the grid is resilient, reliable, flexible in the future, and can withstand extreme events. Although this work focuses on GEP, I believe the methods developed here can be applied to other problem classes with a similar problem structure to GEP. The contributions are:

Chapter 2 develops large-scale production cost models for the western US with higher spatial, operational, and temporal complexity. It includes a decomposition approach to ensure computational tractability by separating detailed transmission constraints and operational constraints of generators. Approximations to ensure tractability generally include reducing spatial complexity, omitting operational constraints of generators, or representing the entire

year using time blocks. These approaches can potentially over-state the system's flexibility during daily operation and times of high stress on the system. I compare power systems models with different operations and spatial complexity to estimate the impacts on the power system under the conditions of future water stress. I also consider the effects on the economy because of the water stress and whether using power system models of different complexity under-state or over-state these impacts. I look at whether spatial or operational constraints provide maximum benefit to the model by assessing the impacts on the power grid, the economy, and the computational effort.

Chapter 3 focuses on addressing uncertainty in future power system planning using multi-stage stochastic optimization. It addresses this by using an approximate dynamic programming framework applied to GEP considering long-term uncertainties of the cost of carbon and gas prices. Approximate dynamic programming is an approach that uses a low dimensional approximation of the cost function and samples investments and uncertainties to update the approximation. I develop a novel sampling scheme within approximate dynamic programming using importance sampling to address the 'explore vs. exploit' problem called Q-Importance Sampling (QIS). This method balances exploration vs. exploitation, focusing on selecting investment decisions that provide maximum information without sampling insignificant investment decisions. I show the value of QIS by comparing it with the commonly used epsilon greedy sampling scheme in literature for solution quality and computational effort by applying it to multi-stage stochastic GEP.

Chapter 4 compares the approximate dynamic programming framework proposed in chapter 3 with two multi-stage stochastic optimization approaches present in the literature: progressive hedging and stochastic dual dynamic programming with an application to future investment planning. I compare these three approaches under various investment decisions, uncertainties, problem sizes, and decision stages. Our metrics are solution quality, computational effort, and variances of the estimates. I also provide abstract advantages and disadvantages of the three methods.

Chapter 5 summarizes the work in this dissertation and provides some directions for future work.

Chapter 2
The Value of Complexity: A comparison of power systems models in a coupled water-power-economy framework

2.1 Introduction

The electric sector is exposed to extreme events happening in nature, which has raised concerns for operators and managers about the resiliency and flexibility of these systems. Electric systems are unique because of requirements of low cost and smooth daily operations but should also be resilient to extreme events such as storms, wildfires, droughts, and extreme temperatures, to name a few. The effect of these extreme events has been extensively studied on the electric sector. One such extreme event is the unavailability of water for cooling power plants because of exceeding water temperature thresholds for thermal generation as a result of lower water levels. We refer to this extreme event as 'water stress'.

The impacts of water stress on the power system have been studied with a reduction in available capacity to the system with the potential to affect the reliability of the system [1–4,56]. All the studies have focused on the inputs to the operations of the power system, which is the current available capacity. Recently the effect of unavailable plants on the output of power system operation in terms of electricity prices and unmet demand was studied [57, 58]. This study used an hourly power system model representing the operations and transmission constraints of the system to show that the system's timing, outage capacity, and transmission network play an essential role in climate-induced outages in the power sector.

Different representations of the power system would capture the impact of operations in the electricity grid distinctly under water stress. These can be broadly defined as - a)Operations constraints, b)Temporal complexity and c)Spatial complexity. Models without operational constraints of power plants such as ramping and uptime/downtime could increase the system's flexibility than actually present. The effect of increasing operations complexity has been studied in long-term power simulation studies [45, 49, 59], where operational constraints have an impact on the future generation and transmission investments. Temporal complexity is the number of hours/days/weeks used to approximate the operations for the entire year. Simulating an entire year of operations is computationally intensive and approximations have been developed to select hours/days/weeks in long term planning models [60–62], production cost simulation software tools [63, 64] and other energy applications [65, 66]. In this work, we consider an entire year for simulating dispatch and not reducing temporal complexity, leaving this area for future work.

Spatial complexity is of two types: spatial granularity in the system and improved network representation. Spatial granularity refers to the number of buses or zones in the network. Many of the commercial tools used in power systems simulation such as PLEXOS [63], PSO [67] and REEDS [64] allow the users to specify the spatial resolution for system simulation. A comparison of the value of spatial resolution for future power system planning was investigated in [68]. They observed that increasing spatial resolution results in different investment decisions, with solar being most sensitive to aggregation. Many studies focus on a specific spatial resolution such as a zone or entire region [29,69–72], but the loss of accuracy is not discussed. The network representation refers to the modeling of the transmission flows on the lines. These can be directed flow networks [35, 73, 74], DC flow approximations [60, 64, 75] or the most sophisticated AC power flow representation [76, 77] with hybrid versions between them. AC power flow representation is usually hard to solve for large systems, although decomposition schemes have been developed to make them computational tractable [78–80]. Still, they are not used in large-scale system studies, which commonly use the DC power flow approximation that assumes bus angles are small and unit voltages [64, 81]. Some studies consider a hybrid version with some lines modeled as DC power flow approximation and others using a directed flow [59, 82]. This is primarily done to reduce the computational effort of modeling every line as a DC power flow approximation. Instead, focusing on some critical lines as DC power flow approximations and others as directed flows, the models can be solved with reasonable effort. In our system, higher spatial complexity can result in congestion of lines and inability of power transfer to regions with higher water stress compared to low

spatial complexity models, which could supply energy from a wider area disregarding the localized constraints. The addition of the spatial and operations constraints increases the model complexity and computational effort. It is necessary to understand which of these constraints would enhance the model the most and give a valid estimate of impacts under water stress in a coupled water-energy model. This is one of the contributions of this paper.

Another feedback present in the water-energy nexus is a response from the economy with demand adjustment, which has only been considered recently. Increasing prices because of more expensive power plants coming online due to water stress result in a feedback loop of reduced demand until equilibrium is achieved between the two systems. An engineering-economic model can be used to obtain valid estimates of impacts from outages because of its ability to capture two effects: The power system model is able to represent the physical and engineering constraints of the generators such as ramping, uptime/downtime, and minimum/maximum generation. Considering these constraints, the model determines the optimal set of generators to run to meet demand. The economic model, on the other hand, can recognize the ability of the end consumer to substitute across sectors with increasing prices or electricity supply disruption [83–85]. Representing both these effects will neither under-estimate the impacts considering an economic model only nor will it over-estimate the impacts considering only an engineering model. This coupling approach has been studied with power system models of low resolution [86–88], using a quadratic programming approximation [89] or recently with a power system of high spatial and temporal resolution [57]. Power system models of different complexity affect these impacts distinctly, especially within a coupled water-energy-economy approach. It would be beneficial to understand which of these constraints would provide the maximum enhancement to the coupled water-energy-economy model and give valid estimate of impacts under water stress.

We apply the multi-sectoral dynamics framework presented in [57] for the coupled water-energy-economy system with our application based on the Western Electricity Coordinating Council (WECC). This framework consists of overlapping power, water, and economy networks in a model with future climate forcing scenarios affecting water temperatures and availability. Higher water temperatures could potentially influence all these sectors, and the inter-connected nature could either mitigate or amplify the responses to each sector. We compare the benefit of increasing spatial resolution, better network representation, and increasing operations complexity of the power grid. Using the coupled framework would not only provide insights based on the power system but also from the economy on which model complexity provides maximum enhancement.

The remainder of this chapter proceeds as follows: Section 2.2 outlines the models and methods used in this analysis. Section 2.3 describes the case study and experimental design. Section 2.4 presents the analysis results and gives insights regarding which level of model complexity would provide the most value. Section 2.5 discusses the trade offs with increasing model complexity. Section 2.6 provides a concluding discussion and future work.

2.2 Models and Methodology

In this section, we give an overview of the models and methods used in this analysis. The models used in this analysis are: a daily water balance model (WBM), a state-level economy model (REM), and four power system models of different complexity. When describing the interaction of the different power system models with WBM and REM, we use the acronym PSM (Power System Model) for ease of notation. Each of the variations of power system models has an acronym as described below. The interaction between PSM, WBM, and REM is shown in Figure 2.1. Below we describe the models and the power system model-economy model coupling.

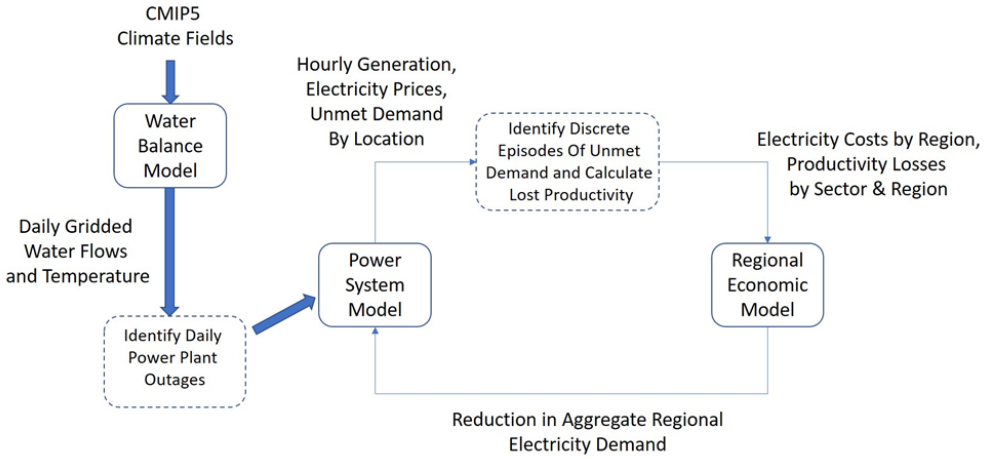


Figure 2.1: Model Components: Reproduced with permission from Webster et al

2.2.1 Power System Models

This work focuses on four power system models with different spatial and operations complexity but still simulating the entire year (8736 hours) of operations for the Western Electricity Coordinating Council (WECC). They are as follows:

- Economic Dispatch (ED): This is the simplest operations model with the objective to minimize total cost subject to meeting demand in the system and transmission limits between zones. These zones are developed by aggregating the network nodes(buses) into 13 partitions. Each partition corresponds to a state in the western united states approximately except for Washington and Oregon, considered a single zone. The total cost only includes the fuel and variable O&M costs. Transmission lines are present between zones using a transport (directed flow) type model. The constraints in this model include the minimum and maximum generator limits, and total generation should equal demand. This model does not consider the operational constraints of the generators or the higher spatial complexity to represent the transmission lines within zones as shown in figure 2.3(a).

- Unit Commitment (UC): In addition to the constraints of the ED model, this model solves to obtain the generator output minimizing fuel, variable O&M, and startup cost subject to meeting demand and operational constraints. The operational constraints of the generators represented are the minimum time the plant must be online/offline (uptime/downtime), the hourly rate at which the generator can change output (ramping), and system-wide requirements to adjust for variability in demand and renewable energy generation (reserve). This model also has binary variables to determine whether a generator is online or not and whether the plant has to be turned online. The mixed-integer programming formulation is tight and compact based on Morales-Espana et al. [90]. The spatial complexity of the model is similar to ED and is shown in figure 2.3(a).

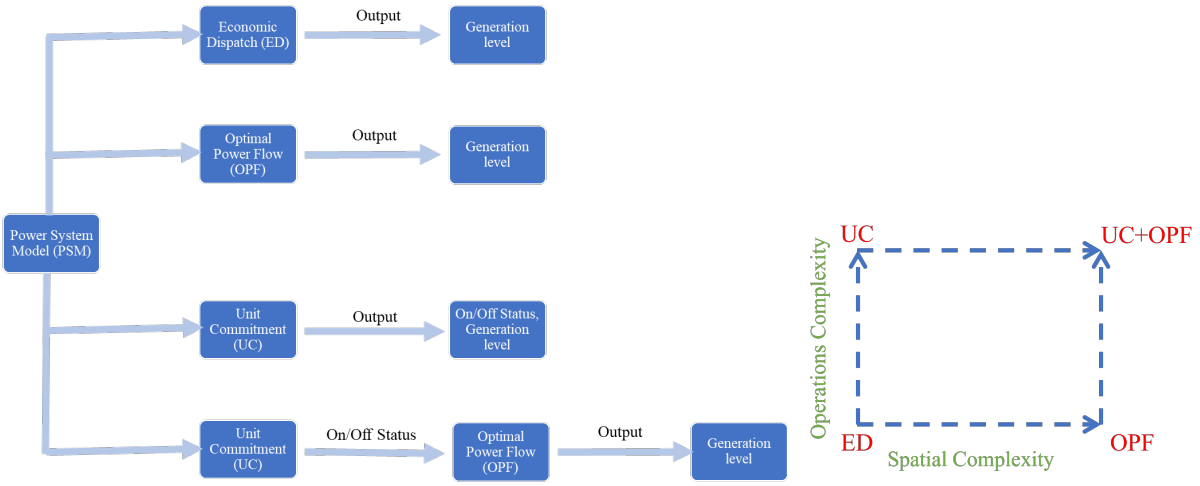
- Optimal Power Flow (OPF): This model solves for the generator output minimizing the fuel, and variable O&M cost subject to meeting demand and more detailed transmission constraints than ED or UC described above and shown in figure 2.3(b). These detailed transmission lines are between buses within and across states which have a maximum transmission capacity. The transmission flow over these lines is modeled as a linear programming approximation of the optimal power flow, called DCOPF, which assumes that the bus angles are small, and the voltages are close to the set point. This formulation of the power flow along with the ED constraints is commonly used in power simulation studies [29, 43, 70, 91]. This model does not consider the operational constraints of the generators, such as ramping and minimum uptime/downtime or system-wide reserve requirements.

- Unit Commitment + Optimal Power Flow (UC+OPF): This model considers the above two models UC and OPF, together. This model solves to obtain generator output considering the operational constraints of generators as in the UC model and the detailed transmission constraints representing high spatial resolution as in the OPF model. A single model representing both these constraints would be computationally intractable and so we make an approximation as follows:

  1. Run the Unit Commitment (UC) model to obtain the on/off status of generators subject to meeting demand, zonal transmission constraints, and operational constraints of generators.
  2. Use the on/off results from UC to fix the plants which would be online and solve the DCOPF model to obtain the generator output and electricity prices.

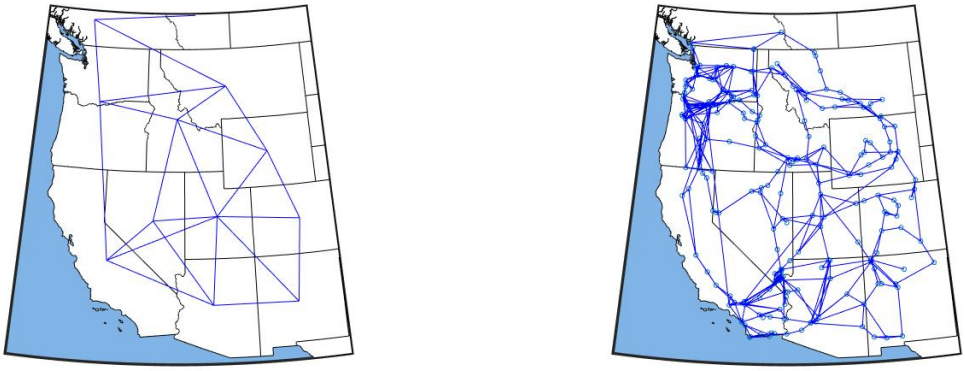
  This approximation would capture both the UC and OPF models' constraints while being computationally tractable to solve the entire year. The spatial complexity for this model for the UC model is shown in figure 2.3(a) and for the OPF model is shown in figure 2.3(b).

  An outline of the different power system models, with their outputs, is shown in Figure 2.1. Figure 2.2(b) is the visualization of the power system models depending upon the spatial and operations complexity. The model's spatial complexity increases from ED and UC to UC and UC+OPF. The operations complexity increases from ED to UC and OPF to UC+OPF.



(a) (b)

Figure 2.2: **2.1** is the different power system models used in this analysis along with their outputs. **2.2(b)** is the conceptual framework for spatial and operations complexity.



(a) (b)

Figure 2.3: **2.3(a)** is the spatial complexity for Economic Dispatch(ED) and Unit Commitment(UC) models. **2.3(b)** is the spatial complexity for Optimal Power Flow model. UC+OPF uses the spatial complexity of Unit Commitment to determine the on/off status of generators and the complexity of OPF to determine the final generation level and electricity price.

2.2.2 Water Balance Model

The water temperature scenarios in the WECC region to develop scenarios for stress in the power system were simulated using the University of New Hampshire Water Balance

Model [92, 93]. The model is a process-based spatially distributed watershed model that estimates spatially and temporally varying hydrological variables at multiple spatial scales from local and regional watersheds to continental and global systems. It contains many hydrological processes above the ground, such as river transport, human controls of water resources at dams and reservoirs, and groundwater usage. WBM also calculates the river water temperature [94] using the equilibrium temperature approach [95]. WBM was run with a grid resolution of 6x6 arc minute spatial resolution and was initialized with a spin-up period using historical data from the MEERA-2 climate drivers [96]. Downscaled climate fields from CMIP5 drive the daily water balance model for the years 2006-2099.

2.2.3 Regional Economic Model

The regional economic model is a static inter-regional computable general equilibrium model [97, 98] of the United States. This model can capture sectoral impacts and aggregate the responses to the entire sector within the economy. Four economic agents of consumers, producers, government, and trade sector are represented in this model.

The production sector comprises ten sectors (agriculture; mining; construction; manufacturing; electric power; telecommunications and utilities; trade and retail; finance, insurance, real estate; services, and public sector). Maximum profit is being considered for the producer, with both direct and indirect effects accounted for. The consumers of the nine commodities follow utility maximization, whereas the government collects taxes and then uses them to supply goods and services. The trade sector is modeled using the Armington approach [99], which assumes that domestic and imported goods are imperfect substitutes. There are three regions represented in the model: California (C.A.), the rest of WECC (ROWECC) (AZ, CO, ID, MT, NE, NM, NV, OR, UT, WA, WY), and the rest of the U.S (ROUS). This model is formulated as a complementary problem with the objective of maximizing social welfare and is solved using the PATH solver.

2.2.4 Electricity Economy Coupling

Once a water stress scenario is developed using WBM and CMIP5 climate drivers, the scenario determines the generators unavailable for cooling because of increased water temperatures. This scenario is then applied to simulate the PSM of specified complexity for the entire year to determine the marginal electricity prices and non-served energy. The impacts from higher prices and unmet demand are passed to REM which adjusts the demand

based on substitution away from the electric sector. The lower demand is then passed over to PSM, which re-calculates the dispatch, marginal electricity price, and non-served energy. This process is continued until the marginal loss of higher electricity prices is equal to the marginal loss from lower electricity demand: equilibrium is achieved because there is no benefit in changing prices or demand any further. The coupling methodology developed by Rutherford and Bohringer [89] was applied to obtain convergence. The steps of the coupling are as follows:

1. Apply water stress by simulating the CMIP5 climate model for a future year on a water balance model, resulting in unavailable generators and inputting this data to PSM.

2. Depending upon the complexity of the PSM, it can have maximum two components:

    a) Run UC/ED for the entire year (8736 hours) with 52 weeks running in parallel to obtain the on/off status of generators, electricity prices, and non-served energy. For ED and UC, the model stops here, and we proceed to step 3. UC+OPF goes to step 2b, and OPF does not consider this step.

    b) After obtaining the on/off status of generators, run OPF to determine the generator output, electricity prices, and non-served energy. OPF will only do this step and not 2a.

3. Obtain results from 52 weeks (8736 hours) and post-process it to obtain the annual average electricity price for California (C.A.) and the Rest of WECC (ROWECC). Calculate the percentage change in electricity price from the base scenario with no water stress. Details on the aggregation of the electricity prices to a single value for each region are given in the next section.

4. Obtain productivity impacts from unmet demand using the table of loss coefficients developed using [84]. Details on the productivity impacts because of unmet demand are given in the next section.

5. Using the percentage change in electricity price and productivity loss as input, run REM to obtain percentage change in demand.

6. Check termination criteria: The criteria is that demand reductions from PSM and REM do not change between consecutive iterations.

7. If termination criteria achieved, then stop. If not, go back to step 2 with a reduction in demand for CA and ROWECC obtained in step 4 to be applied to the required PSM.

2.3 Data and Experimental Design

This section gives an overview of the data and experimental design used in this analysis.

2.3.1 Data

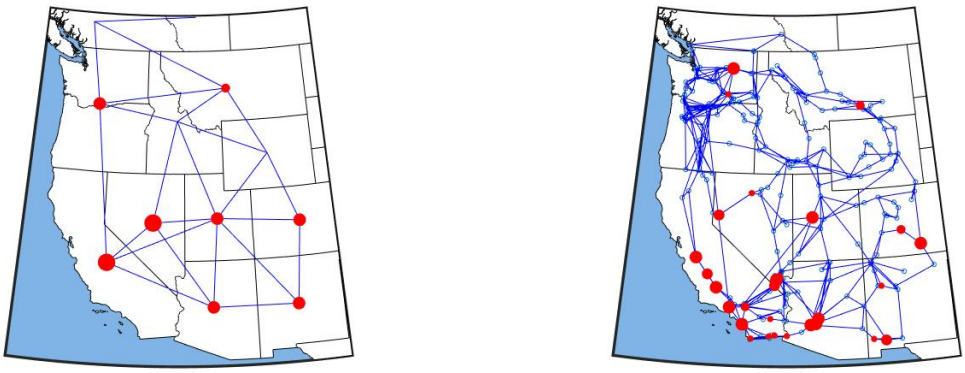
The case study considered in this paper is from the Western Electricity Coordinating Council (WECC), which covers 11 states in the United States, predominantly west of the Rocky Mountains, Alberta, British Columbia, and northern Mexico. A reduced network representation of this system is considered, which was developed by John Hopkins University [60, 100, 101] based on methods developed by Shi et al [102]. The model consists of 3569 generators, 1558 of which are conventional fuel plants and 2011 are renewable sources of energy. The zones in ED and UC are developed by aggregating the network nodes into partitions. Each partition corresponds to a state except for Oregon and Washington, which are considered a single partition. Thirty-six transmission lines are considered between zones adjacent to each other using a directed flow representation. The reduced network representation used in OPF and UCF+OPF consists of 312 buses and 654 transmission lines between the buses. This representation gives a higher spatial complexity within and across states compared to the zonal representation in UC and ED models. Demand and renewable data for the year 2010 are obtained from WECC [103, 104]. Transmission line data for the zonal and bus model are obtained from WECC. The model consists of 17 different generator types. Generator-specific data is obtained from eGrid [105] database and WECC reduced case study developed by John Hopkins.

The WBM runs were driven by output from one CMIP5 Global Circulation Model (GCM): GFDL-CM3 over 2006-2099 for RCP 8.5. The primary sources of data for the calibration of REM were supplied by the IMPLAN 2010 social accounting matrices [106]

2.3.2 Experimental Design

This section gives additional details of how the three models and output from the Global Circulation Model are used. The GCM drives the water balance model (WBM) from 2006-2099. The output of WBM was daily river temperature and flows to provide a set of yearly samples from 2006-2099. We use the daily water flows from 2041-2099 (a total of 59 scenarios) as plausible future water scenarios for the power system simulation. Each of these years provides one future scenario of water stress which is used as input to the power system models.

Within the WECC power system, we identify 487 power plants that require water for cooling. The geographic location of these plants is then matched with the grid locations in WBM and the buses/zones within the power system depending upon the spatial complexity. The output of the daily simulation from WBM is the daily water temperatures at each of the 487 generator locations for the future year considered. To determine the generator outages from these hydrological conditions, a federal standard water temperature threshold of 32 °C is considered, whereby locations with temperatures greater than this threshold would result in an outage for the plants at that location. The outages can be shown by multiple metrics and in Appendix B, we show three metrics : Number of hours with outages, average capacity outage and maximum capacity outage. The cumulative number of outages at each location for a representative week (week 29) in the GFDL-CM3 2089 scenario for UC+OPF and OPF models is shown in figure 2.4(b). The outages in UC and ED models at every geographic location in a particular zone are combined into one hypothetical location, as shown in figure 2.4(a) where all the generators in that zone are present. This can lead to a perceived increase in system flexibility compared to the bus representation in OPF and UC+OPF models because of the lack of transmission lines within and across a zone to constrain generation transmitted between and within zones.



(a) (b)

Figure 2.4: **2.4(a)** is the cummulative number of outages in week 29 for Economic Dispatch(ED) and Unit Commitment(UC) models. Outages at every location within a zone are combined into one hypothetical location for UC and ED models. **2.4(b)** is the cummulative number of outages for the OPF model. UC+OPF uses the outages of **2.4(a)** for Unit Commitment (UC) and **2.4(b)** for Optimal Power Flow. The size of the red dot indicates the magnitude of the outage. .

Power system models of different complexity then simulate each future hypothetical year

as one scenario, considering these outages for a total of 59 scenarios. The outputs of the power system models are electricity prices, electricity generation, and unmet electricity demand. Depending upon the spatial complexity of the power system, these outputs could be at a bus or state level along with the entire year (8736 hours). Because the REM is an annual model, the electricity price change needs to be aggregated and done as follows: The marginal cost of electricity is aggregated spatially to obtain an hourly cost for buses/zones in California (CA) and the Rest of WECC(ROWECC). An annual cost of electricity in each region is calculated by the weighted average of each region's hourly cost and hourly generation. When unmet demand is present in a specific bus/zone for a particular hour, it is substituted by $268 per MWh, the cost of the most expensive generator in the region for this case study. The percentage change of electricity price is then calculated relative to the solution of the benchmark scenario with no outages from higher water temperatures. The benchmark for each power system model is estimated separately with no outages.

Another output from power system models is the unmet demand, and this can have productivity losses in other sectors. This was studied by Sullivan et al. [84]. This study included a survey of approximately 12000 firms and 8000 households. They found that the largest impacts of unmet demand are for one-hour outages and on the manufacturing sector in the western united states. The authors develop a table of customer loss coefficients depending upon the region, sector, season, day of the week, and time of day. To capture these productivity impacts on the different sectors, we determine discrete episodes of unmet demand from the power system models and correspond the specific hour to the customer loss coefficients. The discrete episodes are calculated as consecutive hours of unmet demand with one hour of no unmet demand in between two episodes. This productivity impact is calculated for each power system model for 18 scenarios out of the possible 59. These 18 future scenarios were taken to account for years which induced non-zero impacts on the PSM without coupling. For all other scenarios, the change in electricity prices or unmet demand was not substantial to consider the coupling with REM. Finally, we estimate the sectoral impacts of the shock and the re-adjustment of substitution to show the effects on the economy using power system models of different complexity.

2.3.3 Implementation Notes

The power system models were implemented in GAMS with Cplex as our solver. REM is also implemented in GAMS with PATH as the solver. The pre and post-processing of data are done in Matlab. The power system models are solved for the entire year (8736 hours) but

solve the 52 weeks separately to take advantage of parallel and high-performance computing. Each week of the PSM is solved in parallel on one 2.2 GHz Intel Xeon Processor with 20 G.B. of RAM and 40 Gbps Ethernet. An optimality gap of 10^{-6} is considered for UC with a maximum time limit of two hours for each model. A penalty of \$1000/MWh is applied for non-served energy and shed energy to ensure feasibility in the power system models.

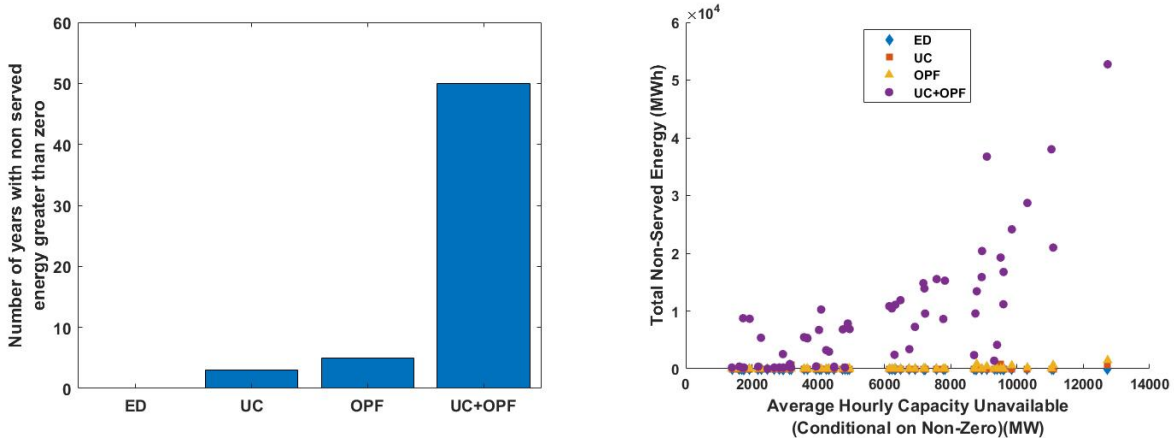
The maximum number of iterations is 15 for the electricity-economy coupling process. A gradient search algorithm was employed for the electricity-economic coupling because of the nonlinear nature of PSM and the economic model. For more details about the convergence, we request the reader to look at Webster et al. [57].

2.4 Results

The results mentioned are for all 59 scenarios unless otherwise mentioned. When we consider the economic impacts, we look at only 18 scenarios as mentioned above.

2.4.1 Unmet demand differences

Power system models with low spatial and operations complexity can accommodate greater flexibility to meet demand and, as a result, have lower non-served energy compared to systems with higher complexity. This is observed in Figure 2.5(a) which shows the number of years with non-served energy greater than zero for ED, UC, OPF, and UC+OPF. We observe that with a representation of both spatial and operations complexity in UC+OPF, a higher proportion of the future years with water stress have non-served energy compared to just representing operations (UC) and transmission (OPF). Without representing operations constraints and lower spatial resolution, no year has non-served energy, as seen in ED. Along with the frequency of years with non-served energy greater than zero, UC+OPF has a higher magnitude of non-served energy compared to the other three models, as shown in Figure 2.5(b). There is no significant difference in unmet demand (magnitude and count) between the UC and OPF model, suggesting that representing either of these constraints has similar impacts.



(a) (b)

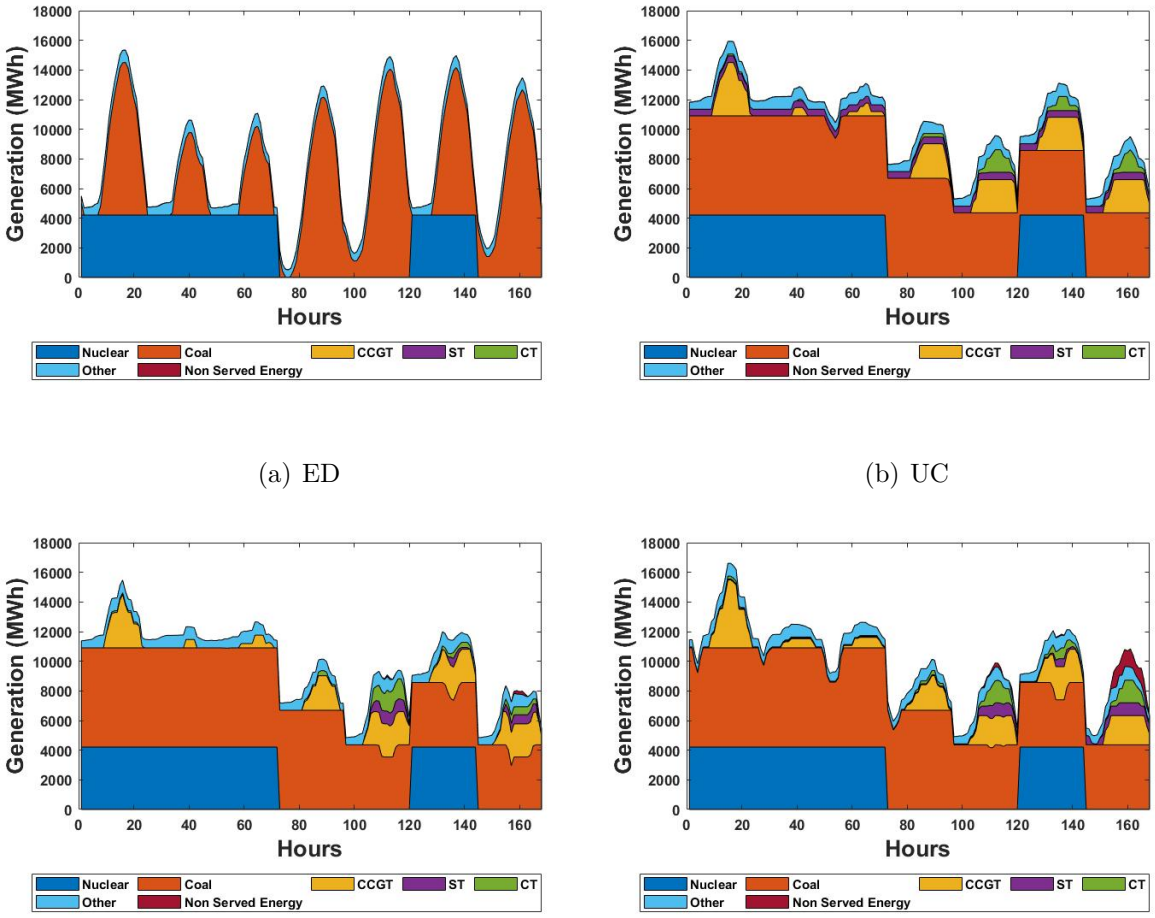
Figure 2.5: **2.5(a)** shows the number of years with non-served energy greater than zero for Economic Dispatch (ED), Unit Commitment (UC), Optimal Power Flow (OPF) and Unit Commitment+Optimal Power Flow (UC+OPF) for GFDL climate scenarios from 2041-2099. **2.5(b)** shows a scatter Plot with the average hourly unavailable capacity because of water shortages versus total non-served energy for GFDL climate scenarios from 2041-2099 for Economic Dispatch (ED), Unit Commitment (UC), Optimal Power Flow (OPF) and Unit Commitment+Optimal Power Flow (UC+OPF). The average hourly unavailable capacity in the X axis is the hourly capacity not available over all hours that it is greater than zero.

2.4.2 Power system Impacts

2.4.2.1 Generation Differences

Power system models of different complexity could potentially have very different generation profiles during times of water stress because of the constraints present in the system. To show these differences, we look at the generation profile of the four models in Figure 2.6 for one week in Arizona for GFDL-CM3 2097 scenario. This week (week 32) is chosen because it is the time of highest stress in the system with the maximum number of cumulative outages. Nuclear and coal meet the net demand in Economic Dispatch with no gas plants required to come online. Also, the variation in coal generation between hours is very high, which is unrealistic considering the downtime/uptime and ramping constraints of coal plants. The UC model represents the operations, but it overestimates the flexibility of the system to meet demand by importing energy from surrounding states. As a result, it does not have non-served energy compared to OPF and UC+OPF, which have non-served energy in times of highest water stress (last 24 hours).

OPF model represents the spatial constraints of the system but has no representation of the system's operational constraints. As a result, we see some unusual behavior with coal and combined cycle gas plants turning offline and online within a couple of hours, violating the uptime/downtime and ramping constraints for these plants. The UC+OPF model represents the operational constraints in UC and the spatial constraints in the OPF model. We observe that by representing the uptime/downtime constraints, the plants go to minimum generation instead of shutting down. The additional spatial complexity in OPF and UC+OPF constrains the system from importing energy from surrounding states, thereby resulting in some unmet demand. The OPF model provides a good approximation to UC+OPF because it has a similar generation profile to UC+OPF and has unmet demand in times of highest water stress, which is absent in UC.



(a) ED

(b) UC

(c) OPF

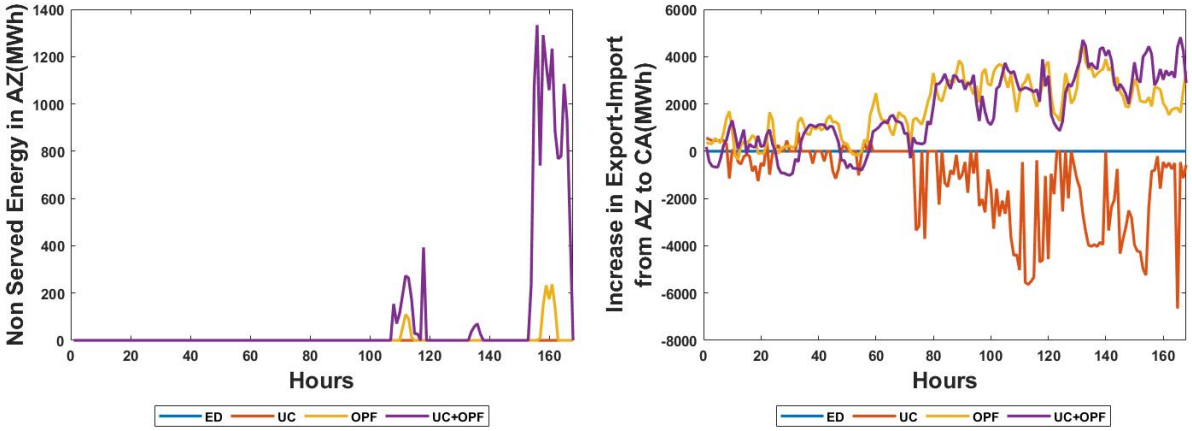
(d) UC+OPF

Figure 2.6: **2.6(a)** is the generation profile of Arizona in week 32 under water stress from GFDL-CM3 2097 scenario for ED. **2.6(b)** is the generation profile of Arizona in week 32 under water stress from GFDL-CM3 2097 scenario for UC. **2.6(c)** is the generation profile of Arizona in week 32 under water stress from GFDL-CM3 2097 scenario for OPF. **2.6(d)** is the Generation profile of Arizona in week 32 under water stress from GFDL-CM3 2097 scenario for UC+OPF. 'Other' represents renewable energy and smaller sources of energy such as Biomass and Geothermal.

2.4.2.2 Spatial Complexity Differences

The value of higher spatial complexity is observed when we look at the differences in the export-import from Arizona to California for week 32 in GFDL-CM3 2097 water stress case with the benchmark scenario as shown in Figure 2.7(b). In times of water stress, the UC model exports energy from California to Arizona, whereas OPF and UC+OPF model exports energy from Arizona to California. Arizona exports energy in OPF and UC+OPF to

California to meet the demand in California, where multiple locations with shortages could be present, and to offset an increase in the marginal price of electricity where higher-cost gas plants are present. However, this export of energy results in some unmet demand in Arizona, as shown in figure 2.7(a). An opposite phenomenon is observed in UC because California is considered a single zone. Generation in areas further away from Southern California can also be potentially exported to Arizona without considering the distances between these zones. Increasing the spatial complexity in terms of transmission lines and using buses instead of zones would restrict this flow and can give better estimates of the impacts of water stress.



(a) (b)

Figure 2.7: **2.7(a)** is the Non Served Energy in Arizona for ED, UC, OPF and UC+OPF in week 32 for GFDL-CM3 2097 water stress scenario. **2.7(b)** is the difference in export-import for week 32 in GFDL-CM3 2097 water stress scenario with respect to base scenario with no water stress.

2.4.2.3 Operations Complexity

Models with low operational complexity overestimate the flexibility of the system. To understand this phenomenon, we look at the generation profile of the T.S. Combined Cycle Gas plant in Nevada for one representative week (week 29) in the GFDL-CM3 2089 scenario, as shown in figure 2.8. ED model does not require this plant to be online for the entire period, and the OPF model oscillates between generating at maximum capacity or zero depending upon the time of the day and demand profile. In the UC and UC+OPF model, the plant remains online for the entire period, and depending upon load profile, it oscillates between the minimum and maximum generation. The OPF model does not have uptime/downtime constraints. It can oscillate between the maximum generation and zero to minimize the cost and provide the system with more flexibility.

Model	Min	Med	Max
ED	8199	8413	8517
OPF	94613	95870	97652
UC/UC+OPF	91344	91918	95770

Table 2.1: Minimum, Median and Maximum number of startups of coal and gas plants across the entire system for ED, OPF, UC and UC+OPF.

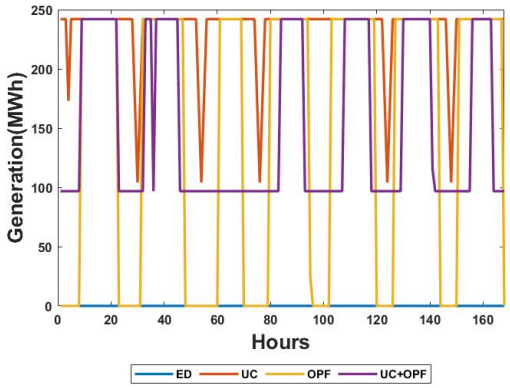


Figure 2.8: Generation profile of TS Combined Cycle gas plant in Nevada in week 29 for GFDL-CM3 20189 future water scenario for ED, OPF, UC and UC+OPF.

We also look at the number of startups for coal and gas plants across all the future water scenarios for the four models as shown in Table 2.1 . The number of startups in UC and UC+OPF is an output from the UC model and is the same for both. For the OPF and ED models, we denote startups as the number of times the generation changes from zero, conditioned upon the previous hour's generation being zero. We observe that the ED model does not require gas plants to come online much because of the flexibility to transfer generation between zones and coal plants satisfying the demand. Therefore, we see the lowest number of startups in this model. The OPF model has more spatial constraints than ED, and we see the maximum number of startups across all scenarios. This is because gas plants come online and offline frequently because of the spatial constraints but violating the operational constraints to minimize the cost of the system. This provides perceived flexibility of the system where plants can come online or offline without the operational constraints of generators. The UC and UC+OPF model have a greater number of startups than ED but lower than OPF. This is because it has more spatial and operational constraints than ED, which reduces the flexibility of coal plants to supply the demand. The UC model satisfies the operational constraints such as uptime/downtime and penalizes the startups by adding a startup cost in the objective function, which reduces the number of startups than the OPF model. From this, we can conclude that operational constraints can reduce the flexibility of the model to meet demand.

2.4.3 Economic Impacts

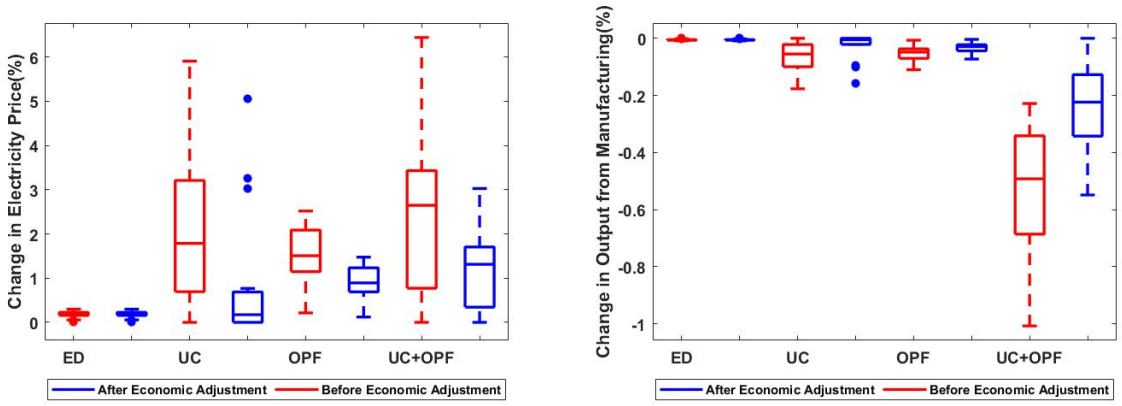
The impacts of using power system models of different complexity are not only on the power sector but also on other sectors. To measure the impacts, we propagate the electricity prices and unmet demand through a regional economic model (REM) for 18 scenarios as described in Section 3.

Models with higher unmet demand have the largest impact on electricity prices, and we observe that in Figure 2.9(a) where UC+OPF model has the maximum change in electricity price across these 18 samples without any economic adjustment. The UC model also has comparable prices to UC+OPF, although the unmet demand from UC is much lower. This is because although demand can be met within any zone because of not having transmission constraints within a zone, it is turning online expensive generators to meet this demand, thereby raising the electricity prices. The UC+OPF model cannot supply this electricity because of transmission constraints, and hence we have unmet demand, which increases the electricity price more than the UC model. The OPF model has a smaller percentage change, although it has some years with unmet demand is because the magnitude of unmet demand is very small. ED has the lowest percentage change, less than 0.5%. This suggests that using operations constraints gives better estimates of electricity price changes than adding spatial complexity in the model alone.

Along with the electricity prices, the REM also gives insights into the impacts on other parts of the economy that cannot be considered when analyzing power systems only. We consider two metrics of % change in consumption and manufacturing output to investigate impacts on the end consumer as shown in 2.10(a) and 2.9(b). We observe that the UC has a bigger change in consumption and manufacturing than OPF and ED, although OPF has a greater number of years with unmet demand but with lower magnitude. The UC+OPF model has the largest % change, although the electricity prices are like UC because of the magnitude of unmet demand and its productivity impact. The effect of the unmet demand on productivity is higher than electricity price changes, as shown in Webster et al. [57].

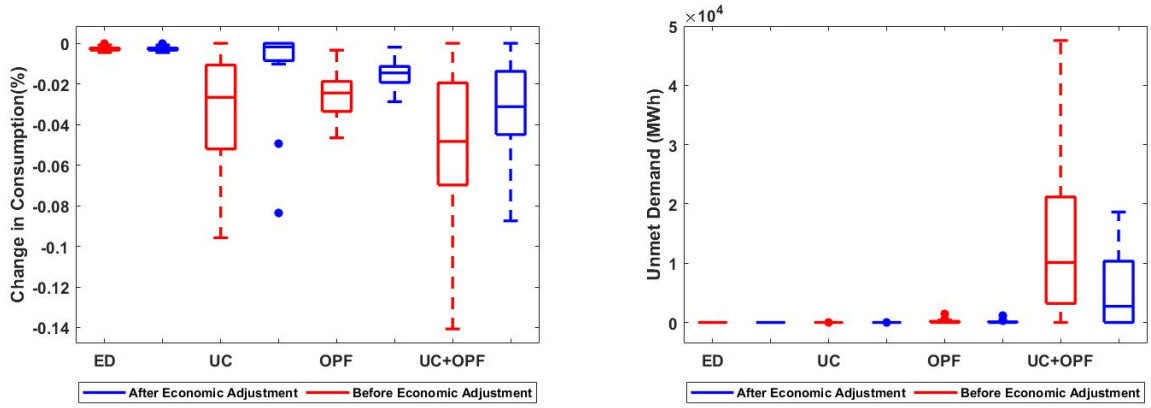
An important characteristic of REM is that it can capture the end effects on consumers and producers because of higher electricity prices and unmet demand. The demand from REM is then adjusted based on the productivity impacts, which are passed over to the PSM. This iterative process is continued until convergence is achieved, as described in Section 2.4. We observe that for all the models, the electricity price, consumption, and manufacturing decreases after accounting for economic adjustment because of the reduction in demand as shown in figure 2.9 and 2.10. However, we note that the % change in manufacturing and

consumption after economic adjustment for UC+OPF is still greater than before adjustment for UC, ED, and OPF in many future water scenarios. This is because even after economic adjustment, the UC+OPF model has higher unmet demand compared to the other models as shown in figure 2.10(b). Detailed impacts on each sector by ED, UC, OPF and UC+OPF after equilibrium are described in Appendix B.



(a) (b)

Figure 2.9: **2.9(a)** is the electricity prices before and after economic adjustment for 18 future water scenarios. **2.9(b)** is the % change in manufacturing output before and after economic adjustment for 18 future water scenarios. **2.10(a)** is the % change in consumption before and after economic adjustment for 18 future water scenarios.



(a) (b)

Figure 2.10: **2.10(a)** is the % change in consumption before and after economic adjustment for 18 future water scenarios.**2.10(b)** is the change in unmet demand before and after economic adjustment for 18 future water scenarios.

2.5 Trade-offs in Increasing Model complexity

Increasing model complexity by adding spatial or operational constraints gives a more valid estimate of impacts in terms of electricity prices, unmet demand, and end consumer effects. The ED model has excess flexibility compared to the other models, and we observe that in the total number of startups in the model. Adding operations complexity to the model alone helps to represent daily operations of the power grid better in the generation profile of individual generators. Still, it has excess flexibility because of low spatial resolution and poor network representation. This system alone would not be beneficial to estimate impacts from water stress because the outages can be in one region with the impacts of these outages in another region. This is because of the interconnected nature of the power grid, which is not represented in the UC model well.

Adding spatial complexity is more beneficial than operations complexity alone for two reasons: Firstly, it can represent the grid transmission lines and restrict flows between regions, thereby resulting in scenarios with unmet demand similar to UC+OPF model, although with a lower magnitude. These scenarios also can represent similar locations of high stress in the system to the UC+OPF model. Secondly, the computational effort associated with solving the OPF model is eighteen times lower than the UC model, as shown in table 2.2. This could especially be advantageous when interacting with more than one model in a multi-sector

dynamics framework. The OPF model is also helpful during the model building process to test new methods and when high-performance computing abilities are unavailable. One downside of using OPF is that flexibility increases regarding startups and no ramping constraints, but this approximation is potentially less detrimental than using operations complexity alone when determining locations of high water stress.

Model	ED	UC	OPF	UC+OPF
Variables(Cont.)	6,10,009	12,95,281	8,66,713	12,95,281
Variables(Discrete)		5,14,416		5,14,416
Constraints	2,185	17,96,393	1,62,289	17,96,393
Solution Time(sec)	24	4926	264	5324

Table 2.2: Characteristics of ED, UC, OPF and UC+OPF in terms of number of continuous variables, discrete variables, constraints and total solution time for one week. The solution time is averaged over 52 weeks of the benchmark scenario for each model.

Adding operational complexity alone is not as beneficial as spatial complexity but adding both gives the best of the two models. Using spatial complexity alone gives locations of high water stress to inform managers to make reinforcements, but the electricity prices and end effects on consumers are diminished. Thus not giving a valid estimate of impacts. The decomposition framework of UC+OPF makes it computationally tractable and does not increase the computation effort substantially more than the UC model. In some instances, when representing operations is necessary, it will not be very computationally intensive to add the spatial complexity to get better estimates of impacts.

2.6 Conclusions

This work compared the effects of using a detailed power systems model with higher spatial and operations complexity than using models focused on one of these constraints under water stress conditions. Omitting one of these constraints could potentially increase the flexibility of the system than what is present. We observed that not considering operations resolution could affect the generation profiles of conventional fuel plants because of excluding ramping, uptime, and downtime constraints. Spatial constraints are necessary to model the power flow across lines within and between states, and omitting these constraints would potentially increase the system's flexibility to supply generation under water stress. Considering both these constraints in a computationally tractable framework by separating them and using

generators' on/off status as linking constraints between the two models, we show that neglecting these constraints could reduce the impacts from water stress in terms of unmet demand and electricity price.

The impacts of water stress are not only on power systems but on other parts of the economy. We show that the impacts of using power systems of low spatial and operations complexity underestimate the impacts on the consumption and manufacturing output reduction because of water stress. This is important because water stress affects the direct physical systems such as energy and can indirectly affect other parts of the economy. Using power systems with higher complexity could give valid estimates of impacts. We observed that spatial constraints are more beneficial than operational constraints to the model in identifying locations of higher water stress and are computationally less intensive. Operational complexity is also helpful to accurately model the flexibility of plants in the system and to get a better estimate of electricity prices and end effects on the consumer, but it comes at the cost of higher effort. The modeler must decide whether this is required or representing spatial complexity is enough. It may be enough to determine locations of high water stress to inform managers about future grid reinforcements. In this work, the insights were obtained using a representative case study of the Western Electricity Coordinating Council (WECC). We believe that these insights can be transferable to other power systems. The testing on other systems to validate these results is an area of future work.

It is important to note that there could potentially be changes to the power grid in the future that could change these impacts, but this work is more focused on showing the differences between using power system models of different complexity and which constraints affect the system the most. Future changes to the power grid, such as renewable energy penetration, would require the representation of more operations and spatial detail than what is currently presented in a coupled energy-economy framework. This is an area of future work. This work has also not considered temporal complexity, where the entire year operation is approximated by simulating few weeks of the year. Many approaches have been proposed to approximate the whole year. It would be interesting to understand which approach and how many hours/days/weeks are necessary to be simulated to approximate an entire year of operations well enough to obtain valid estimates of impacts from water stress.

Chapter 3
Importance Sampling based exploration in Q Learning

3.1 Introduction

Multi-stage stochastic optimization refers to the class of problems where the uncertainty evolves dynamically, decisions are adaptive to the realized uncertainty, and there are three or more stages. For many practical situations, multi-stage stochastic optimization may have a multidimensional and continuous feasible decision space, a multidimensional continuous uncertainty space, or a finite scenario tree with many scenarios. Such problems will generally not be solvable with exact formulations, such as a deterministic equivalent stochastic program [107]. A variety of approaches have been developed for solving very large multi-stage stochastic problems in continuous decision space. One class of algorithms builds on the stochastic programming paradigm and extends two-stage decomposition schemes such as Benders Decomposition [108] or Progressive Hedging [109]. However, these methods depend on the existence of dual variables in the subsequent stage or on efficient and stable quadratic programming solvers and can be difficult to extend to problems with non-convex constraints, although some recent work has been done to address this [33, 35, 110].

Another class of methods applied to continuous state and decision spaces has developed from Stochastic Dynamic Programming, and Markov Decision Problems [111]. Two approaches for solving dynamic programming problems in continuous spaces are discretizing the state and action space [112] to obtain the policy and the use of analytical closed-form solutions for the optimal policy when such a solution exists [113]. The closed-form solution can be applied to specific problems such as Linear Quadratic Control problems, whose analytical solution can be found by solving the Riccati equations. However, extending this approach to problems

with a non-linear structure is difficult because an algebraic solution cannot be obtained. An advantage of discretization is that it does not depend on the problem characteristics (e.g., the convexity of the space), but it suffers from the Curse of Dimensionality and can become computationally intractable for problems with high dimensional continuous decision and state spaces. For very large problems with both continuous state and decision spaces, Monte Carlo sampling-based approaches such as Approximate Dynamic Programming (ADP) methods [114] or Reinforcement Learning (RL) methods [115] can find good approximate solutions to the original problem by sampling uncertainties and decisions and using low-order functional approximations to the value or recourse function. These methods are generally computationally less expensive than an exact Stochastic Dynamic Programming approach and can be used in a multi-stage adaptive framework.

The computational advantage of ADP lies partly in using sampled costs for specific states and actions to estimate a functional approximation that extrapolates the information to other potential states and actions not sampled. Functions can be used to approximate either the value function, which maps states to optimal expected values, or a state-action value function, which maps a state-action pair to the expected value of choosing that action in that state and subsequently following the optimal policy. We focus on state-action approximations applied to continuous state and action spaces within a Q-Learning framework [116].

Q-learning uses a low dimensional approximation of the objective function and progressively updates the approximation as it observes the costs for sampled states and actions. The algorithm's goal is to learn a policy that mimics the optimal policy obtained by exact Dynamic Programming but with less computational effort. An efficient sampling methodology is required when the complete enumeration of the feasible decision space is computationally prohibitive. Q-learning algorithms must balance the tradeoff between sampling state-action pairs where the variance is large to improve the quality of the estimate and exploiting the existing estimate to minimize the cost. This is commonly known as the dilemma of exploration vs exploitation [114,117,118]. Several approaches have been developed to navigate this tradeoff, of which one widely used method is epsilon greedy [119]. Epsilon greedy randomizes the strategy between pure exploration, i.e., sampling any feasible action, and pure exploitation, i.e., choosing the optimal action from the current estimated state-action value function. The relative ratio of explore vs. exploit samples will generally be a function of the iteration count. One challenge for epsilon greedy and similar methods is that in early iterations, the value function approximation is poor, so that the exploit samples may not be useful or can be misleading, and in later iterations, as the approximation improves, exploration samples do not

utilize the global or past information and may inefficiently continue to sample sub-optimal actions. A second challenge is that these methods each have algorithmic parameters that must be heuristically tuned for each application, and convergence can be quite sensitive in practice to the values chosen. The third challenge to epsilon greedy is that an additional optimization problem must be solved to obtain each exploit sample action, increasing the computational cost considerably if the value function approximation is non-linear or non-convex.

Importance Sampling (IS) [120, 121] is a variance reduction technique that disproportionately samples from a distribution other than the "true" distribution to estimate the expected value with fewer samples. Importance Sampling has been widely applied in Stochastic Programming [122, 123] and Reinforcement Learning [124, 125] to provide variance reduction. IS within Stochastic Programming has been applied to efficiently sample large uncertainty spaces, while in these methods, the decisions are obtained by solving linear programs. In Reinforcement Learning, Importance Sampling has been used to eliminate bias when the behavioral or sampling policy differs from the optimal policy. In the update step to improve the approximate value function, the sampled rewards are weighted using the density ratio to reflect the likelihood of sampling the state-action pair. However, these applications use other methods for selecting the samples. Importance Sampling offers a potentially powerful means of sampling actions, using information learned in prior iterations without going to the extreme of pure exploitation. The approximate value function can be used within an accept-reject framework to explore actions disproportionately in the neighborhood of the current optimum, with some exploration of other actions still occurring.

In this chapter, we present a novel sampling scheme to select actions in continuous space within a Q-learning algorithm based on importance sampling. We propose to use an Importance Sampling weighted accept-reject method to sample actions; i.e., importance sampling defines the sampling policy rather than weighting observed rewards from some other sampling policy. We exploit the characteristic of disproportionate sampling in IS to balance exploration and exploitation. We apply this method to multi-stage Stochastic Generation Expansion Planning (GEP), the problem of choosing electricity generation investments in each stage under uncertainty. GEP provides an example of a problem with multidimensional continuous feasible action and state spaces. In the context of this application, we compare the performance of our proposed sampling scheme to two alternative methods used in Approximate Dynamic Programming and Reinforcement Learning for continuous decision spaces, epsilon greedy [119] and epsilon decay. In particular, we demonstrate that the latter methods for decision selection can sometimes result in convergence to sub-optimal policies and that

different algorithmic parameter choices can result in widely differing policies from replications of the same problem. In contrast, we show that the importance sampling-based approach converges to higher quality approximate solutions with less variance across repetitions and that the algorithm increases the density of sampling near-optimal actions as the value function approximation improves.

The contributions of this chapter are:

- A novel algorithm based on importance sampling to sample actions in continuous decision space for Q-learning.
- Application of the method to multi-stage stochastic generation expansion planning and comparison with epsilon greedy sampling methods.
- Demonstration of convergence to more accurate approximate solutions with less variance, without any algorithmic parameters.

The remainder of the chapter proceeds as follows. Section 3.2 reviews sampling methods within Approximate Dynamic Programming and Reinforcement Learning algorithms and the use of Importance Sampling in Reinforcement Learning. Section 3.3 introduces Markov Decision Processes. Importance Sampling is defined in Section 3.4. Section 3.5 presents our Q-learning Importance Sampling method. The application problem, multi-stage generation expansion planning, is introduced in Section 3.6. The proposed algorithm is compared with other sampling methods in Section 3.7, and Section 3.8 provides a concluding discussion.

3.2 Literature Review

Approximation architectures provide a computationally tractable approach to solving high dimensional state-action spaces when exact algorithms cannot provide solutions in reasonable computational time. Approximate Dynamic Programming [114] and Reinforcement Learning [115] offer a general framework that can provide solutions to complex, high dimensional optimization problems under uncertainty. The algorithms within this class of methods use a combination of statistical methods for sampling and estimation, optimization methods, and lower-dimensional representation of the objective function to obtain approximate solutions to the original problem.

The performance of Monte Carlo-based methods depends critically on efficiently adaptively sampling the state-action space to learn the best approximation of the true value function.

The challenge for any sampling policy is that actions that appear sub-optimal at the beginning of the algorithm when the approximation is less accurate may be optimal and vice versa. It is important to sample actions based on the current expected reward and balance this with exploring the action space to improve the policy. This tradeoff is known as the 'explore vs. exploit' problem and has received considerable attention in the research community [114, 117, 118]. Many sampling methods and heuristics have been developed in the literature to better balance exploration and exploitation, including epsilon-greedy, R-Max [126] and E^3 [127]. R-Max and E^3 require enumeration of the state space, which is possible for some reinforcement learning applications but becomes computationally intractable for large or continuous state spaces. Epsilon greedy uses an algorithmic parameter to prescribe the relative frequency of exploration and exploitation. Epsilon decay defines a trajectory for the explore/exploit ratio that changes with the iteration count. In practice, the algorithmic parameters in these methods that specify the relative frequency of exploration and the rate of change in the ratio can be difficult to tune for each application. Epsilon greedy methods have an additional disadvantage for some applications; every exploit sample is generated by solving for the optimal action relative to the current value function approximation. If the approximation architecture is non-linear, this can increase the computational burden.

An active area of research in the Reinforcement Learning literature to promote exploration is the application of an exploration bonus [128, 129] or intrinsic motivation [130, 131] to assist the algorithm in exploring areas that would not otherwise be visited but that will improve the approximation and future decisions. The fundamental idea in these algorithms is to add a bonus term to the Bellman Optimality equation based on the state-action visit counts. Although theoretical guarantees exist for tabular implementations [129], only recently has it been applied to high dimensional state spaces by using a density model over the state space [131–133], a hashing method to cluster similar states [134] or by definition of another functional approximation for the exploration count [128]. All these methods require the state-action space to be countable, limiting the applicability to continuous settings. Further, approaches that rely on modifying the reward as an incentive to explore may not work well for problems in which actions strongly impact future rewards and decisions, such as the capacity expansion problem.

Another strategy for exploration has been the use of Bayesian methods in Approximate Dynamic Programming. Early examples defined a prior over the transition probabilities [135] and applied this approach to multi-armed bandit problems [136]. Variants of this methodology also define a prior over the value function itself [137]. Bayesian methods

were extended to account for correlation in the impact of individual actions across many states [138]. The Knowledge Gradient metric was developed to sample actions within a Bayesian framework [139]. Although the Knowledge Gradient metric evaluation does not require new parameters, the Bayesian model to which it is applied requires the definition of a prior over the value function and requires the specification of a covariance matrix, which requires practical knowledge of the underlying system. The Knowledge Gradient approach can be extended to continuous state spaces but requires the enumeration of the action space and cannot be applied to continuous action spaces without discretization.

Reinforcement Learning for problems with continuous action spaces has applied policy gradient methods [140, 141] and cross-entropy search methods [142, 143]. However, these methods have faced challenges when defining a policy over multidimensional action spaces or within a constrained action space. Actor-critic [144, 145] methods integrate value function approximation and policy gradient algorithms for such multidimensional action spaces, but the iterative estimation of both value and policy function approximations can substantially increase the computational demands. Recently, new variants of value function approximation methods that use neural network architectures have sampled actions using normalized advantage functions [146], convex inputs to a neural network [147] or mixed integer-based formulations [148]. All the above require the optimization of an approximate analytical function to generate proposed samples during the exploit phase and can result in sub-optimal actions if the feasible action space does not satisfy the assumed structure of the analytical function. The above methods are specific to neural networks and may be difficult to adapt to other approximation architectures.

Importance Sampling has been widely applied in the Reinforcement Learning community since it was first introduced using per-decision importance sampling in tabular reinforcement learning [124] and subsequently expanded to include linear function approximations [125]. The problems addressed in the reinforcement learning community typically have long episodes or sample paths; in this context, per-decision importance sampling provides an unbiased estimator with variance reduction so that fewer samples are required to converge [124]. Early success with this approach motivated the development of variance reduction techniques such as weighted importance sampling [149], adaptive importance sampling [150], emphatic TD learning [151], doubly robust methods [152], conditional importance sampling [153] and incremental importance sampling [154]. All these methods focus on facilitating an unbiased update step when the sampled reward results from an action chosen from a policy other than the current optimal policy. Prediction or behavior policies, which generate the

sample state-action pairs, have included epsilon greedy [154], uniform random sampling [149, 152, 155, 156], estimation through multiple trajectories [157], softmax distributions [158] or specified probability distributions of actions [124, 125, 151, 159]. These methods tend to sample actions from the full feasible space without conditioning on the previous samples observed, except when using the current approximate value function to exploit. For example, if we consider epsilon greedy as the behavior policy, an explore sample may randomly select any feasible action. But this sampling policy provides no additional guidance on the relative benefit between actions for improving the approximation. When every action cannot be visited in large or continuous action spaces, the behavior policy should sample the actions that will provide the maximum improvement to the value function approximation. In this paper, we demonstrate the use of importance sampling to guide the sampling of actions in a continuous space, rather than simply correcting the bias for sample rewards obtained by some other means.

3.3 Markov Decision Processes

Markov Decision problems are commonly used for modeling multi-stage optimization problems under uncertainty. The following elements define Markov Decision Processes:

- Stages ($t \in T$): Specify the specific points in time where an action is chosen by the agent based on the information available up to that point.
- State ($s \in S$): The vector of information that makes the process Markovian; i.e., it contains all information required to specify the reward to be received and the state of the process in the next stage that will result from each feasible action.
- Action ($a \in A$): A feasible action that can be chosen by the agent in the current state and stage.
- Exogenous Information ($\omega \in \Omega$): The new information obtained between stages $t-1$ and t for every state; represents the uncertainty in the system.
- State Transition Function: Represents how the state variable changes due to the chosen action and the realized exogenous information. Often takes the form of a probability distribution $P_{s,s'}(s, a, \omega)$.
- Cost or Reward (r): This is the immediate reward or cost to the agent due to choosing action a in state s; it may be deterministic or stochastic.

- Policy (π): A policy maps states to a probability distribution over its actions. The mapping can be invariant over time (stationary) or can change as a function of its history (non-stationary).

There are several distinct classes of MDPs, depending on whether it is a finite horizon or an infinite horizon problem and on the form of the total cost over all stages [112]. Here, we restrict our focus to finite-horizon problems in which the total cost is the discounted sum of costs in each stage. We will therefore include the subscript t to indicate the dependence on stage with the probability transition function defined as $P_{s_t,s_{t+1}}$ for the state transition from stage t to $t+1$. We implicitly include the uncertainty ω_t within the state variable s_t and the transition function $P_{s_t,s_{t+1}}$ to simplify notation. The formulation applies to either total reward maximization or total cost minimization; without loss of generality, we present the cost minimization notation to be consistent with the application in this paper. The expected value of following any feasible policy π is

$$v_t^{\pi}(s) = E_{\pi}\left\{ \sum_{t=1}^{T} \gamma^t r_t \middle| s_t = s \right\} \tag{3.1}$$

This represents the expected value of the sum of discounted costs when the system begins in-state s and then selects actions based on π for all remaining stages. In Equation (3.1), r_t is the cost and s_t is the state at time period t and γ^t is the discount factor.

The objective is to solve for the optimal policy π^* that minimizes the value function (3.1):

$$v_t^*(s) = \min_{\pi} v_t^{\pi}(s) \quad \text{for all} \quad s \in S, t \in T \tag{3.2}$$

The exact solution of the finite horizon MDP in (3.2) can be solved with dynamic programming, a stage-wise decomposition of the full problem into a sequence of smaller problems for each stage and state. The solution for each subproblem is found via the Bellman Equation with $P_{s_t,s_{t+1}}$ representing the probability distribution of the state transition function:

$$v_t^*(s_t) = \min_{a_t} E_{P_{s_t,s_{t+1}}}\left\{ r_t(s_t, a_t) + \gamma v_{t+1}^*(s_{t+1}) \right\} \quad \text{for all} \quad s \in S, t \in T \tag{3.3}$$

The exact solution of (3.3) can be obtained by backward induction, starting in the final stage T and solving for every state, and then moving back one stage and repeating until solved for $t = 1$. This approach is not computationally tractable once the stages, states, actions, and exogenous information signals become too large. Approximate algorithms (e.g.,

ADP, RL) focus on iteratively sampling state-action pairs (s_t, a_t), observing the sampled costs r_t, and updating a low-order approximation of the value function $\hat{v}_t$.

An alternative form of the value function known as the state-action value function, is useful for some applications:

$$q_t(s_t, a_t) = E_{P_{s_t,s_{t+1}}}\{r_t(s_t, a_t) + \gamma \min_{a_{t+1}} q_{t+1}(s_{t+1}, a_{t+1})\} \quad \text{for all} \quad t \in T \tag{3.4}$$

The solution minimizes

$$v_t^* = q_t^*(s, a) = \min_{\pi} q_t^{\pi}(s, a) \quad \text{for all} \quad s \in S, a \in A, t \in T \tag{3.5}$$

A sampling-based approach approximates the state-action value function $q(s, a)$ with a low-order projection in terms of a set of features/basis functions that depend on the state and action [118]. The approximations can be linear or non-linear and an extensive list of functional forms is described in Sutton and Barto [115]. The simplest functional form is a linear approximation with coefficients or weights θ. The algorithm solves for the approximate function $\hat{q}$:

$$\hat{q}_t(s, a, \theta) \approx q_t(s, a) \quad \text{for all} \quad t \in T \tag{3.6}$$

The approximation is updated iteratively by sampling specific state-action pairs with temporal difference learning [115] as in Equation (3.7),

$$\hat{q}_t(s, a, \theta) \leftarrow (1 - \lambda)\hat{q}_t(s, a, \theta) + \lambda(r_t(s_t, a_t) + \min_{a_{t+1}} \hat{q}_{t+1}(s_{t+1}, a_{t+1}, \theta)) \quad \text{for all} \quad t \in T \tag{3.7}$$

The second term in Equation 3.7 can be estimated by looking n steps ahead, where n can take any value from 1 (Equation 3.7) to T (Equation 3.8).

$$\hat{q}_t(s, a, \theta) \leftarrow (1-\lambda)\hat{q}_t(s, a, \theta)+\lambda(r_t(s_t, a_t)+r_{t+1}(s_{t+1}, a_{t+1})...r_T(s_T, a_T)) \quad \text{for all} \quad t \in T \tag{3.8}$$

In equation (3.7, 3.8), we can either use a single state-action sample for the update or we can perform an expected update using a batch of samples. Although the computational complexity is higher for the batch sample update, performance improvement is often observed.

3.4 Importance Sampling

Consider the general problem of estimating the expected value of a function $h(x)$ of a random variable x with probability density function $f(x)$.

$$\mathcal{L} = E_f[h(X)] = \int h(x)f(x)dx \tag{3.9}$$

For estimating extreme values or rare events, the number of samples of x required may be very large. In these cases, a useful strategy is to sample from a different distribution $g(x)$ such that $h(x)f(x)$ is dominated by $g(x)$. Because the expectation is taken with respect to the original distribution $f(x)$, each sample must be weighted by its relative likelihood:

$$\mathcal{L} = \int h(x)\frac{f(x)}{g(x)}g(x)dx \quad g(x) > 0 \text{ for all } x \tag{3.10}$$

Then an unbiased Monte Carlo estimator of $\mathcal{L}$ is

$$\hat{\mathcal{L}} = \sum_i h(x_i)\frac{f(x_i)}{g(x_i)}g(x_i) \tag{3.11}$$

The choice of the importance density function $g(x)$ is critical in determining the variance reduction that can be achieved. It can be shown [160] that the variance of $\hat{\mathcal{L}}$ is minimized by sampling from

$$g^*(x) = \frac{|h(x)|f(x)}{\int |h(x)|f(x)dx} \tag{3.12}$$

The minimum variance density has long been elusive because of the curse of circularity: the denominator in (3.12) is the very quantity we are estimating. Various heuristics have been used to select "good" importance sampling densities (see [160]). More recently, it has been shown that Markov Chain Monte Carlo methods can be used to sample from the optimal density function [123] up to a proportionality constant. Note that if the original density $f(x)$ in (3.12) is uniform (i.e., $f(x)$ is a constant), than the optimal density function $g(x)$ is simply a normalized version of the function $h(x)$ to be estimated. We exploit this fact below in our algorithm.

In addition to statistical estimation and simulation, importance sampling has been used extensively in reinforcement learning [115]. In many RL algorithms, the expected value (the second term in the Bellman equation) is estimated by looking n-steps ahead, as shown in Equation 3.8. When temporal difference learning is applied for two or more steps ahead,

additional state-action pairs must be sampled using the current approximation of the optimal policy. Efficiency is improved by sampling state-action pairs for the additional steps from a policy different from the optimal policy, known as an off-policy approach. Because these samples are used to estimate the expected value of future decisions, the likelihood of future rewards must match the likelihood of visiting those states in the learned/optimal policy. Therefore, in the prediction step of these algorithms, off-policy sampling uses the importance sampling ratio to correct these estimates [125, 149].

The original method of importance sampling in estimation (3.10-3.11) performs two distinct tasks. The first is to *sample* more efficiently by focusing the measurement or computation on the most useful regions of the sample space. The second is to then *correct the bias* from disproportionately sampling in the estimated expected value. The applications of importance sampling in off-policy reinforcement learning only perform the latter of these functions while using other methods (e.g., epsilon greedy or uniformly random) for selecting the state-action samples [149, 154, 156].

In the proposed method, we focus on the former objective, efficiently sampling actions. Building on the insight that the minimum variance sampling density $g(a)$ is at least proportional to the function being estimated $\hat{q}(s, a)$, and adapting the MCMC sampling approach from [123], we use an accept-reject algorithm to adaptively sample actions using the current estimated approximate Q-function $\hat{q}$.

We apply this sampling approach to a one-step Q-learning framework. As noted in [115] (see Sections 7.3 and 12.10), when sampling off-policy in one-step algorithms, sample action a only impacts the reward $r(s, a)$ and the state transition $P_{s,s'}(s, a)$ and therefore the importance sampling ratio is not needed to correct the bias. In other words, in a one-step Q-learning framework, we are estimating the value of $\hat{q}(s, a)$ for the action a that has been sampled already and not the expected value of many different values of a. Further, Q-learning directly approximates the optimal Q-function(q) and is sampling over the state/uncertainty transitions once the action a is selected using the off policy approach. Therefore, we do not need to perform the bias correction step with the importance sampling ratio. Note that the correction would be required if i) one also applied importance sampling to the exogenous uncertainty that determines the state transitions, or ii) if applying this sampling approach to other RL algorithms with two or more steps ahead sampled off-policy. In these cases, one could adapt the Kernel Density Estimation procedure presented in [123].

3.5 Q-learning Importance Sampling

We present the Q-learning Importance Sampling (QIS) method (Algorithm 1). The name of the algorithm derives from the fact that it uses the function approximation $\hat{q}$ within an importance sampling framework to sample actions. The QIS algorithm uses accept-reject sampling, where the likelihood of accepting a random action given the current state is proportional to the current approximation of the Q-function $\hat{q}$. Proposed sample actions ξ are generated, and the approximate value $\hat{q}(\xi)$ for the proposed action is obtained using the current approximation $\hat{q}$. Proposed samples are accepted or rejected based on the $qRatio$, as defined in Equation 3.13,

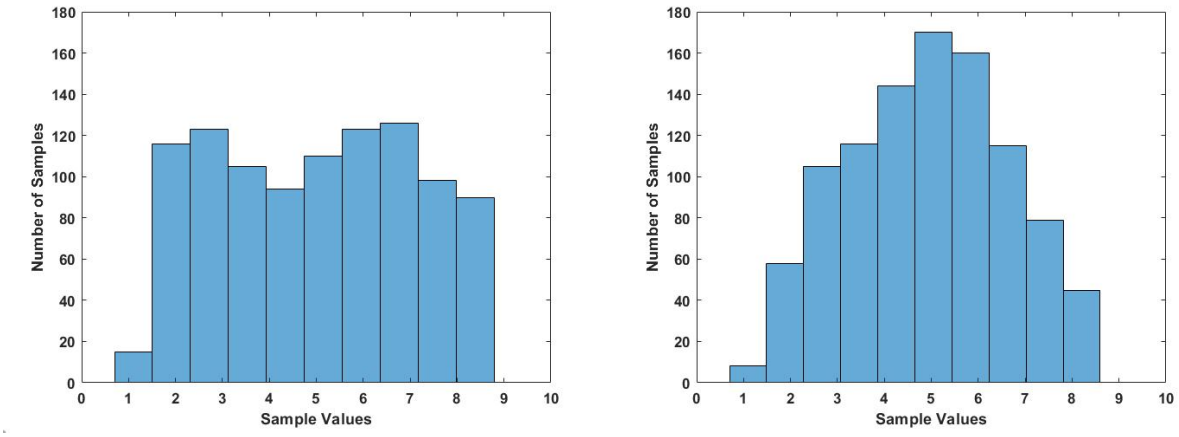
$$qRatio = \frac{qMax - \hat{q}(\xi)}{qMax - qMin} \tag{3.13}$$

where $qMax$ and $qMin$ are the minimum and maximum approximate $\hat{q}$ function values determined each iteration from all samples observed so far. In early iterations when the function approximation is poor, the estimates of $qMin$ will also be poor and the probability of a sample in the entire space being accepted will be high because the denominator of $qRatio$ may be small. As more samples are obtained, the estimate of $qMin$ will improve and the probability of proposed samples with values closer to $qMin$ will become more likely to be accepted than samples with values closer to $qMax$, although no proposals will have zero probability of being accepted. The result is a shift in the distribution of sampled actions from more uniformly distributed in the early iterations when the approximation is poor to distributions with higher density near the optimal action and lower density elsewhere in later iterations. Unlike epsilon greedy, this method will never perform a pure exploit sample, sampling exclusively the optimal action for a state as determined from the approximate Q-function $\hat{q}$.

We illustrate this characteristic of the QIS algorithm on a small one-dimensional problem in Example 1. Figure 3.1 shows the initial and final 1000 samples accepted in Example 1 for one representative replication. We observe that the initial 1000 accepted samples are approximately uniformly distributed, and the distribution of the final 1000 accepted samples exhibits a higher concentration of samples around the optimal action.

Example 1. *Minimize the following function:*

$$Q(x) = 25 + (x - 5)^2 \tag{3.14}$$



(a) Initial 1000 Samples

(b) Final 1000 Samples

Figure 3.1: Initial and Final 1000 Samples of QIS algorithm applied to Example 1.

Assume that the initial estimates of the minimum and maximum Q-function values are equal to 35 and 40, respectively. The true minimum of (3.14) is 25 at x=5. We perform K=5 iterations of QIS with M=1000 samples at every iteration. Figure 3.1 shows the initial and final 1000 samples accepted by QIS for one representative replication.

The detailed steps of QIS are given in Algorithm 1. The algorithm iteratively samples each stage t within each iteration k. Steps 1-2 of the algorithm initialize the parameter values for the algorithm, the Q-function approximation $\hat{q}_t$ coefficients for stages $t = 1, .., T$, the approximate minimum ($qMin_t$) and maximum ($qMax_t$) function values for each stage, and the initial state of the system s_0. If bootstrapping is not performed, the approximations for all stages will have coefficients of zero and approximate minimum($qMin_t$) and maximum($qMax_t$) function values assigned the values 0 and 1, respectively.

For each iteration k and stage t, the algorithm performs a forward pass (Steps 4-35) in which it selects M samples of state-action pairs for each stage, followed by a backward pass (Steps 36-41) in which it uses the costs obtained for these samples to update $\hat{q}_t$. In the forward pass, for all stages $t \geq 2$, M samples of the exogenous uncertainties $\omega_t \in \Omega_t$ are generated to create the state vector s_t^m (Steps 6-15). Steps 16-33 generate sample actions for each sample state. An action ξ is proposed and its value, $\hat{q}(\xi)$, is calculated using the current value function approximation $\hat{q}_t$. $qMax_t$ and $qMin_t$ are updated if $\hat{q}(\xi)$ is greater than $qMax_t$ or less than $qMin_t$. The proposed sample action is accepted if its $qRatio$ (3.13) is greater than a sample from $U(0, 1)$ (Steps 24-30). The cost for this state-action pair is $r_t^m(s_t^m, a_t^m)$. The process iterates until M samples are accepted for each stage. Steps 36-41 constitute the

Algorithm 1 QIS

Initialize $\hat{q}_t$ for $t = 1..T$
Initialize $qMin_t$ and $qMax_t$ for $t = 1..T$ and s_0
for $k = 1..K$
    Begin Forward Pass
    for $t = 1..T$
        if $t = 1$
            for $m = 1..M$
                $s_t^m = s_0$
            end for
        else
            for $m = 1..M$
                Sample $\omega_t^m \in \Omega_t$
                $s_t^m = [a_{t-1}^m \;\; \omega_t^m \;]$
            end for
        end if
        while $m \leq M$
            Propose an action ξ and calculate approximate value $\hat{q}(\xi)$ using $\hat{q}_t$.
            if $\hat{q}(\xi) > qMax_t$
                $qMax_t = \hat{q}(\xi)$
            end if
            if $\hat{q}(\xi) < qMin_t$
                $qMin_t = \hat{q}(\xi)$
            end if
            Calculate $qRatio = \dfrac{qMax_t - \hat{q}(\xi)}{qMax_t - qMin_t}$
            Generate a random number $U \sim$ U(0,1)
            if $qRatio > U$
                $m = m + 1$
                $a_t^m = \xi$
                Calculate $r_t^m(s_t^m, a_t^m)$.
            else
                Proposed action ξ is rejected.
            end if
        end while
    end for
    End Forward Pass
    Begin Backward Pass
    for $t = T..1$
        Update $\hat{q}_t$ using M samples in forward pass using equation 3.7.
        Obtain new estimates of $qMax_t$ and $qMin_t$ using all samples for iterations 1 to k.
    end for
    End Backward Pass
end for

backward pass of the algorithm, in which the function approximation $\hat{q}_t$ is updated using temporal difference learning (3.7) with the M sample costs obtained in the forward pass. New estimates for $qMax_t$ and $qMin_t$ are obtained using the updated approximation $\hat{q}_t$. The algorithm repeats until the maximum number of iterations K is reached.

The QIS algorithm has two components: 1) the accept-reject scheme using $qRatio$ in the forward pass to obtain new sample actions, and 2) in the backward pass, after the value function approximation, $\hat{q}_t$, is updated, the estimates of $qMax_t$ and $qMin_t$ are updated using all the samples observed from previous iterations. This update of $qMax_t$ and $qMin_t$ necessarily entails re-evaluating the new approximate Q function values for every prior state-action sample because these values will change for the updated approximation, which will alter the approximate maximum and minimum observed Q-function values for each stage. However, this update procedure becomes computationally expensive as the number of iterations and cumulative samples increase. To reduce the computational burden, we propose a variant of QIS that updates the estimates of $qMax_t$ and $qMin_t$ every $\hat{K}$ iterations, where $\hat{K}$ can take values between 1 (QIS) and K (never update). The required computation for re-evaluation is decreased by a factor of at least $\hat{K}$. We call this algorithm QIS-RE because of the reduced frequency of updating $qMax_t$ and $qMin_t$.

There are three advantages to the QIS algorithm. (i) The first advantage is that it can be applied to continuous action spaces because it does not require complete enumeration or discretization of the feasible actions. (ii) A second advantage is that it does not require the solution of an optimization problem during the sampling step. In the epsilon greedy algorithm, when an action is to be sampled using exploitation, the optimal action for the current state must be determined using the current value function approximation $\hat{q}$. For many applications, the functional form of the Q-function approximation, $\hat{q}$, will include non-linear terms to accurately represent the true Q-function, q; in such cases, a non-linear program must be solved to obtain the next sample action. In QIS, the current value function approximation, $\hat{q}$, is evaluated to determine the $qRatio$ for a proposed sample, but it is not necessary to solve an optimization problem. (iii) A third advantage to the QIS algorithm is the absence of tuning parameters for sampling actions. In epsilon greedy, ϵ is a heuristic parameter that is difficult to tune for each specific problem. ADP/RL algorithms require other assumptions, for example, the basis function to be used for the low dimensional approximation of the true Q-function, q or the parameter λ in temporal difference updating (3.7). Each additional heuristic algorithmic parameter further increases the difficulty in achieving convergence and reasonable approximate solutions for each new application.

3.6 Application: Multi-stage Generation Expansion

We demonstrate the QIS algorithm with an application to a multi-stage stochastic electricity generation expansion problem. Stochastic Generation Expansion Planning (SGEP) is a long-term investment problem that solves optimal new capacity investments in each stage that minimize the expected total (capital plus operating) cost with respect to uncertainty in some parameters. The decision vector in each stage is the amount of new capacity to build of each generation type. We first present a mathematical formulation of SGEP and then describe a reformulation of this problem as a Markov Decision Process.

The objective function (3.15) minimizes the sum of expected investment and operations costs over all investment periods $t \in T$, all generation technology types $g \in G$, and all scenarios $\sigma \in \Sigma$. The operations cost is summed over demand load blocks $l \in L$, with H_l hours in demand load block l. Scenarios are used to represent the uncertainty, and each scenario is associated with a probability $p(\sigma)$. The decision $y^g_{t\sigma}$ represents the new capacity added for each generation type g in scenario σ in investment period t. Hourly Demand within each year is represented by a load duration curve, in which electricity demand D_{lt} is specified for each demand block $l \in L$, and the demand blocks represent hours with different levels of Demand within the year. We assume a 2 % per annum growth rate in aggregate annual Demand. The long-term investment problem includes the dispatch sub-problem, choosing $x^g_{lt\sigma}$, the energy output from generation technology type g in scenario σ to meet the demand in block l in investment period t. The sum of generation over all sources must meet the Demand in every demand block for every scenario (3.16), and the output from each generation technology in every scenario for every demand block must be less than or equal to the available capacity in that investment period t (3.17).

$$\min_{x^g_{lt\sigma}, y^g_{t\sigma}} Z = \sum_{t=1}^{T}\sum_{g=1}^{G}\sum_{\sigma=1}^{\Sigma}\sum_{l=1}^{L}(OC^g_{t\sigma}x^g_{lt\sigma}H_l)p(\sigma) + \sum_{t=1}^{T}\sum_{g=1}^{G}\sum_{\sigma=1}^{\Sigma}(IC^g_{t\sigma}y^g_{t\sigma})p(\sigma) \quad g \in G,\ t \in T,\ \sigma \in \Sigma,\ l \in L \tag{3.15}$$

$$\sum_{g=1}^{G} x^g_{lt\sigma} = D_{lt\sigma},\ l \in L,\ t \in T,\ \sigma \in \Sigma \tag{3.16}$$

$$x^g_{lt\sigma} \le y^g_{t\sigma},\ t \in T,\ \sigma \in \Sigma,\ g \in G,\ l \in L \tag{3.17}$$

We reformulate the above SGEP as a Markov Decision Process with discrete time steps and continuous state and action spaces. The main components of the MDP correspond to the SGEP as follows:

- Stages: The stages in the MDP directly correspond to the investment periods t as formulated above. Each stage is a representative year, and the time interval between stage t and stage $t+1$ may be multiple years.
- State Space: The state vector s_t is defined as the current total capacity $z^g_{t\sigma}$ of each generation type g in stage t and scenario σ.
- Action Space: The action vector in each state is the amount of new capacity $y^g_{t\sigma}$ to be built for each generation type g for scenario σ in stage t.
- Exogenous Information: The exogenous uncertainty ω_{st} is the vector of uncertain operation and investment costs, which are elements of the cost parameters $OC^g_{t\sigma}$ and $IC^g_{t\sigma}$ in the objective function (3.15).
- Transition Function: Determines the new total capacity available in stage t, after the addition of new capacity. Mathematically it will be as in (3.18), with $z^g_{t\sigma}$ representing the generating capacity of type g available in scenario s, stage t.

$$z^g_{t\sigma} = z^g_{t-1\sigma} + y^g_{t\sigma},\ t \in T,\ \sigma \in \Sigma,\ g \in G \tag{3.18}$$

- Current Stage Cost: The current cost is the sum of the investment cost and the operating cost for stage t.
- Policy: The policy π maps the the state variable $s_t \equiv z^g_{t\sigma}$ to the optimal investment decision $a^*_{st} = y^g_{t\sigma}$. The policy provides a decision rule for the optimal capacity investment plan for every state in every stage.

The operations sub-problem in SGEP can include additional considerations and details such as including non-convex operational constraints of generators or representing transmission network flows and constraints, but at the expense of more computation time when obtaining each sample cost for state-action samples in each stage. Because our focus here is to compare the relative performance of alternative adaptive sampling approaches, we assume a simple form of the sub-problem, economic dispatch, which solves for the output of each generation type to meet Demand in each of 16 load blocks that approximate the Demand over an

entire year. The stylized experiment for this work is based on publicly available data for the Western Electricity Coordinating Council (WECC) [29, 60, 161]. The load duration curve consists of sixteen blocks, four demand levels from the hourly observations for each of the four seasons. To account for renewable energy generation, which are not considered for investment, net demand is calculated for 16 load blocks. The current capacity for each generator type is also considered and is shown in Table 3.2. We neglect consideration of transmission constraints and generator operating constraints. The methods shown here are applicable to more detailed representations of the dispatch sub-problem, and other operations models could be substituted.

We explore the investment strategy for four generation technologies: Natural Gas Combustion Turbine (GT), Combined Cycle Natural Gas Turbine (CCGT), Coal and Nuclear. The model has three investment stages, each representing one year, with stages assumed to be 20 years apart. We do not consider construction time lags for new capacity; new capacity built-in stage t is available for generation. The uncertain parameters are natural gas prices and future carbon prices. They are assumed to be uniformly distributed random variables between the upper and lower bounds provided in Table 3.1.

	Gas Price		**Carbon Price**	
Stage	**Lower Bound**	**Upper Bound**	**Lower Bound**	**Upper Bound**
Stage 1	3.2	3.2	50	50
Stage 2	3	7	0	100
Stage 3	3	11	100	300

Table 3.1: Uncertainty bounds for natural gas price and carbon price

Fuel Type	**Existing Capacity**
Natural Gas Combustion Turbine	9760
Combined Cycle Gas Turbine	12260
Coal	9260
Nuclear	8260

Table 3.2: Existing capacity for each generator type at $t = 0$.

3.7 Numerical Results

We compare the performance of the proposed QIS algorithm to two variants of epsilon greedy for the generation expansion problem described above. We assume a linear architecture for the approximate Q-function, but include bilinear and square basis functions to better capture the shape of the underlying value function. Choosing the best architecture for the approximate Q-function is an art in itself and is not the main focus of this paper; here, we focus on methods to sample actions from a continuous space for a given architecture. Because the problem has a continuous feasible action space, many exploration algorithms for RL (e.g., R-Max, E^3, Knowledge gradient) cannot be applied without some discretization of the decision space. We, therefore, leave comparisons to such methods to future work. The algorithms compared are:

1. **QIS**: The proposed algorithm described in Section 5.

2. **QIS-RE**: The proposed variant of QIS that reevaluates the approximate Q-values, $\hat{q}$, for all prior samples and updates $qMax_t$ and $qMin_t$ less frequently.

3. **Epsilon Greedy**: This algorithm prescribes the probability ϵ of randomly sampling the action; the optimal action for the current value function approximation, $\hat{q}$, for the sample state is selected with probability $1 - \epsilon$ [119, 162]. We present the results for values of ϵ from 0 to 1 in increments of 0.1, held constant over all iterations.

4. **Epsilon Decay**: This variant of epsilon greedy decreases the value of ϵ as the iterations progress. It, therefore, will have a higher proportion of exploration samples in early iterations and fewer exploration samples in later iterations. We assume that the value of ϵ is modified as in (3.19) with decay rate δ. The parameter δ is derived from specified initial and terminal values $\epsilon_{\text{Initial}}$ and ϵ_{Final} respectively, and the the maximum number of iterations K (3.20).

$$\epsilon_{\text{k}} = \epsilon_{\text{k-1}} * \delta \tag{3.19}$$

$$\epsilon_{\text{Final}} = \epsilon_{\text{Initial}} * \delta^{\text{K}} \tag{3.20}$$

5. **SP**: A deterministic equivalent implementation of the multi-stage stochastic optimization model (3.15-3.17) is used as a benchmark to validate the solutions of the various

Q-learning experiments. For the SP model, an exogenous scenario tree is constructed to represent the uncertainties in natural gas price and carbon price by normalizing the range of values to the interval [0,1] and discretizing in steps of 0.1 in each dimension. The investment decisions are continuous positive variables in the SP model.

The results for four algorithms (QIS, QIS-RE, Epsilon Greedy and Epsilon Decay) are presented below. Default values for algorithm parameters assume $M = 10$ samples per iteration, $K = 900$ total iterations, with $\hat{K}$=20 for QIS-RE. Q-function approximation, $\hat{q}_t$, coefficients are initialized to zero, and $qMin_t$ and $qMax_t$ are equal to 0 and 1, respectively, unless otherwise specified. All experiments assume a Temporal Difference learning rate $\lambda = 0.1$ to update the approximation coefficients θ (3.7). The models using QIS, QIS-RE, Epsilon Greedy and Epsilon Decay were implemented in Matlab. The non-linear program solved to obtain exploitation samples in Epsilon Greedy and Epsilon Decay were solved using the interior point algorithm with the fmincon function in Matlab. The simulation of the policy in the Q-learning algorithms to obtain the total cost was performed on the same scenario tree used in the deterministic equivalent implementation. The SP model and simulation of the Q learning policy are implemented in GAMS and solved by CPLEX 12.8.0. All experiments are performed on a high-performance computing cluster with 2.2 GHz Intel Xeon Processor, 24 cores, 128 GB RAM, and 40 Gbps Ethernet.

3.7.1 Stability of the optimal solution

An effective sampling algorithm within ADP should facilitate convergence to a close-to-optimal policy in a finite number of iterations. Moreover, the solution should be robust to the algorithmic parameters and consistent across multiple independent repetitions. We evaluate the consistency of the policy obtained from 10 replications of each algorithm. The QIS and QIS-RE algorithms obtain the same first stage policies for all replications, and exhibit variation in the optimal cost of less than 0.1% (Table 3.3). We also show the solution from the SP model for comparison. The computational effort for these models is described later in Section 3.7.4.

In Epsilon Greedy, the critical algorithmic parameter is ϵ, the probability of randomly sampling the next action, as opposed to choosing the optimal action with respect to the current approximate Q-function $\hat{q}$. In general, this value is chosen heuristically and is application-specific. Figure 3.2 shows the % difference in the expected optimal cost from the Epsilon Greedy version relative to the optimal benchmark cost from the Stochastic

Programming version across ten replications by simulating the optimal policies for epsilon values ranging from 0 to 1 in increments of 0.1. The choice of 0.6 for ϵ appears to provide the best results, with most repetitions diverging from the SP solution by less than 0.1% except for a few outliers. For values of ϵ between 0.0 and 0.3, the infrequent exploration of actions often leads to premature convergence to a sub-optimal solution. For values of ϵ above 0.7, the frequent exploration results in greater variance across different repetitions.

Epsilon Decay is a variant of Epsilon Greedy in which the exploration rate decreases as a function of the iteration count. The value of ϵ decreases at an exogenously specified rate; as more cumulative sample costs are observed, actions are randomly sampled less frequently. We parameterize the decay rate as in (3.20), where the user specifies the initial and final values of ϵ. We present the results from the Epsilon Decay version for all combinations of final values of ϵ {0,0.1,0.2,0.3} and initial values {0.7,0.8,0.9,1} (Figure 3.3). For most choices of the initial, final, and decay rate, the Epsilon Decay version can sometimes converge to a solution that is close to the optimal benchmark cost, but the variance across independent repetitions of the algorithm is large in most cases.

Algorithm	**% Nuclear**	**% Coal**	**% CCGT**	**% GT**	**Total Cost**	**% Difference from SP**
QIS	33.3	0.0	33.3	33.3	2.565E+11	0.05%
QIS-RE	33.3	0.0	33.3	33.3	2.565E+11	0.05%
Epsilon Greedy	50.0	0.0	0.0	50.0	2.575E+11	0.45%
Epsilon Decay	0.0	0.0	0.0	100.0	2.695E+11	5.12%
SP	32.0	0.0	32.0	36.0	2.564E+11	

Table 3.3: Comparison of the first stage policy obtained for QIS, QIS-RE, Epsilon Greedy with $\epsilon = 0.5$, Epsilon Decay with $\epsilon_{Initial} = 0.7$ and $\epsilon_{Final} = 0.2$ and SP for one of the replications of the algorithm. % Nuclear, % Coal, % CCGT and % GT represent the percentage of the first stage capacity decisions. Total cost is obtained by stochastic simulation of the policy for the same scenario tree as SP. % Difference from SP refers to the % difference in optimal expected cost relative to the SP solution from simulating the optimal policies obtained in QIS, QIS-RE, Epsilon Greedy, Epsilon Decay for one of the replications of the algorithm.

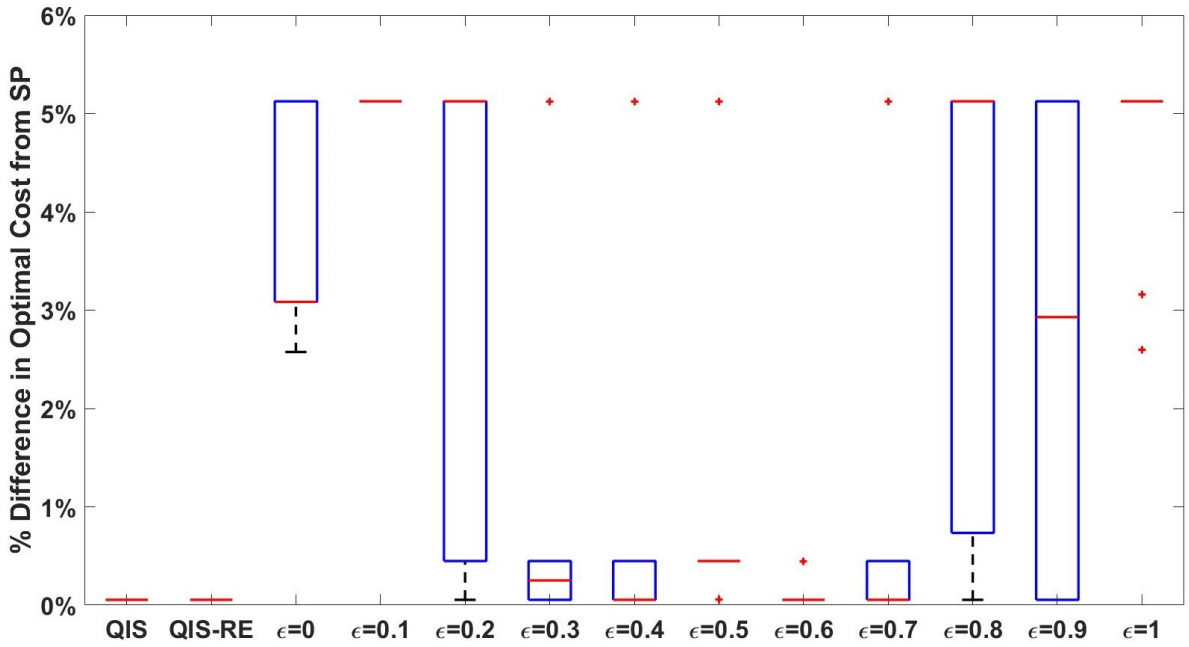


Figure 3.2: Comparison of % difference in optimal expected cost relative to the SP solution from simulating the optimal policies obtained in Epsilon Greedy for ϵ values from 0 (pure exploit) to 1 (pure explore) in increments of 0.1 for 10 replications with QIS and QIS-RE. Boxes enclose the 50% interval, midlines indicate the median value, whiskers indicate the 90% interval, and outlies are shown as '+'.

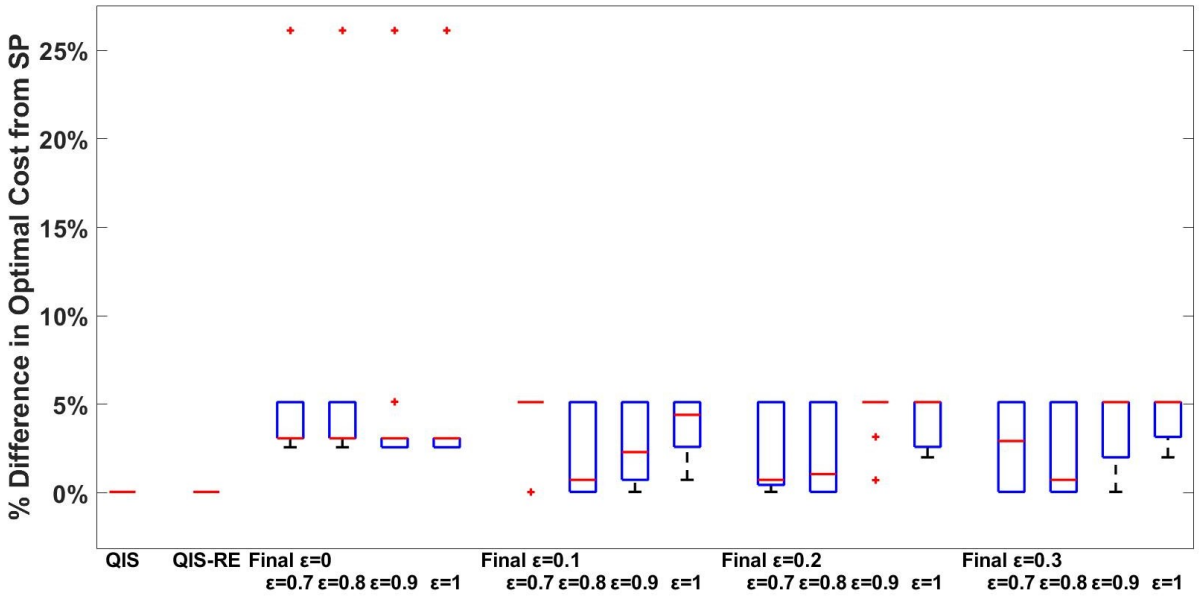


Figure 3.3: Comparison of % difference in optimal expected cost relative to the SP solution from simulating the optimal policies obtained in Epsilon Decay for different starting and final ϵ values with reductions according to Equation 3.20 for 10 replications with QIS and QIS-RE. Boxes enclose the 50% interval, midlines indicate the median value, whiskers indicate the 90% interval, and outlies are shown as '+'.

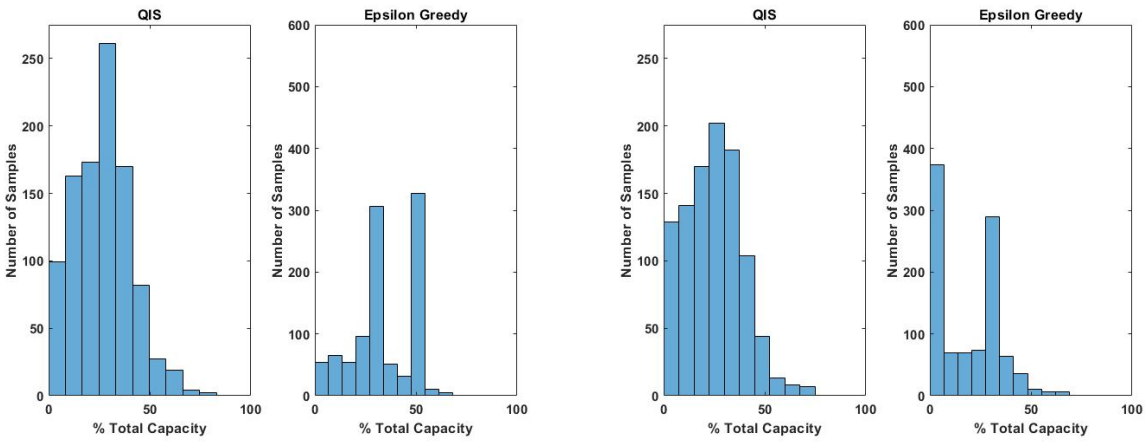
3.7.2 Balancing Exploration and Exploitation

Sampling strategies for ADP must balance exploration, sampling actions to improve the approximate value function with exploitation, sampling the optimal action to improve the estimate of its expected value. Epsilon Greedy achieves this balance with the parameter ϵ, the probability of sampling the following action randomly. QIS balances exploration and exploitation using the qRatio, which accepts proposed sample actions proportionally to their current estimated approximate Q-Value, $\hat{q}$. The QIS algorithm has a non-zero probability of sampling any feasible action, but in contrast to Epsilon Greedy, does not assume a uniform distribution over the feasible action space. By sampling proportional to the approximate Q-function, QIS will continue to explore more actions in the neighborhood of the current approximation optimum. Samples from Epsilon Greedy form a mixture distribution of uniformly random samples (explore) and a single value at the current optimum (exploit).

The difference between sampling distributions from QIS and Epsilon Greedy is illustrated in Figure 3.4, which shows the histogram of the final 1000 samples for new capacity of each generator technology in Stage 1 for one representative replication. Sample actions are

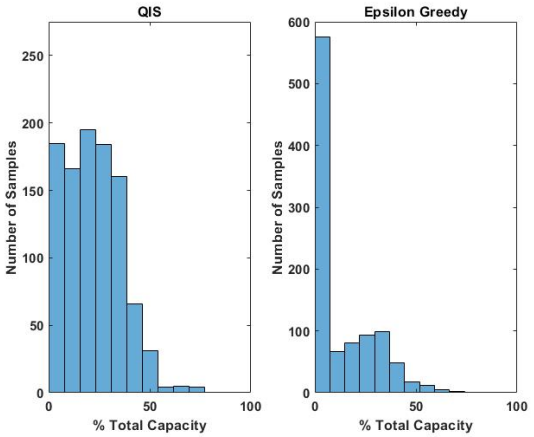
expressed as a percentage of the required new capacity for each generator technology; the percentage of total new capacity built must sum to 100%. The samples from QIS for new Nuclear, CCGT, and GT capacity additions more frequently fall close to the optimal decisions from the SP (Table 3.3). Samples of new coal capacity additions shift towards nearly equally spaced samples for capacity below 40%. Epsilon Greedy can sometimes result in frequently sampled actions from two distinct regions of the decision space (Figure 3.4). In this example, sampled actions oscillate between the optimal SP new capacity shares (Table 3.3) and a suboptimal policy of 50% Nuclear and 50% GT.

The use of temporal differencing (3.7) to update the approximation for continuous action spaces limits the incremental information value of repeatedly visiting the exact same action. Early iterations can often lead to an approximation with a local optimum. Sampling very suboptimal actions may not be sufficient to move the approximation away from the local optimum towards a global optimum. A sampling strategy that more frequently explores the immediate neighborhood of the optimal action, as in QIS, is better able to avoid getting stuck on a suboptimal solution. All replications performed have behavior similar to the one presented in Figure 3.4.

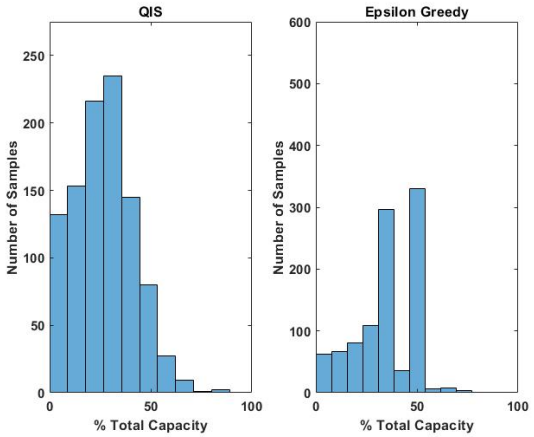


(a) GT Samples

(b) CCGT Samples



(c) Coal Samples



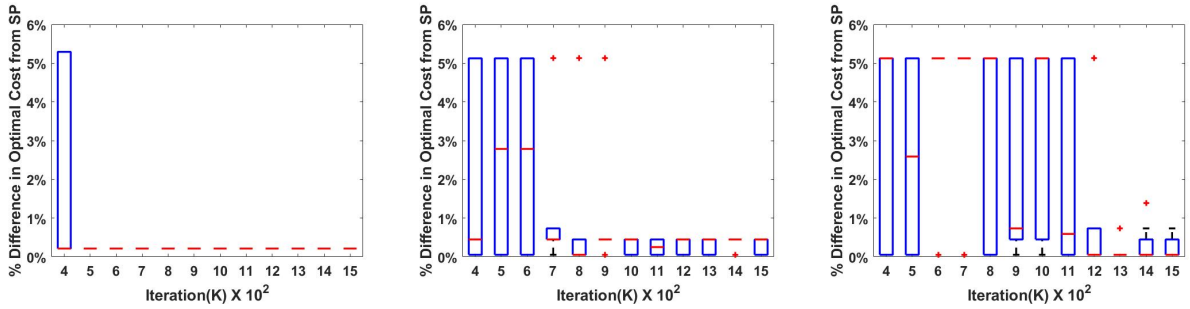
(d) Nuclear Samples

Figure 3.4: Final 1000 action samples for QIS and Epsilon Greedy as a percentage of Stage 1 new capacity for Natural Gas Combustion Turbine (3.4(a)), Combined Cycle Gas Turbine (3.4(b)), Coal (3.4(c)) and Nuclear (3.4(d)) technologies for one representative replication. Epsilon Decay and QIS-RE have not been shown because of similar behavior to Epsilon Greedy and QIS respectively. Note the change in the Y-axis scale between QIS and Epsilon Greedy.

3.7.3 Sensitivity to the number of iterations and the number of samples per iteration

This section compares the convergence rate across multiple replications to the optimal policy for QIS, QIS-RE, Epsilon Greedy, and Epsilon Decay. Figure 5 represents the %

difference in the estimated expected optimal cost of the policy, relative to the expected cost from the SP model, for ten replications after different numbers of iterations. We show results for one instance of Epsilon Greedy that assumes $\epsilon = 0.5$ (Fig. 3.5(b)), and one instance of Epsilon Decay that assumes $\epsilon_{Initial} = 0.7$ and $\epsilon_{Final} = 0.2$ (Fig. 3.5(c)). QIS and QIS-RE both consistently converge within 500 iterations to the optimal SP policy for all replications. The other two methods converge after 1000 iterations but exhibit higher variability across replications. For this example, Epsilon Greedy finds better solutions with less variability than Epsilon Decay. One possible reason for the poor solution quality from Epsilon Decay in these results is that it shifts to more exploitation samples before the approximation is sufficiently accurate. A fair comparison across these methods would also include the computational effort with iterations. For the sake of clarity, this is described in 3.7 in section 3.7.4.



(a) QIS & QIS-RE (b) Epsilon Greedy (c) Epsilon Decay with $\epsilon_{Initial}$=0.7 and ϵ_{Final}=0.2

Figure 3.5: Comparison of % difference in optimal expected cost relative to the SP solution from simulating the optimal policies obtained by QIS and QIS-RE (3.5(a)), Epsilon Greedy with $\epsilon = 0.5$ (3.5(b)) and Epsilon Decay (3.5(c)) with $\epsilon_{Initial} = 0.7$ and $\epsilon_{Final} = 0.2$. Shown are results for $K = 400$ to $K = 1500$ in increments of 100 for 10 replications. Boxes enclose the 50% interval, midlines indicate the median value, whiskers indicate the 90% interval, and outlies are shown as '+'.

We also explore the effect of different sample sizes within each iteration. We repeat the experiments with $M = 1, 2, 5, 10, 25, 50$ and 100, and present the % difference in the estimated expected optimal cost of the policy, relative to the optimal expected cost from the SP solution. The results are provided at two iteration counts, $K = 500$ and 900 in Table 3.4. For small sample sizes of 1, 2, and 5, the QIS results exhibit high variability at $K = 500$. However, all sample sizes converge to the same policy by $K = 900$ iterations, although some outliers are still observed for sample sizes of $M = 1$ and 2 for $K \geq 900$ (not shown). The QIS-RE algorithm exhibits behavior similar to that of QIS but with greater variability in expected

optimal costs for small samples sizes. Small sample sizes and infrequent re-evaluation of Q-values can result in poor estimates of the gradient and the temporal difference error in Equation (3.7). Larger per-iteration sample sizes enable better estimates of the gradient and error, which improves the solution quality and stability. The impact of the per-iteration sample size for the QIS algorithm is illustrated in Figure 3.6 in terms of the normalized Stage 1 optimal Q-function value; larger sample sizes result in a smoother evolution.

The estimated expected optimal costs for Epsilon Greedy with $\epsilon = 0.5$ have significant variability for all sample sizes at $K = 500$ iterations. After $K = 900$ iterations, the variance in the optimal cost across replications is smaller for sample sizes of 2 or more. However, the variability of solutions when using Epsilon Greedy is still greater than the solutions when using QIS or QIS-RE. Epsilon Decay's solutions exhibit the greatest variation across replications for all sample sizes relative to the other sampling methods. Further, Epsilon Decay sometimes finds an optimal cost close to the true optimum after $K = 500$ iterations but then diverges from this optimum with additional iterations. This suggests that as the frequency of exploitation increases, the model is choosing sub-optimal decisions because of local optima when solving for the optimal action or because of insufficient samples of other feasible actions.

		QIS			**QIS-RE**			**Epsilon Greedy**			**Epsilon Decay**		
K	**M**	**Min(%)**	**Med(%)**	**Max(%)**	**Min(%)**	**Med(%)**	**Max(%)**	**Min(%)**	**Med(%)**	**Max(%)**	**Min(%)**	**Med(%)**	**Max(%)**
500	**1**	0.05	0.05	5.12	0.05	0.05	5.12	0.05	5.12	5.12	0.05	0.8	5.12
	2	0.05	0.05	5.12	0.05	0.05	5.12	0.05	5.12	5.12	0.05	5.12	5.12
	5	0.05	0.05	5.12	0.05	0.05	5.12	0.05	5.12	5.12	0.05	5.12	5.12
	10	0.05	0.05	0.05	0.05	0.05	0.05	0.05	2.79	5.12	0.05	2.59	5.12
	25	0.05	0.05	0.05	0.05	0.05	0.05	0.05	2.59	5.12	0.05	0.05	5.12
	50	0.05	0.05	0.05	0.05	0.05	0.05	0.05	5.12	5.12	0.05	0.05	5.12
	100	0.05	0.05	0.05	0.05	0.05	0.05	0.05	5.12	5.12	0.05	0.05	5.12
900	**1**	0.05	0.05	0.05	0.05	0.05	5.12	0.45	0.69	5.12	0.09	0.45	5.12
	2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.45	0.45	0.05	0.45	5.12
	5	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.25	0.45	0.73	5.12	5.12
	10	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.45	5.12	0.05	0.73	5.12
	25	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	5.12	0.05	5.12	5.12
	50	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.45	0.05	2.93	5.12
	100	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.45	0.05	2.93	5.12

Table 3.4: Minimum, Median and Maximum % difference in the optimal expected cost relative to the SP solution from simulating the optimal policies obtained by QIS, QIS-RE, Epsilon Greedy with $\epsilon = 0.5$, and Epsilon Decay with $\epsilon_{Initial} = 0.7$ and $\epsilon_{Final} = 0.2$ for $K = 500$ and 900 with sample size $M = 1, 2, 5, 10, 25, 50$, and 100 for 10 replications.

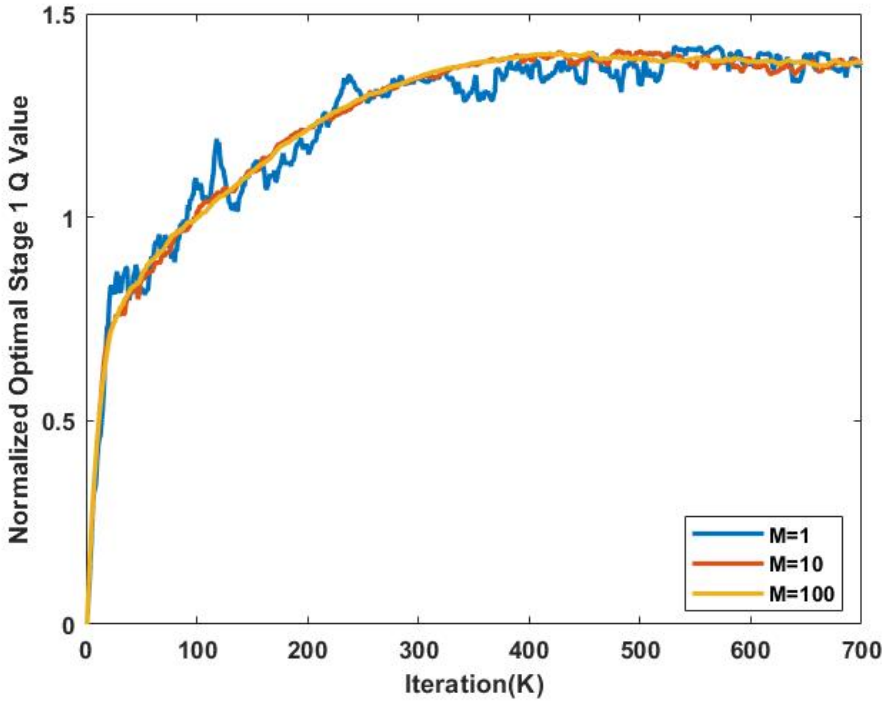


Figure 3.6: Normalized Optimal Stage 1 Q-value from the QIS algorithm for iterations $K = 1$ to 700 in increments of 1 for samples size $M = 1, 10$, and 100 for one representative replication.

3.7.4 Computational Effort

In addition to producing high-quality, low variance estimates of the optimal cost and policy, an effective sampling method should also have minimal computation time. Less computation time expended obtaining samples allows for larger problems to be solved. We compare the total computation time for all four sampling methods, averaged across ten replications, as a function of iteration count (Figure 3.7). The QIS algorithm requires the most computation time, followed by Epsilon Decay, Epsilon Greedy, and QIS-RE, which requires the least computation time. To understand the differences across methods, we distinguish the time required within three distinct algorithmic sub-tasks (Table 3.5): the "Sampling Time" is the time required to obtain the sample action for the given state; the "Evaluation Time" is the time required to recompute all approximate Q-values and update the estimated $qMax_t$ and $qMin_t$; and "Other Time" includes all other tasks including updating the approximation coefficients and parallel computing communication. The Evaluation Time is not required for Epsilon Greedy and Epsilon Decay because these methods do not update

estimates of $qMax_t$ and $qMin_t$. QIS requires the most computational effort because of the need to recompute the approximate Q-values for all samples from previous iterations after every iteration. Epsilon Greedy and Epsilon Decay require similar computation time; Epsilon Decay requires slightly more time because the increasing frequency of exploit samples requires more optimizations of the Q-function to obtain these sample actions. Both QIS and QIS-RE only evaluate the Q-function, but do not need to optimize it. QIS-RE requires the least total computation time among these algorithms because it does not have to optimize the value function, as in Epsilon Greedy and Epsilon Decay, and because it reduces the frequency of recomputing all Q-values, which takes the majority of the time for the QIS algorithm. In the example shown in Table 3.5, QIS-RE recomputes the approximate Q-values every 20 iterations ($\hat{K} = 20$), which significantly reduces the total computation time.

The QIS-RE does require an additional algorithmic parameter, $\hat{K}$, the frequency of recomputing $qMax_t$ and $qMin_t$. Table 3.6 presents the proportion of the replications that converge to the optimal Stage 1 policy (as in Table 3.3) for several values of $\hat{K}$, at different iteration counts, and for two different sample sizes. If QIS-RE never updates the Q-values, convergence to the optimal policy requires more iterations. Relatively infrequent reevaluation, such as every 50 or 100 iterations, has nearly the same performance as more frequent reevaluation, but requires less computation time. Furthermore, larger sample sizes ($M = 10$) provide better solution consistency independent of the frequency of reevaluation. Even the No Update version converges to the optimal policy with less variance than either Epsilon Greedy or Epsilon Decay if larger per-iteration sample sizes are used. We recommend updating the estimates of $qMax_t$ and $qMin_t$, even if infrequently because it accelerates convergence.

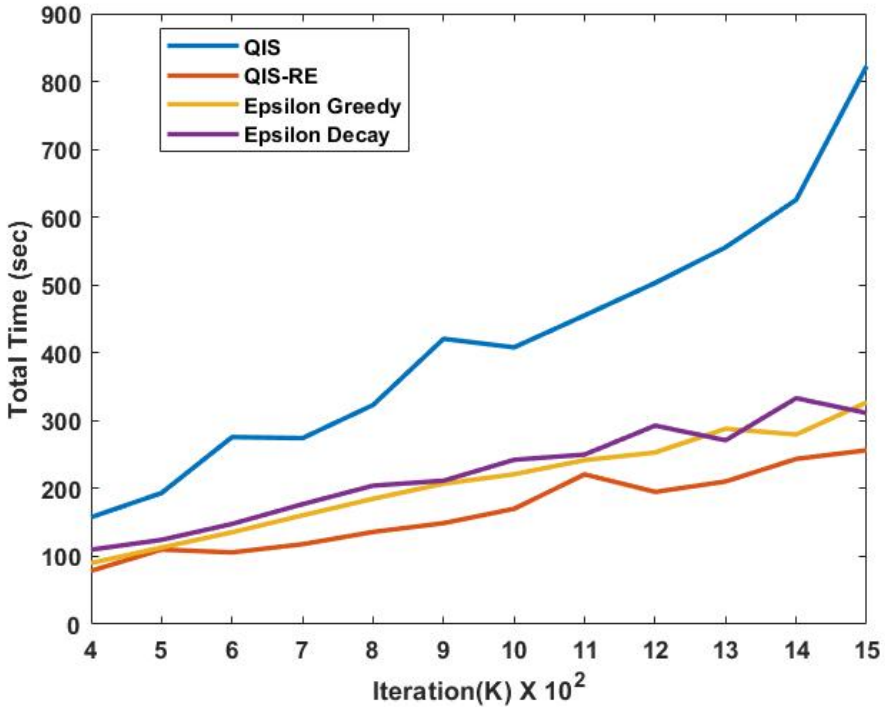


Figure 3.7: Total computation time (sec), averaged across 10 replications, for QIS, QIS-RE with $\hat{K} = 20$, Epsilon Greedy with $\epsilon = 0.5$ and Epsilon Decay with $\epsilon_{Initial} = 0.7$ and $\epsilon_{Final} = 0.2$ for iterations $K = 400$ to 1500 in increments of 100 with $M = 10$.

Algorithm	**Sampling Time (sec)**	**Evaluation Time (sec)**	**Other Time (sec)**	**Total Time**
QIS	58.7	267.7	94.4	420.8
QIS-RE	53.1	12.8	83.1	149.0
Epsilon Greedy	102.9	-	104.3	207.2
Epsilon Decay	113.5	-	98.1	211.6

Table 3.5: Computation time (sec) for Sampling Time, Evaluation Time, Other Time, and Total Time, averaged across 10 replications of QIS, QIS-RE with $\hat{K} = 20$, Epsilon Greedy with $\epsilon = 0.5$ and Epsilon Decay with $\epsilon_{Initial} = 0.7$ and $\epsilon_{Final} = 0.2$. Results are for $K = 900$ iterations and $M = 10$ samples.

	M=1				M=10			
$\hat{K}$ vs K	**K=400**	**K=500**	**K=600**	**K=700**	**K=400**	**K=500**	**K=600**	**K=700**
$\hat{K}$=1	40%	70%	90%	90%	60%	100%	100%	100%
$\hat{K}$=10	50%	100%	100%	90%	50%	100%	100%	100%
$\hat{K}$=20	60%	80%	90%	100%	60%	100%	100%	100%
$\hat{K}$=50	80%	100%	90%	90%	50%	100%	100%	100%
$\hat{K}$=100	90%	60%	90%	100%	70%	100%	100%	100%
No Update	60%	60%	60%	70%	30%	90%	90%	100%

Table 3.6: Frequency within 10 replications that the optimal Stage 1 policy from QIS-RE is the policy shown in Table 3.3. Results show current optimal policy for different reevaluation frequencies $\hat{K}$, for sample sizes of $M = 1$ and 10, and for iteration counts $K = 400, 500, 600,$ and 700.

3.8 Conclusions

We have presented the QIS algorithm, a novel scheme for sampling continuous action spaces in Approximate Dynamic Programming formulations. The approach uses importance sampling to sample actions proportional to the current state-action value function approximation. In contrast to methods that select the current optimal action or sample uniformly across the feasible action space, we have demonstrated superior solution quality and lower variance across replications when using QIS. One advantage of this algorithm is the absence of tuning parameters when sampling actions because it relies only on the approximate value function. Another advantage is that it avoids solving a (possibly nonlinear) optimization to select the optimal action within the sampling step. One disadvantage of the QIS algorithm is the additional computational effort required to reevaluate prior samples after every iteration. We have also proposed a variant on the algorithm, QIS-RE, that requires less computational effort and achieves the same solution quality and stability across replications as QIS. QIS-RE does require an additional parameter for the frequency of reevaluations, although our results show that convergence is relatively insensitive to different assumptions.

Both the QIS and the QIS-RE algorithms can scale to larger multidimensional continuous state and action spaces, and they can be applied to any value function approximation architecture. The action space in the case study presented was continuous, but this method is likely applicable to discrete decision spaces as well; this is left for future work. A potential

improvement to the algorithm, not explored here, would be to gradually increase the relative density of sampled actions very close to the optimal action as iterations progress by means of an exponential parameter applied to the qRatio. This parameter would have low values in the initial iterations to promote exploration (more samples accepted farther from the optimal action) and higher values later to increase exploitation (most accepted samples very close to optimal action). The approximation architecture used in this paper for the Q-function was linear, but we expect this sampling scheme to be applicable to non-linear function approximation architectures, such as neural networks. The QIS sampling approach could be especially advantageous for continuous decision spaces when Epsilon Greedy methods are difficult to apply within a neural network based approximation. These areas of exploration are left for future work.

Much of the work on the application presented here, generation capacity expansion for electric power systems, has relied on methods from within the stochastic programming paradigm. As multi-stage stochastic problems grow large, the decomposition methods within that paradigm often rely on dual variables and other assumptions about the studied system's functional form (e.g., convexity). Many of the applied questions in the area of energy transition and achievement of a low carbon energy system will require larger multi-stage problems that cannot omit critical non-convex features of the power system, which can limit the applicability of traditional decomposition schemes. Although the case study in this work was intentionally simplified to provide an optimal benchmark, the method here offers new avenues for solving large multi-stage stochastic problems with non-convex features. An exploration of these problems is another area ripe for future work.

Chapter 4
A comparison of linear multi-stage stochastic optimization methods

4.1 Introduction

Multistage stochastic optimization refers to the class of problems which facilitate decision-making under uncertainty present in the system with three or more stages. Traditionally, these problems were solved with dynamic programming using Bellman's equation, which has the advantage of not relying on the problem structure and can be used with integer and non-linear constraints [163]. However, with the advancement of linear programming solvers, dynamic programming became less popular because of the requirement to completely enumerate or discretize each stage's decision and uncertainty space. Efficient linear programming solvers led to the reformulation of these problems as one large linear program with an explicit scenario tree called the deterministic equivalent [164, 165].

Even with the advancement of linear programming solution techniques, deterministic equivalent formulations can be very large for specific problems with prohibitively long solution times. An alternative is the Nested Benders decomposition [166] technique for multistage stochastic problems, an extension of the L-shaped method [108] in 2 stage stochastic programs. Nested Benders is an iterative method that utilizes the dual information from the next stage to construct a hyperplane to approximate the expected cost of the next stage. The Nested Benders algorithm uses an explicit scenario tree and solves linear programs for every scenario in each stage, and the dual information is passed to the predecessor node in the previous stage. This enables stage-wise decomposition, which reduces the computational effort considerably compared with solving one large deterministic equivalent problem for all stages at once. Another class of methods is scenario decomposition methods such as

Progressive Hedging [167], an extension of the ideas in Lagrangian Relaxation [168] and the proximal point algorithm [169]. This method relaxes the non-anticipativity constraints so that each scenario path in the optimization problem can be solved independently. In addition to the Lagrangian parameter, an additional penalty term is added to the objective function that penalizes the variation of decisions within each scenario sub-problem from its average at a particular node. This accelerates the convergence as all scenarios coming out of a node should have the same decisions because of the non-anticipativity constraints. The scenario decomposition allows individual scenario sub-problems to be solved in parallel, reducing the computational effort.

Both Nested Benders and Progressive Hedging require the discretization of the uncertainty space. Discretization means sampling a finite number of scenarios from a continuous distribution at each stage and generating a scenario tree from these samples. A scenario tree that sufficiently represents the joint continuous distribution of the uncertainty space will be large and time-consuming in many practical situations. Further, solving a large scenario tree with progressive hedging or nested benders can be computationally prohibitive. An alternative to scenario tree approaches is to randomly sample from the uncertainty space and use Dynamic Programming methods such as Stochastic Dual Dynamic Programming(SDDP) [170], and Approximate Dynamic Programming(ADP) [114], or Reinforcement Learning(RL) [115] from the computer science literature. SDDP and ADP were developed on Bellman's equation in dynamic programming, but do not enumerate the decision and uncertainty space, thus partially alleviating the Curse of Dimensionality. These methods can also take advantage of high-performance computing environments, as they are easily parallelizable.

Stochastic Dual Dynamic Programming is a technique for solving multistage stochastic optimization problems that approximates the recourse function using cutting planes based on the dual information from the subsequent stage, similar to nested benders. Increased efficiency is achieved by assuming or reformulating the problem to be stage-wise independent, allowing the cuts to be shared across scenarios. This avoids the discretization of states required in Nested Benders Decomposition. Stage-wise, independence is a strong assumption and may not be satisfied for some applications. One example is representing the stochastic process of future carbon tax evolution, in which the future carbon tax is strongly dependent on current policies. Another drawback of SDDP is the increase in computational effort as the number of decision stages grows [171].

Approximate Dynamic Programming is a Monte-Carlo-based sampling method that can find good approximate solutions to the original problem by sampling decisions and

uncertainties and using a low dimensional function to approximate the value function. Approximate Dynamic programming takes advantage of stage-wise decomposition as well as problem decomposition. It decomposes the problem by sampling decisions and uncertainties at each stage and then evaluating the cost for that decision and uncertainty. This contrasts with SDDP, in which uncertainties are sampled, but decisions are obtained by solving a linear program. The solution of a linear program for every subproblem can become computationally expensive as the number of uncertainties grows or when the subproblem itself is computationally expensive. For example, in generation expansion planning, solving a linear program with investments and dispatch of generators will be more time-consuming than if investments are sampled, and only dispatch costs are calculated for the particular investment.

These three methods, PH, SDDP, and ADP, have all been widely applied to various settings in the literature, including hydro-thermal scheduling [172–174], surgical planning [175, 176], generator expansion planning [35, 170, 177, 178] and revenue management [110, 179]. Many studies have focused on improving one of these methods to obtain better solution quality and/or to reduce the computational effort [180–189]. However, to the author's knowledge, a comparison of these methods in terms of solution quality and computational effort has not been performed in the literature and is a primary contribution of this paper. In this paper, we apply these three methods to the multistage stochastic generator expansion planning (GEP) problem. The GEP problem chooses a set of generation investments at each stage under uncertainty, subject to meeting the electricity demand and other system requirements. All three methods have been applied to GEP problems [35, 170, 177, 178], and GEP provides a useful case study with multiple investment decisions under multi-dimensional continuous uncertainty. GEP also provides a setting to compare the performance across methods as a function of problem size. When solving GEP, a policy is obtained by solving the investment and dispatch decision at each stage. When renewable generation capacities were small, relatively few representative hours were required to estimate the operating costs of candidate investment plans. However, as explained in [45, 49], it is necessary to use hourly resolution in the dispatch decisions with higher levels of renewable generation. This study systematically compares the three methods applied to GEP with hourly resolution in the dispatch subproblems, thereby comparing them as the problem size increases.

The contributions of this work can be summarized as follows:

- Demonstration of the stage-wise and problem decomposition framework of approximate dynamic programming can provide good approximate solutions with reasonable

computational effort.

- A comprehensive comparison of Progressive Hedging, Stochastic Dual Dynamic Programming, and Approximate Dynamic Programming in terms of solution quality and computational effort across multiple dimensions of decision, uncertainties, stages, and problem size. In addition to the empirical comparisons of the three algorithms, we also explain the abstract advantages and disadvantages of the three methods. We also provide recommendations on the problem classes that each of the methods can be applied to, considering solution quality, computational effort and constraints of the decision maker.

- Application of all three methods to multistage generator expansion planning with hourly operations resolution in the sub-problem. To be the best of the author's knowledge, this is the first work that considers the entire year (8736 hours) in the dispatch problem within a multistage generator expansion planning problem.

The rest of the chapter proceeds as follows: Section 4.2 defines the problem followed by a brief overview of the three methods in section 4.3. Section 4.4 introduces the application, generation expansion planning, and section 4.5 presents the numerical results for this application. Section 4.6 gives an abstract comparison of the three methods. Section 4.7 gives recommendations on which of the three methods are better suited for different problem classes with a concluding discussion in section 4.8.

4.2 Problem Formulation

We consider multi-stage stochastic linear programs of the following form

$$\min \mathbf{Z} = c_1 x_1 + \mathrm{E}_{\zeta_2}[\min c_2(\omega_2)x_2(\omega_2) + ...[\mathrm{E}_{\zeta_\mathrm{T}} \min c_T(\omega_T)x_T(\omega_T)]....] \tag{4.1}$$

subject to

$$A_1 x_1 = b_1 \tag{4.2}$$

$$A_t(\omega_t)x_t(\omega_t) = b_t(\omega_t) - W_t(\omega_t)\hat{x}_{t-1}(\acute{\omega_t}), \qquad t = 2.....T \tag{4.3}$$

$$x_t(\omega_t) \geq 0 \tag{4.4}$$

$$x_t(\omega_t) = x_t(\acute{\omega}_t) \qquad t = 1....T \tag{4.5}$$

Equation 4.1 is the objective function, which minimizes the total expected cost of the system across all periods. Constraint 4.2 and 4.3 can be viewed as structural constraints of the system. For example, in an inventory problem, this would be to meet the demand at each scenario, or for capacity expansion, it would be to have the generation lower than the current capacity. Constraint 4.4 represents the non-negativity constraints, and Constraint 4.5 represents the non-anticipativity constraints on the system. We assume without loss of generality that $c_t \in R^{n_t}$, $A_t \in R^{m_t \times n_t}$, $W_t \in R^{m_{t-1} \times n_t}$, $B_t \in R^{m_t \times 1}$. The problem is deterministic for $T = 1$ and can be random for stages thereafter. The random variable ω_t can be a D dimensional random vector with positive upper and lower bounds. We refer the interested reader to [107] for a detailed overview of multi-stage linear stochastic optimization.

4.3 Methods

This section provides an overview of the decomposition methods that are compared in this paper: Progressive Hedging, Stochastic Dual Dynamic programming, and Approximate Dynamic Programming.

4.3.1 Progressive Hedging

Progressive hedging is a scenario-based decomposition technique that relaxes the non-anticipativity constraints by penalizing them in the objective function. It was developed in the late 1980s by Rockafellar and Wets [109] based on the proximal point algorithm in non-linear programming and augmented Lagrangian decomposition method. Before proceeding with the algorithm, we make some preliminaries on the notation: $\lambda^k_{\omega,t}$ is the Lagrangian multiplier associated with non-anticipativity constraint on $x_{\omega,t}$ and k is the iteration counter of the algorithm. We also define the probability-weighted average decision for every scenario ω at time t for iteration k using Equation 4.6. This equation gives the projection of the solutions to all scenarios to a single value which should be the same across all scenarios according to the non-anticipativity constraints in Equation 4.5.

$$x^{\hat{v}}_{\omega,t} = \frac{\sum_{\omega\in\Omega} p_\omega x^v_{\omega,t}}{\sum_{\omega\in\Omega} p_\omega} \tag{4.6}$$

We then define a saddle point function for every scenario $\omega \in \Omega$ using Equation 4.7. The two penalty terms in the equation correspond to the Lagrangian multiplier and a penalty to penalize any solution that deviates from the average since all scenarios should have the same decision based on the non-anticipativity constraints. The coefficient ρ determines the penalty amount for the deviation from the average.

$$\min_x \sum_{\omega\in\Omega} p_\omega \sum_{t=1}^{T} c_t(\omega_t) + \lambda^k_{\omega,t} x_{\omega,t} + \rho \left\| x^v_{\omega,t} - x^{\hat{v}}_{\omega,t} \right\|^2 \tag{4.7}$$

Algorithm 4.2 outlines the Progressive Hedging algorithm. Step 1 is to initialize the model parameters, ρ to penalize the deviation from average for each scenario sub-problem, and ϵ is the parameter for stopping criteria of the algorithm. Steps 2 and 3 are to obtain the decisions for every scenario without any penalty terms. Step 4 calculates the weighted average value of the decisions and then evaluates the stopping criteria in 5. This criterion assesses whether all scenario subproblems are within an acceptable threshold of the average, and if they are, the algorithm terminates. If they are not within the ϵ tolerance, then the Lagrangian parameter is updated in Step 6, and Step 7 evaluates the objective function with penalty terms to obtain a new set of decisions for every scenario. Step 8 obtains the weighted average value of the decisions and the algorithm continues until the stopping criteria is met.

Algorithm 4.2 Progressive Hedging

1: Set the value for ρ and ϵ and $k = 1$.
2: **for** w $\in \omega$
3: Solve Equation 3.15 s.t. Equation 4.4 to obtain $x^v_{\omega,t}$
4: **end for**
5: Calculate $x^{\hat{v}}_{\omega,t}$ using Equation 4.6
6: **while** $\left\| x^v_{\omega,t} - x^{\hat{v}}_{\omega,t} \right\| < \epsilon \forall \ \ \omega \ \ \in \Omega$
7: Update Lagrangian parameter as Equation 4.8

$$\lambda^{k+1}_{\omega,t} = \lambda^k_{\omega,t} + \rho(x^v_{\omega,t} - x^{\hat{v}}_{\omega,t}) \tag{4.8}$$

8: Solve Equation 4.7 s.t. Equation 4.4 to obtain $x^v_{\omega,t}$.
9: Calculate $x^{\hat{v}}_{\omega,t}$ using Equation 4.6
10: **end while**

Progressive Hedging algorithm is very popular, and the main appeal of this algorithm lies

with its easy implementation and scenario parallelization across multiple processors during the evaluation of the relaxed objective for every scenario as in Equation 4.7. One of the drawbacks of the algorithm is that it requires an efficient and stable non-linear/quadratic programming solver for the evaluation of Equation 4.7 which could be a bottleneck if the sub-problem is very large by itself.

4.3.2 Approximate Dynamic Programming

Approximate Dynamic Programming is a Monte-Carlo sampling based optimization technique that is built on the fundamentals of dynamic programming invented by Bellman. The Bellman based dynamic programming method requires discretization of the entire decision and uncertainty space and suffers from the curse of dimensionality. ADP or Reinforcement Learning in the computer science literature provides an approach to give good approximate solutions to multi-dimensional stochastic optimization problems which are more computational tractable. We first re-frame the original multi-stage stochastic optimization problem in Equation 3.15 to 4.5 in terms of the Bellman equation as follows:

$$v_t^*(s_t) = \min_{a_t} E_{P_{s_t,s_{t+1}}} \left\{ r_t(s_t, a_t) + \gamma v_{t+1}^*(s_{t+1}) \right\} \quad \text{for all} \quad s \in S, t \in T \tag{4.9}$$

where t is the time period/stage, s_t represent the vector that makes the process Markovian, a_t is the decision set of all possible actions, $P_{s_t,s_{t+1}}$ representing the probability distribution of the state transition function, $r_t(s_t, a_t)$ is the reward/cost for choosing action a_t in state s_t and $v_t^*(s_t)$ is the value function associated with the optimal policy π. The solution to Equation 4.9 can be obtained by doing backward induction on each stage and repeating the process till we reach t=1. This approach becomes computationally intractable as the number of stages, uncertainties, or decisions increase [111, 112].

Approximate algorithms instead focus on developing a lower order approximation for the value function $v_t(s_t)$ and sampling state-action pairs (s_t, a_t), observing the reward r_t and updating the lower order approximation using stochastic approximation. The lower order approximation is obtained by using a set of features, and an extensive set of features can be found in Sutton and Barto. The simplest approximation is a linear function of the basis function with weights θ, which solves for the approximate value function $\hat{v}_t$ as shown in Equation 4.10. It can also solve for the value function that depends both on state and action, commonly called the Q function as in Equation 4.11.

$$\hat{v}_t(s,\theta) \approx v_t(s) \quad \text{for all} \quad t \in T \tag{4.10}$$

$$\hat{q}_t(s,a,\theta) \approx q_t(s,a) \quad \text{for all} \quad t \in T \tag{4.11}$$

The approximation is updated iteratively by sampling specific state-action pairs with temporal difference learning as in Equation (4.12),

$$\hat{q}_t(s,a,\theta) \leftarrow (1-\lambda)\hat{q}_t(s,a,\theta) + \lambda(r_t(s_t,a_t) + \min_{a_{t+1}} \hat{q}_{t+1}(s_{t+1},a_{t+1},\theta)) \quad \text{for all} \quad t \in T \tag{4.12}$$

The outline of the algorithm is given in Algorithm 4.3. The algorithm samples decisions for each iteration k for each stage t. Step 1 of the algorithm initializes the parametric function approximation for the Q function as $\hat{q}_t$ and the initial state of the system as s_0. For steps 3-19, the algorithm has a forward pass where it samples a decision using the sampling policy and calculates the reward for this state action pair as $r_t^m(s_t^m, a_t^m)$. This is continued for M samples at each stage t. Steps 20-23 are the backward pass of the algorithm where the parametric function approximation is updated using temporal difference learning using equation 4.12 with the samples obtained in the forward pass. The entire algorithm with forward and backward passes is performed until the maximum number of iterations K is reached. One of the aspects of approximation algorithms that sample decision is an efficient sampling policy that balances the trade-off between exploring enough samples to improve the estimate of the approximation and exploiting the current estimate to minimize cost. In this paper, we use a modified framework of the QIS algorithm [190] to obtain decisions within a Q-learning framework.

The approximate dynamic programming method provides a unique problem + stage-wise decomposition framework when applied to multi-stage generation expansion planning. The stages are solved separately resulting in stage-wise decomposition. The generation expansion planning problem has two decisions at each stage: a) Investment decisions which decide the new capacity of each technology to be added to the system, and b) the dispatch decisions which determine the generation of each technology for the hours represented in the model. The actions a_t^m obtained from the sampling policy in ADP determines the investment decisions only. Once these investment decisions are sampled, r_t^m determines the cost for this investment. The cost calculation is a linear optimization problem which also decides the dispatch of each technology. The problem decomposition framework in ADP reduces the size of the optimization problem and is not observed in the other decomposition methods (SDDP and

PH), where investment and dispatch decisions are solved together in one optimization model.

Algorithm 4.3 Approximate Dynamic Programming

Initialize $\hat{q}_t$ for $t = 1..T$ and s_0
for $k = 1..K$
    Begin Forward Pass
    for $t = 1..T$
        if $t = 1$
            for $m = 1..M$
                $s_t^m = s_0$
            end for
        else
            for $m = 1..M$
                Sample $\omega_t^m \in \Omega_t$
                $s_t^m = [a_{t-1}^m \;\; \omega_t^m \,]$
            end for
        end if
        for $m = 1..M$
            Propose an action a_t^m using a sampling policy.
            Calculate $r_t^m(s_t^m, a_t^m)$.
        end for
    end for
    End Forward Pass
    Begin Backward Pass
    for $t = T..1$
        Update $\hat{q}_t$ using M samples in forward pass using equation 4.12.
    end for
    End Backward Pass
end for

4.3.3 Stochastic Dual Dynamic Programming

Stochastic Dual Dynamic Programming is a technique developed by Perreira and Pinto for solving large-scale multistage stochastic linear programs based on the framework of Dynamic Programming invented by Bellman. SDDP develops a piece-wise outer approximation of the future cost defined by cutting planes by solving linear programs at each stage and assuming stage-wise independence instead of completely discretizing the uncertainty and action space as in dynamic programming. These cuts are similar to the Nested Benders decomposition algorithm, but the cuts can be shared across different scenarios, thus collapsing the scenario tree, unlike Nested Benders decomposition.

By assuming stagewise independence of the uncertainties ω_t, the stochastic program defined by Equations 4.1 to 4.5 can be reformulated as below

$$min\ \mathbf{Z} = c_1 x_1 + \mathrm{E}(\mathrm{Q}_2(\mathrm{x}_1, \omega_2)) \tag{4.13}$$

subject to

$$A_1 x_1 = b_1 \tag{4.14}$$

$$x_1 \geq 0 \tag{4.15}$$

The second stage cost in this formulation is defined by $Q_2(x_1, \omega_2)$, which will be the expected cost given a decision x_1 and uncertainty ω_2 belonging to a D dimensional space. The second and future stage problems to be solved are as follows:

$$Q_t(x_{t-1}, \omega_t) = \min c_t x_t + \mathrm{E}(\mathrm{Q}_{\mathrm{t}+1}(\mathrm{x}_\mathrm{t}, \omega_{\mathrm{t}+1})) \tag{4.16}$$

subject to

$$A_t(\omega_t)x_t(\omega_t) = b_t(\omega_t) - W_t(\omega_t)x_{t-1}(\omega_t) \qquad [\pi_{\omega_t}] \tag{4.17}$$

$$x_t \geq 0 \tag{4.18}$$

We assume again without loss of generality that $c_t \in R^{n_t}$, $A_t \in R^{m_t \times n_t}$, $W_t \in R^{m_{t-1} \times n_t}$, $B_t \in R_{m_t} \times 1$. π_{ω_t} is the dual vectors associated with the constraints at each stage. These dual vectors are used to construct the outer piece-wise linear approximation using Benders cuts as in Equation 4.19, where $\bar{\pi}^k_{t+1} = \mathrm{E}[\pi^{\mathrm{k}}_{\mathrm{t}+1,\mathrm{j}}(\omega_{\mathrm{t}+1})]$ can be understood as the expectation of all the dual vectors in an iteration for all cuts k=1..K. α_{t+1} is the under approximation of the expected cost $\mathrm{E}(\mathrm{Q}^{\mathrm{k}}_{\mathrm{t}+1,\mathrm{j}}(\mathrm{x}_\mathrm{t}, \omega_{\mathrm{t}+1}))$ in stage t+1 and is replaced in Equation 4.13 and 4.16. The t stage formulation of the algorithm is given by Equation 4.20 to Equation 4.23 with no uncertainty in stage 1 can be represented by removing ω.

$$\alpha_{t+1} \geq (Q_{t+1}(x^k_t, \omega_{t+1}) + \bar{\pi}^k_{t+1} W_{t+1}(x_t - x^k_t) \quad \text{for cuts k=1...K} \tag{4.19}$$

$$Q_t(x_{t-1}, \omega_t) = \min c_t x_t + \alpha_{t+1} \tag{4.20}$$

subject to

$$A_t(\omega_t)x_t(\omega_t) = b_t(\omega_t) - W_t(\omega_t)x_{t-1}(\omega_t) \qquad [\pi_{\omega_t}] \tag{4.21}$$

$$x_t \geq 0 \tag{4.22}$$

$$\alpha_{t+1} \geq (Q_{t+1}(x_t^k, \omega_{t+1}) + \bar{\pi}_{t+1}^k W_{t+1}(x_t - x_t^k) \quad \text{for cuts k=1...K} \tag{4.23}$$

The cuts added to the problem to obtain the optimal policy are through a series of forward and backward passes of the algorithm. N scenarios are obtained in every forward pass, and the decision and cost at every node are calculated, which gives the sample upper bound for the system. In the backward pass, the system adds N cuts to the problem starting from the last stage. Every node in stage t solves all sampled scenarios of the next stage to add the Benders cuts. As it solves all the scenarios coming out of stage t, it takes the expected value of the objective function and the dual vectors over the uncertainty realization to obtain the cuts. The outline of the SDDP algorithm is given in Algorithm 4.4.

4.4 Application: Multi-Stage Generation Expansion Planning

We compare the three methods with an application to a multistage stochastic electricity generation expansion problem. Stochastic Generation Expansion Planning (SGEP) is a long-term investment problem that solves the optimal new capacity investments in each stage that minimize the expected total (capital plus operating) cost with respect to uncertainty in some parameters. The decision vector in each stage is the amount of new capacity to build of each generation type.

The objective function (4.24) minimizes the sum of expected investment and operation costs over all investment periods $t \in T$, all generation technology types $g \in G$, and all scenarios $\sigma \in \Sigma$. The operation cost is summed over demand load blocks $l \in L$, with H_l hours in demand load block l. Scenarios are used to represent the uncertainty, and each scenario is associated with a probability $p(\sigma)$. The decision $y_{t\sigma}^g$ represents the new capacity added for each generation type g in scenario σ in investment period t. Hourly demand within each year is represented by a load duration curve, in which the electricity demand D_{lt} is specified

Algorithm 4.4 : Stochastic Dual Dynamic Programming

Initialize an upper and lower bound equal to ∞ and -∞ and iteration counter i
while some stopping criteria is met
    Sample M scenarios $\omega_t^m \in \Omega_t$
    Forward Pass
    for k=1....M
        for t=1...T
            $\bar{x}_t^k \leftarrow$ Solve Equation 4.20 to 4.23 for every stage t for every scenario ω_t^k
        end for
        $z_k = \sum_{t=1}^{T} c_t x_t$
    end for
    $\bar{z} = \frac{1}{M} \sum_{k=1}^{M} z_k$
    Calculate upper bound: $UB = \bar{z} + t_\alpha \frac{\sigma}{\sqrt{M}}$
    Backward Pass
    for t=T...2
        for k=1..M
            for j=1..N_t
                $[Q_{tj}^k(\bar{x}_t^k), \pi_{tj}^k]$, $\leftarrow$ Solve Equation 4.20 to 4.23 for every stage t for every scenario ω_t^k and decision $\bar{x}_{t-1}^k$.
            end for
            $Q_t(x_t^k, \omega_t) = \frac{1}{N_t} \sum_{j=1}^{j=N_t} Q_{tj}^k(\bar{x}_t^k)$; $\pi_t^k = \frac{1}{N_t} \sum_{j=1}^{j=N_t} \pi_{tj}^k$
            Use $Q_t(x_t^k, \omega_t)$ and π_t^k in equation 4.23.
        end for
    end for
    Calculate the lower bound: Solve Equation 4.13 to 4.15 and the objective value equals the lower bound.
end while

for each demand block $l \in L$, and the demand blocks represent hours with different levels of demand within the year. We assume a two % per annum growth rate in aggregate annual demand. The long-term investment problem includes the dispatch sub-problem, choosing $x_{lt\sigma}^g$, the energy output from generation technology type g in scenario σ to meet the demand in block l in investment period t. The sum of generation over all sources must meet the demand in every demand block for every scenario (4.25), and the output from each generation technology in every scenario for every demand block must be less than or equal to the available capacity in that investment period t (4.26). The output from renewable energy generators(Rg) is equal to the current investment multiplied with the renewable energy factor for the load period l and technology Rg(4.27).

$$\min_{x^g_{lt\sigma}, y^g_{t\sigma}} Z = \sum_{t=1}^{T}\sum_{g=1}^{G}\sum_{\sigma=1}^{\Sigma}\sum_{l=1}^{L}(OC^g_{t\sigma}x^g_{lt\sigma}H_l)p(\sigma)+ \sum_{t=1}^{T}\sum_{g=1}^{G}\sum_{\sigma=1}^{\Sigma}(IC^g_{t\sigma}y^g_{t\sigma})p(\sigma) \quad g \in G\,,\ t \in T\,,\ \sigma \in \Sigma,\ l \in L \tag{4.24}$$

$$\sum_{g=1}^{G} x^g_{lt\sigma} = D_{lt\sigma},\ l \in L,\ t \in T,\ \sigma \in \Sigma \tag{4.25}$$

$$x^g_{lt\sigma} \leq y^g_{t\sigma},\ t \in T,\ \sigma \in \Sigma,\ g \in G,\ l \in L \tag{4.26}$$

$$x^{Rg}_{lt\sigma} = y^{Rg}_{t\sigma} RFac(Rg, l),\ t \in T,\ \sigma \in \Sigma,\ Rg \in G,\ l \in L \tag{4.27}$$

The operation problem can include additional details about the power system, such as the non-convex operational constraints of the generators. In this paper, our focus is on comparing linear multistage stochastic optimization models. Hence, we omit these constraints and assume a simple form of the dispatch problem called economic dispatch, which solves for the generator output to meet in each of the demand blocks l at each stage t.

We consider two types of demand load blocks to represent the demand for the entire year in our experiment. The first load blocks consist of 16 hours to approximate the whole year's operation cost. This model is used to validate the solutions of the three methods with a deterministic equivalent implementation. The second load blocks are 1, 4, 8, 12, and 24 weeks selected from the entire year to approximate the operations. The two types of load blocks are calculated using the alfa algorithm described in [60]. The experiment related to a higher number of weeks to approximate the operations is chosen to understand the effect of operations problem size increasing on accuracy and computational effort. The alfa algorithm samples 3000 investments and uncertainties at each stage to be run in parallel to obtain hourly dispatch. This algorithm then uses Principal component analysis and a D-Optimal design algorithm to select hours/weeks that accurately represent the demand for the entire year. The demand to obtain these load blocks are based on the available data from the Western Electricity Coordinating Council [29, 60, 161].

We explore the investment strategy across multiple investment decisions and uncertainties and multiple stages with increasing sub-problem size. The generator technologies considered for investment are Coal, Nuclear, Combined Cycle Natural Gas Turbine (CCGT), Natural Gas

Combustion Turbine (GT), Coal with Carbon Capture and Storage (Coal-CCS), Combined Cycle Natural Gas Turbine with Carbon Capture and Storage (CCGT-CCS), Wind and Solar. The current capacity for each generator type is also considered and is shown in Appendix A. The uncertain parameters are future carbon prices, gas prices, nuclear prices, the capital cost of nuclear plants, the capital cost of wind, the capital cost of solar, and capital cost of natural gas combustion turbines with a uniform distribution between upper and lower bounds as shown in Appendix A. The experimental design to investigate the effects of various parameters is as below:

- We look at investment strategies for 4, 5, 6, 7, and 8 investments with two uncertain natural gas prices and carbon cost parameters and 16 hours at each stage to approximate the cost of the yearly operations. The solutions of the three methods are compared with a deterministic equivalent implementation on the same scenario tree used in PH. The order of investments added to the models is chronological, as mentioned above.
- We look at investment strategies for 4, 5, 6, 7, and 8 investments with two uncertain parameters of natural gas prices and carbon cost and with 4, 8, 12, 24, and 52 weeks at each stage to approximate the yearly operations cost. The order of investments added to the models is chronological, as mentioned above.
- We then consider the effect of 2, 3, 4, 5, 6, 7 uncertainties with eight investment technologies and 4, 8, 12, 24, and 52 weeks at each stage to approximate the cost of yearly operations. The order of uncertainties added to the models is chronological, as mentioned above.
- We then look at the effect of multiple stages considering five uncertainties and eight investment technologies with four and twelve weeks in the dispatch problem. The number of stages increases from 3 to 7 in increments of 1.

4.5 Numerical Results

We compare the three methods: (i) Progressive Hedging(PH), (ii) Approximate Dynamic Programming(ADP), and (iii) Stochastic Dual Dynamic Programming(SDDP), on the generator expansion planning problem described above for solution quality and computational effort. One of the difficulties in comparing the three methods is the different sample complexity and stopping criteria that each algorithm requires. We propose below the different parameters and stopping criteria used for the three methods:

- Progressive Hedging: Considering the large uncertainty set along with the computational tractability, we consider an upper bound of 5000 scenarios paths. The emphasis was on solution quality, so we use $\rho = 200$ and $\epsilon = 10000$ for our experiments, similar to [35].

- Approximate Dynamic Programming: We assume a linear architecture for the approximate Q-function but include bilinear and square basis functions to better capture the shape of the underlying value function. 3000 uncertainties and decisions are sampled at every stage for bootstrapping to obtain the initial function approximation. This is then followed by using ten samples per iteration (M) with the total number of iterations (K) equal to 2000. We observed that increasing the number of iterations does not have much impact on the solution quality. Lower values of M also decrease the solution quality with diminishing improvement for values greater than M=10. A learning rate of $\lambda = 0.05$ is assumed to update the approximation coefficients θ (4.12).

- Stochastic Dual Dynamic Programming: We set the maximum number of iterations equal to 150 as the stopping criteria for SDDP. Ten samples per iteration are considered similar to ADP. We mention that the total number of uncertainties samples at every stage is 1500, which is less than the 3000 samples at every stage used in [178, 191], although with 1 sample per iteration used in those studies. However, this is a tuning parameter, and the choice of these parameters can be different as the problem complexity increases or decreases.

The three methods discussed above applied to generator expansion planning were implemented in Matlab. The linear and quadratic programs (for progressive hedging) for every scenario were solved using Cplex with the Matlab Cplex API. The policy simulation to obtain the total cost in all the three methods was performed on the same scenario tree used in the Progressive Hedging implementation. The outline of the analysis is shown in figure 4.1. Before running the individual models, 3000 uncertainties and investments are sampled at each stage, and the alfa algorithm is used to select specific weeks/hours to approximate the entire year. Once the weeks/hours have been chosen, the three models of PH, SDDP, and ADP are run using these weeks/hours in the dispatch calculation and with the stopping criteria discussed above. Once these are completed, a simulation of the policies obtained in each of the three methods is performed on the same scenario tree as in Progressive Hedging to obtain the simulated total cost. All experiments are performed on a high-performance computing cluster with a 2.2 GHz Intel Xeon Processor, 24 cores, 256 GB RAM, and 40 Gbps Ethernet.

Figure 4.1: Analysis flowchart: Sample 3000 investments and uncertainties and run the alfa algorithm to obtain the specific hours/weeks for each problem. After obtaining these weeks, run ADP, SDDP and PH using these weeks to obtain a policy. Simulate this policy on the same scenario tree as PH to obtain the simulated cost.

4.5.1 Model Validation

This section compares the three methods with a deterministic equivalent (SP) formulation for the 3 stage generator expansion planning problem with 16 operating hours at each stage. The deterministic equivalent was implemented on the same scenario tree as in PH for all ten replications. Figure 4.2 shows the median % difference in simulated cost from SP for ADP, SDDP, and PH across ten replications for 4-8 investment technologies. We observe that all the methods reach close to the SP optimal solution, with PH and SDDP having similar performance. ADP has the worst performance consistently across multiple investment decisions with the solution differences between 1-2% for different investment decisions. ADP is an approximate method with a low dimensional approximation of the objective function and a sampling scheme to select investment decisions. This approximate nature of the technique can result in inferior solutions to SDDP and PH, which use optimization techniques with better guarantees.

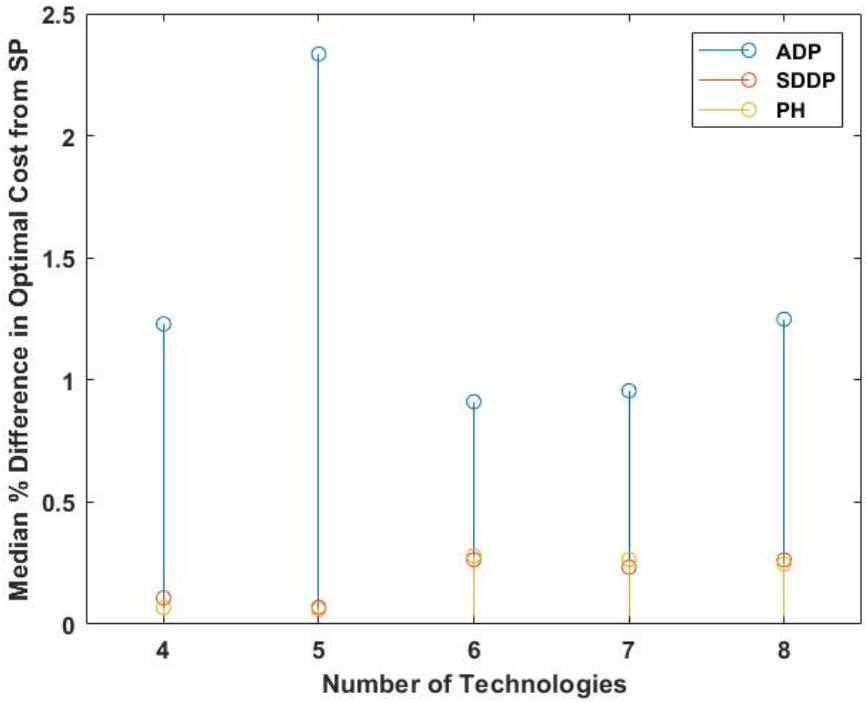


Figure 4.2: Median % difference in Simulated Total Cost from SP across 10 replications for ADP, SDDP and Progressive Hedging(PH) for 4, 5, 6, 7 and 8 investment technologies at each stage. The simulation of the total cost for ADP, SDDP and PH is on the same scenario tree as used in Progressive Hedging. The deterministic equivalent solution is also on the same scenario tree as PH.

4.5.2 Effect of Multiple Technologies

We explore the effects of the number of technologies within each of the three methods. We compare the effects of 4-8 investment technologies for 16 hours and 4, 8, 12, 24, and 52 weeks in the dispatch problem. Figure 4.3(a), 4.3(c) and 4.3(e) shows the median cost across 10 replications for 16 hours, 4 weeks, and 52 weeks respectively. We observe consistently that PH performs the best amongst all three methods with increasing problem sizes for different number of technologies. SDDP has slightly poor performance compared to PH but has a better performance than ADP for multiple investment decisions with increasing problem sizes. ADP being an approximation method has a median cost increase of 1-2% from SDDP and PH.

Figure 4.3(b), 4.3(d) and 4.3(f) shows the average time across 10 replications for 16 hours, 4 weeks, and 52 weeks respectively. PH requires the lowest computational effort when the

number of technologies and problem size is small. This is observed when the number of hours in the dispatch problem is 16 for any number of technologies and when the number of hours is four weeks, and the number of technologies is 4. Once the number of technologies increases or the problem size increases, the computational effort for PH increases exponentially compared to the SDDP and ADP. This is because of two reasons; First is that as the sub-problem size increases, the number of iterations to reach convergence increases. Second is that for every sample path, the linear/quadratic program size increases with sub-problem size. SDDP and ADP have an interesting behavior in which for problems with 16 hours, ADP has lower effort than SDDP. However, as the number of weeks increases to 4, we observe that SDDP has a lower effort than ADP, which reverses again with an increase in the number of weeks. We also observe that the computational effort does not increase substantially for SDDP with 16 hours, four weeks, and eight weeks (Appendix A) in the dispatch problem. Nonetheless, the efforts increase substantially when the dispatch problems size increases to 12, 24, and 52 weeks. In the case of ADP, we see an increase in effort when the subproblem size increases consistently, and effort is lower than SDDP for 24 and 52 weeks, consistently with increase in number of technologies. This suggests that with the problem structure in SDDP of investment and dispatch decisions being solved together, there is a minimum threshold of computational effort present which could be higher/lower than ADP depending upon the size of the linear program, which depends on the number of hours in the dispatch problem. As the sub-problem size increases, ADP has a substantial computational advantage over SDDP because of its ability to sample investments and only solve an operations problem for that investment.

We also observed from the results that the computational effort increases with number of technologies for all methods within any specific sub-problem size. This is expected for all methods because the number of variables in the optimization problem increase with the number of technologies. For ADP, another factor that increases computational effort is that the sampling policy takes more time to accept samples when the number of technologies increase.

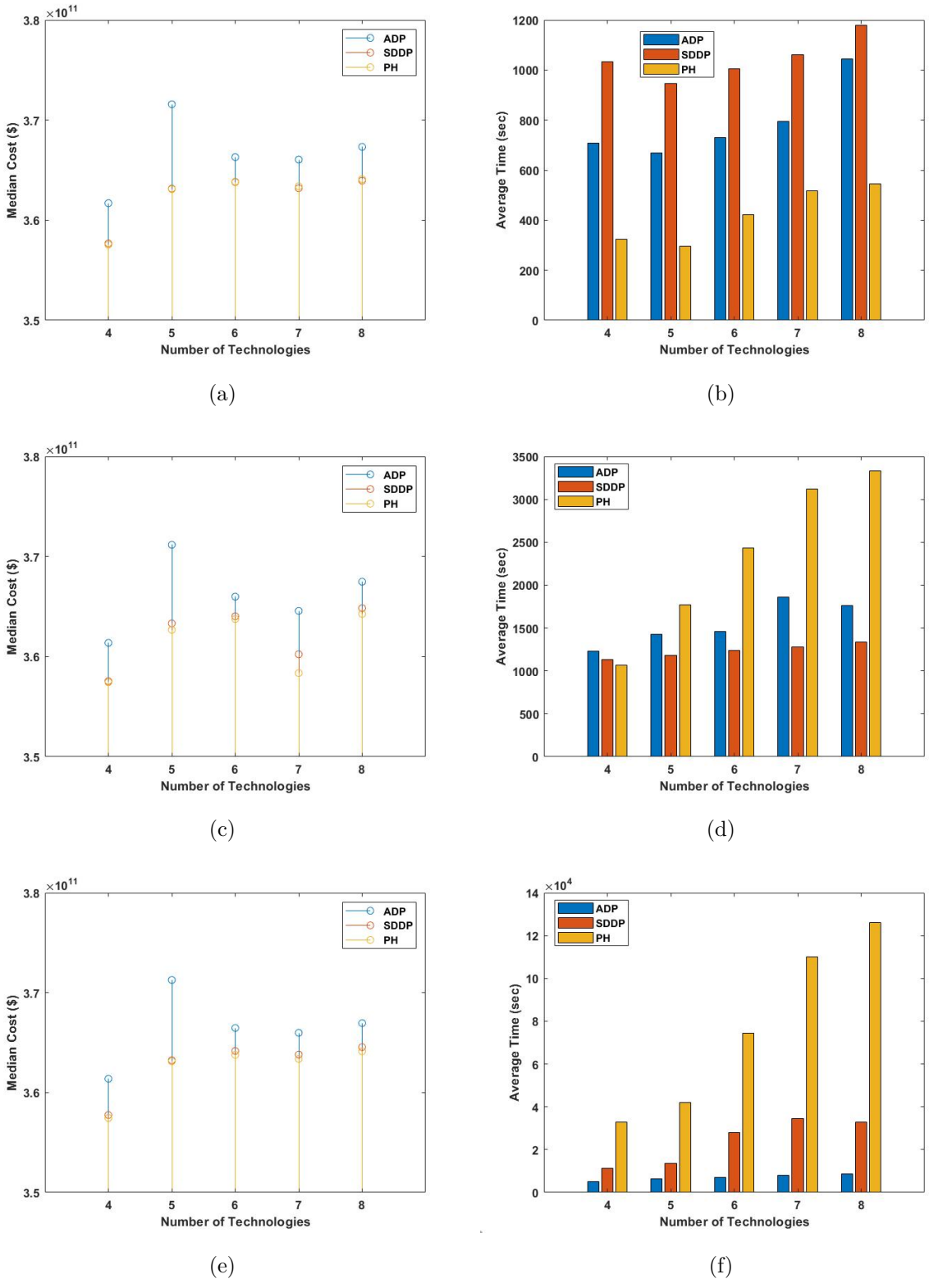


Figure 4.3: **4.3(a)**, **4.3(c)** and **4.3(e)** is the median simulated cost across 10 replications for ADP, SDDP and PH with 16 hours, 4 weeks and 52 weeks in the dispatch problem respectively for 4, 5, 6, 7 and 8 technologies.**4.3(b)**, **4.3(d)** and **4.3(f)** is the average time across 10 replications for ADP, SDDP and PH with 16 hours, 4 weeks and 52 weeks in the dispatch problem respectively for 4, 5, 6, 7 and 8 technologies.Please note change in Y axis scale for**4.3(b)**, **4.3(d)** and 4.3(f).

4.5.3 Effect of Multiple Uncertainties

We examine the effects of increasing the uncertainties in the system across all three methods with 4, 8, 12, 24, and 52 weeks in the dispatch problem. Figure 4.4(a) and 4.4(c) show the median simulated cost across ten replications for ADP, SDDP, and PH with 4 and 24 weeks in the dispatch problem. We observe that for four weeks, PH has the best performance amongst all methods. There is no clear distinction between ADP and SDDP because either one can outperform the other for problems with a different number of uncertainties. Similar behavior is observed for 24 weeks in the dispatch problem as no method is discernibly better than the other, albeit the differences are pretty small. PH has better performance for smaller problem sizes indicating that when the problem size is small, the non-linear program can provide better solutions than when the problem size increases.

Figure 4.4(b) and 4.4(d) show the average time across ten replications for ADP, SDDP, and PH with 4 and 24 weeks in the dispatch problem. For four weeks in the dispatch problem, SDDP has the lowest computational effort, followed by ADP except for seven uncertainties and then PH. However, as dispatch problem size increases, ADP has the lowest computational effort, followed by SDDP and then PH. The computational effort for all methods increases with problem size. but PH has an exponential rise compared to the other two methods. This is because a more computationally intensive quadratic program(for PH) has to be solved at each stage compared to a linear program(for SDDP and ADP) for 24 weeks compared to 4 weeks. An increase in the number of uncertainties has a linear increase in the computational effort for ADP with no substantial increase for SDDP and PH. This is because the linear approximation size increases with the number of uncertainties, increasing the time required for sampling the investment decisions. This is not present in SDDP and PH, for which the uncertainties are sampled uniformly between upper and lower bounds, and the size of the linear or quadratic program depends on the number of investment technologies for a single scenario path rather than the number of uncertainties present in the system.

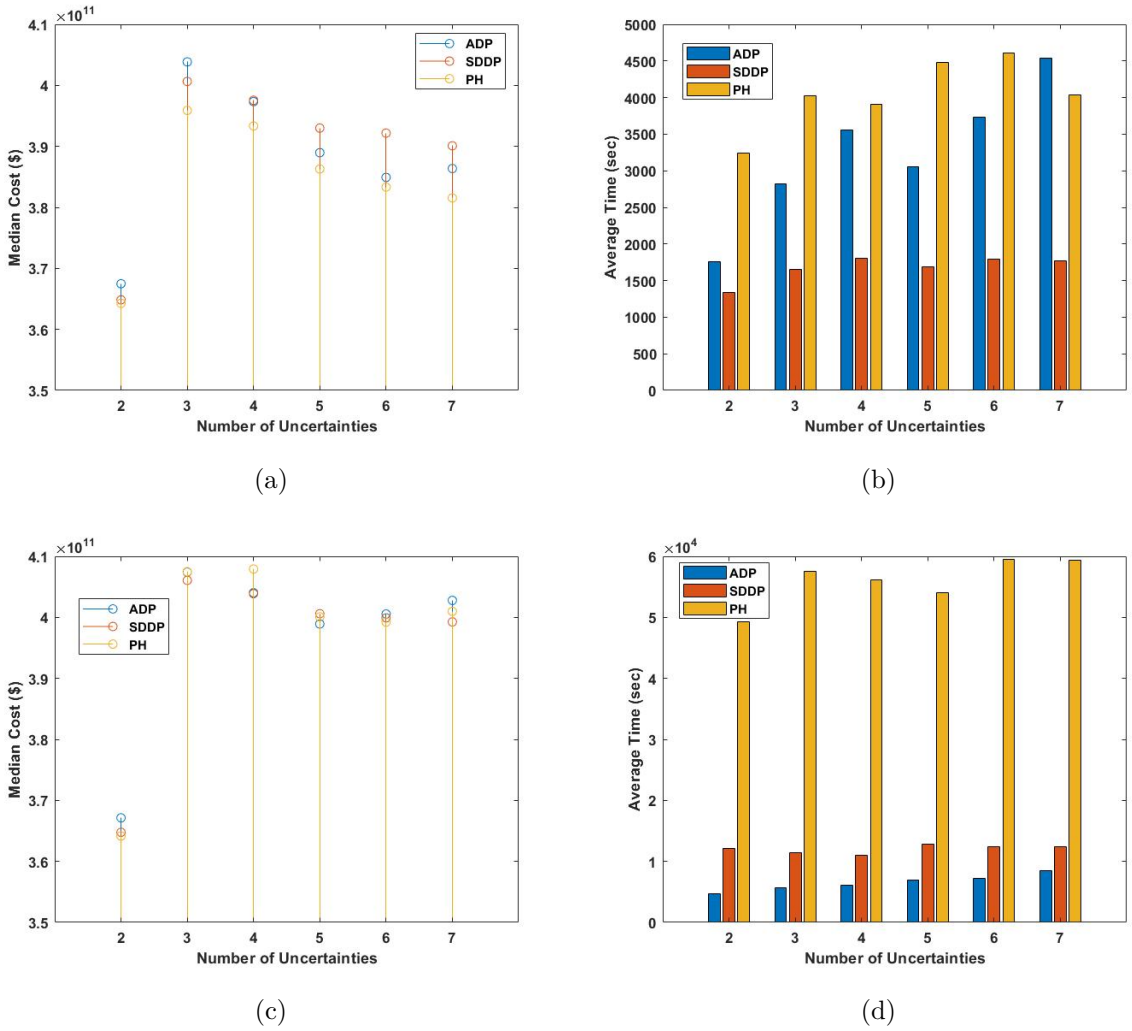
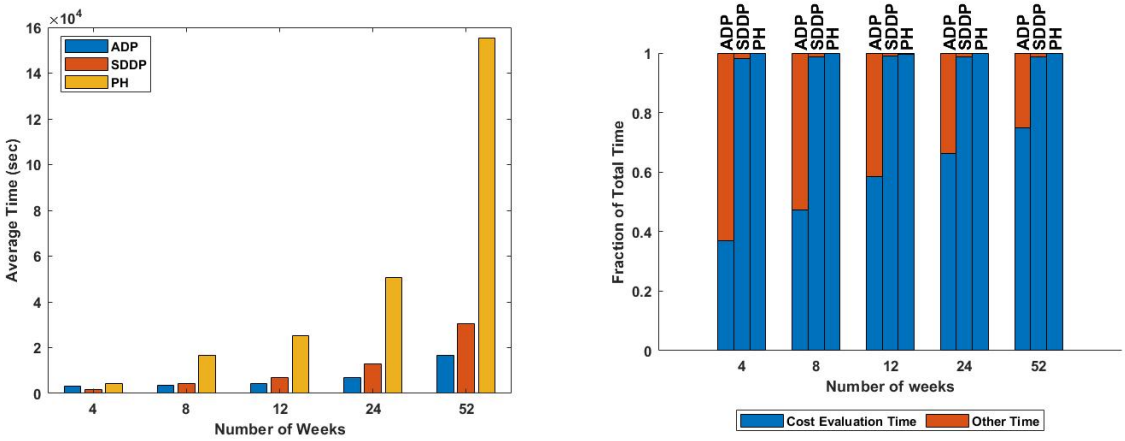


Figure 4.4: **4.4(a)** and **4.4(c)** is the median simulated cost across 10 replications for ADP, SDDP and PH with 4 weeks and 24 weeks in the dispatch problem respectively for 2, 3, 4, 5, 6, 7 uncertainties at every stage.**4.4(b)** and **4.4(d)** is the average time across 10 replications for ADP, SDDP and PH with 4 weeks and 24 weeks in the dispatch problem respectively for 2, 3, 4, 5, 6, 7 uncertainties at every stage. Please note change in Y axis scale for **4.4(b)** and **4.4(d)**.

The effect of increasing the size of the problem has an exponential increase in computational effort for PH, whereas the effort on ADP and SDDP has a linear increase as shown in Figure 4.5(a) for five uncertainties and eight investment decisions with 4, 8, 12, 24 and 52 weeks in the dispatch problem at each stage. For larger problem sizes, SDDP has higher effort than ADP because an investment and operation problem needs to be solved together, which can take substantially higher time than just an operations problem alone, especially as the number of constraints increases as the iterations progress because of the cuts added from

the next stage. No new constraints are added to the optimization problem in ADP. So the time spent in optimization is relatively constant across iterations, with the time spent in sampling investments determining the computational effort. In PH, the time spent within each iteration is constant, and it depends on the time required to solve the quadratic/non-linear program. This is also observed when we partition the total time spent by each of the methods into two sub-tasks as shown in figure 4.5(b): "Cost Evaluation Time" is the time spent in the linear/quadratic program evaluation; "Other Time" is the time spent in sampling and parallel communication between workers. For up to 8 weeks in ADP, we observed that the "Other Time" is the determining factor and not the time spent in cost evaluation. The Cost Evaluation Time then becomes dominant, increasing to 70% of the total time when the number of weeks increases to 52. For PH and SDDP, the time spent in Cost Evaluation is always the determining factor, increasing with problem size. The sampling time in ADP remains constant irrespective of problem size because it depends on the linear function approximation only. Hence, for smaller problems, ADP has a higher effort than PH or SDDP. Still, as the problem size increases, ADP starts to have the higher benefit of separating the investment and operations decisions.



(a) (b)

Figure 4.5: **4.5(a)** is the Average Total time (in hours) across 10 replications for ADP, SDDP and PH for 4, 8, 12, 24 weeks in the sub-problem with 5 uncertainties and 8 investment decisions at each stage. Only one iteration of PH with 52 weeks is performed because of the very high computational effort of 40 hours. SDDP and ADP average time for 52 weeks is averaged across 10 replications. **4.5(b)** is the average fraction of total time spent in linear or quadratic program evaluation (Cost Evaluation Time) and in sampling and other communications(Other Time) across 10 replications for ADP, SDDP and PH for 4, 8, 12, 24 weeks in the sub-problem with 5 uncertainties and 8 investment decisions at each stage. Only one iteration of PH with 52 weeks is performed because of the very high computational effort of 40 hours. SDDP and ADP average fractional of total time for 52 weeks is averaged across 10 replications.

4.5.4 Effect of Multiple Stages

We investigate the effect of increasing the number of investment stages from 3-7 with eight decision variables and five uncertainties with 4 and 12 weeks in the dispatch problem as shown in Figure 4.6. As the number of stages increases, the solution quality of ADP and SDDP is more inferior to PH. Both ADP and SDDP have comparable solution quality with no significant difference between them. This suggests that increasing the number of stages results in the ADP and SDDP information from later stages not being passed efficiently to previous stages, and more iterations may be required to obtain better solutions. It is important to note that when the number of stages increases, PH has very few distinct scenarios representing each stage (Considering that the total number of scenario paths is bound to 5000), and so these results may also be biased.

The cost of a better solution in PH comes at a substantial increase in computational effort as the number of stages increases, or the dispatch problem becomes larger. PH has

the highest solution time across all three methods, whereas ADP has lower computational effort than SDDP with 12 weeks in the dispatch problem. The order between SDDP and ADP is reversed when we consider four weeks in the dispatch problem, a similar behavior we have observed before when examining the effect of increasing the number of uncertainties or investment decisions.

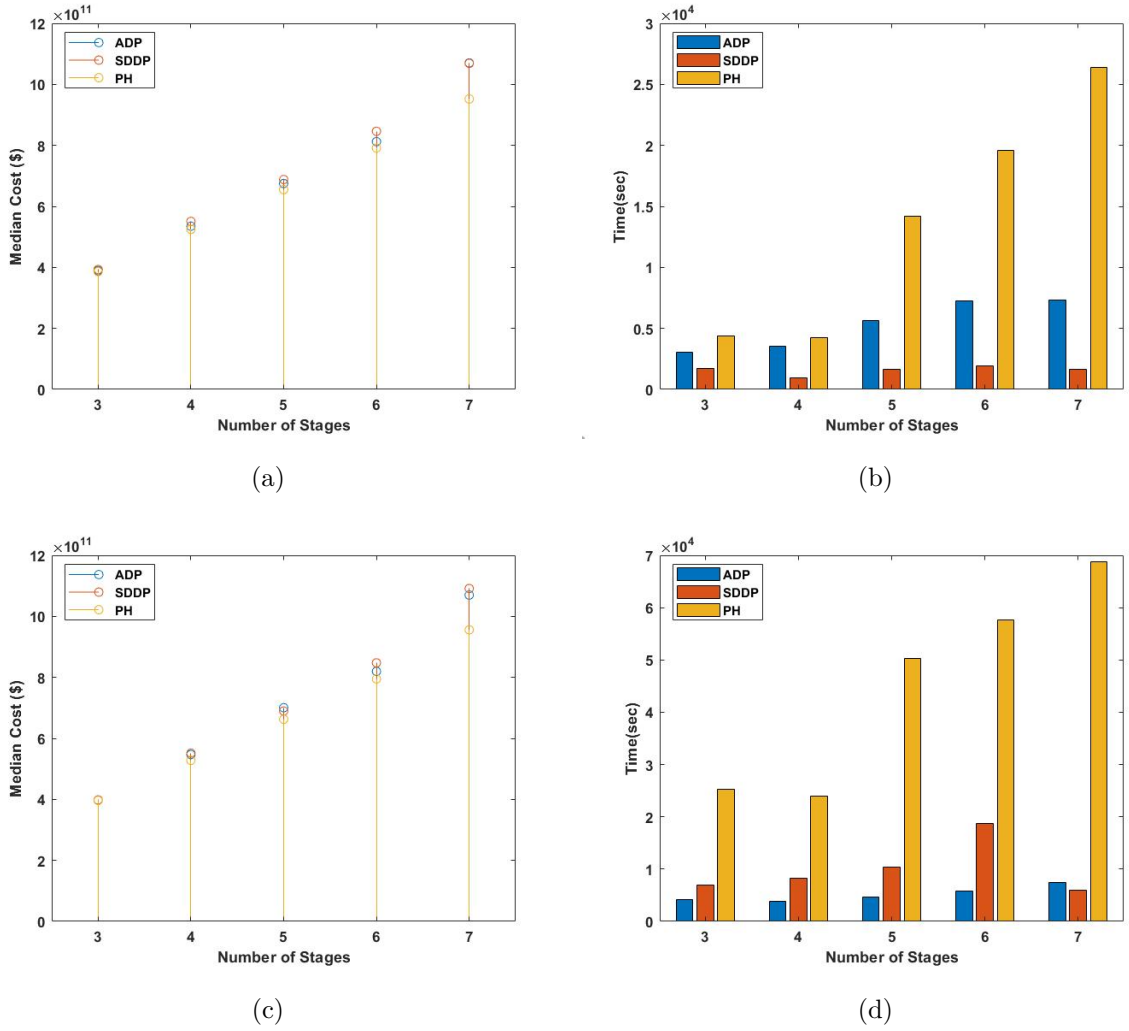


Figure 4.6: **4.6(a)** and **4.6(c)** is the median simulated cost across 10 replications for ADP, SDDP and PH with 4 weeks and 12 weeks in the dispatch problem respectively for 3, 4, 5, 6, 7 stages with 5 uncertainties and 8 investment decisions. **4.6(b)** and **4.6(d)** is the average time across 10 replications for ADP, SDDP and PH with 4 weeks and 12 weeks in the dispatch problem respectively for 3, 4, 5, 6, 7 stages with 5 uncertainties and 8 investment decisions. Please note change in Y axis scale for **4.6(b)** and **4.6(d)**.

4.5.5 Solution Variability

Along with the average solution quality, it is essential to look at the variance of the methods across replications. It is necessary to consider methods that produce a low variance and whether the variance is comparable with the average or small enough to be overlooked for other technique benefits. Table 4.1 shows the standard deviation for the ADP, SDDP, and PH for 2-7 uncertainties with 4, 8, 12, 24, and 52 weeks across ten replications. We observe that no method has any specific pattern in reducing/increasing variance as the number of uncertainties increases or the problem size increases. ADP and SDDP have a high standard deviation for particular problems ($\geq 10^{10}$), with the percentage of standard deviation by mean, can be as high as 4-5 %. However, for most problems, the standard deviation is below 10^{10} and the percentage of standard deviation by mean is less than 1%. PH generally has the lowest variance for any combination of weeks in the dispatch problem and uncertainty. This can be because the PH solution is simulated on the same scenario tree over which it is optimized. We note that the variance of ADP and SDDP can potentially reduce when using Monte-Carlo simulation, which we describe next.

Stochastic Dual Dynamic Programming and Approximate Dynamic Programming are methods developed on the tenets of dynamic programming invented by Bellman. These methods output a policy, which can be thought of as a set of decision rules for the different states/uncertainties present in the system similar to dynamic programming. The advantage of these methods is that they can be generalized. The obtained policy could be simulated for multiple future scenarios to obtain a set of decisions without solving an optimization problem every time. On the other hand, Progressive hedging outputs a solution to a specific scenario tree, and the optimization problem needs to be resolved for every different scenario tree.

Another advantage of SDDP and ADP is to obtain better estimates of the variance of the policies by simulating these policies across one set of scenarios. This could especially be useful when doing Monte Carlo simulation to provide theoretical guarantees of the variance of these methods. Using different scenario trees to evaluate the policy for independent replications could potentially provide an inaccurate estimate of the variance than what is present, as shown in Table 4.2. Table 4.2 shows the standard deviation of the total cost between ADP and SDDP when using the scenario tree simulation used in PH for different replications and the Monte Carlo simulation of the policies using the same scenarios across ten replications. We observe that the standard deviation has close to 40-50 % reduction in the Monte Carlo simulation for specific problems than a scenario tree simulation. ADP has a lower variance than SDDP for 3 and 4 uncertainties, but it increases substantially as the uncertainties

Method	Uncertainties	Weeks(10^8)				
		4	8	12	24	52
ADP	2	17.89	10.17	8.98	9.23	12.03
	3	242.43	22.00	19.84	25.58	3.64
	4	241.76	26.29	27.68	31.75	5.67
	5	67.00	20.55	32.67	28.43	104.63
	6	127.01	63.20	97.96	79.56	66.50
	7	58.48	78.23	76.90	76.22	127.44
SDDP	2	23.68	5.94	7.90	135.57	5.61
	3	230.17	19.37	128.53	19.38	6.94
	4	70.29	33.33	33.61	30.30	14.79
	5	105.00	37.52	29.64	27.83	14.87
	6	109.81	36.04	35.93	37.60	9.99
	7	54.58	29.81	30.58	27.78	9.35
PH	2	8.11	6.73	4.58	3.17	3.22
	3	50.03	43.00	0.00	21.99	14.11
	4	61.72	44.92	34.02	27.68	33.45
	5	21.56	55.39	15.68	21.59	15.01
	6	49.74	21.84	36.41	31.18	0.00
	7	88.88	19.11	29.27	29.68	27.71

Table 4.1: Standard Deviation for ADP, SDDP and PH for 2, 3, 4, 5, 6 and 7 uncertainties with 4, 8, 12, 24 and 52 weeks in the dispatch problem across 10 replications.

increase. This suggests that increasing the number of uncertainties results in a less accurate approximation of the value function used in ADP, thereby reducing solution quality.

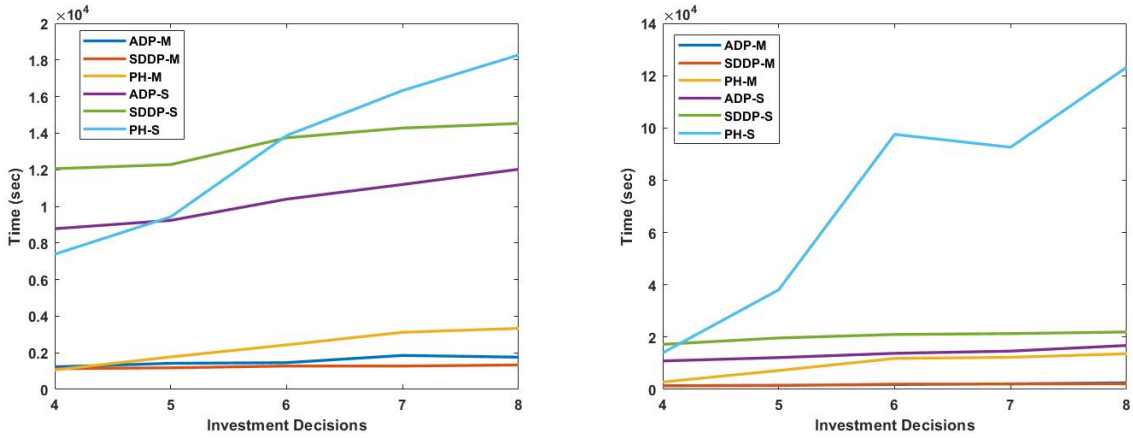
	Scenario Tree		Monte Carlo	
Uncertainties vs Method	**ADP** (10^8)	**SDDP**(10^8)	**ADP**(10^8)	**SDDP**(10^8)
3	3.64	6.94	4.78	6.50
4	5.67	14.79	7.13	12.60
5	104.63	14.87	15.35	14.18
6	66.50	9.99	33.23	9.83
7	127.44	9.35	84.46	10.61

Table 4.2: Standard Deviation of the total cost in SDDP and ADP between Scenario Tree simulation and Monte Carlo simulation for 3, 4, 5, 6 and 7 uncertainties with 8 investment decisions and 52 weeks in the dispatch.

4.5.6 High Performance Computing

The availability of high-performance computing environments has increased in the last decade. These facilities have been used to address computationally intensive problems with parallel computation, which could not be handled before. This section compares the effects of the three methods using distributed computing with 20 processors and 256 GB of RAM with a single processor and 20 GB of RAM for multiple investment decisions for 4, 8, 12, 24, and 52 weeks in the dispatch problem. We also set a computational time limit of 50 hours for this experiment. Figure 4.7(a) and 4.7(b) show the average time across ten replications for 4 and 8 weeks in the dispatch problem, respectively, for 4, 5, 6, 7, and 8 investment decisions with three investment stages. 'M' and 'S' indicate parallel processing and single-core computation respectively in front of each method name in the plot legends. We observe that ADP has the lowest computational effort when using single core computation except for four technologies and four weeks in the dispatch when PH has the lowest effort. It is also observed that SDDP does consistently worse than ADP, but the differences between PH and SDDP depend on problem size, with PH having higher computational effort than SDDP for problems with more decisions, uncertainties, or weeks with single-core computation. Using parallel computation, the computational effort decreases for all methods with a maximum percentage decrease of SDDP, followed by ADP and PH similarly as shown in Table 4.3. However, problems with 12, 24, and 52 weeks in the dispatch problem for PH can only be solved using parallel computation within the 50-hour time limit, thus showing the benefits of using multiple processors. Parallel computation in SDDP is beneficial because of stage-wise decomposition, allowing multiple scenarios at each stage to be solved in parallel, especially in

the backward pass. In ADP, the problem is already decomposed by stage, and by having investments sampled, the benefit of using parallel processing is less than SDDP, although still useful for large problems. This also shows that separating investment and operation problems has a benefit over only stage-wise decomposition in SDDP. PH benefits from parallel processing because instead of solving each scenario path sequentially, the scenarios could be sent to multiple processors for faster computation. However, considering the number of scenarios and multiple iterations required for convergence, the time needed in PH is larger than ADP and SDDP with parallel computation.



(a) (b)

Figure 4.7: Average Time across 10 replications for 4 weeks(4.7(a)) and 8 weeks(4.7(b)) for 4, 5, 6, 7 and 8 investment decisions with single core and parallel computation.'M' and 'S' indicates parallel processing and single core computation respectively in front of each method name in the plot legends.

	4 Weeks			8 Weeks		
Decisions vs Method	**ADP**	**SDDP**	**PH**	**ADP**	**SDDP**	**PH**
4	5.90	10.08	6.00	7.07	12.66	4.01
5	5.71	9.66	4.59	6.99	11.65	4.28
6	5.95	10.29	4.72	6.82	9.18	7.20
7	5.57	10.36	4.04	6.08	9.36	6.29
8	5.47	9.85	4.66	5.73	8.97	8.19

Table 4.3: Median fractional increase in single core computation time from parallel processing time for 4 and 8 weeks in the dispatch problem for ADP, SDDP and PH with 4, 5, 6, 7 and 8 investment decisions across 10 replications.

4.6 Abstract Comparison

Along with comparing each of the methods for solution quality and computational effort, the methods have abstract advantages and disadvantages of their own. PH and SDDP retain the mathematical formulation of the linear program, which can be beneficial for problems with abstract decision rules or uncertainties for which obtaining a lower order approximation in ADP will be more challenging. Obtaining an approximation in ADP is more an art than science. It can be time-consuming, especially in an adaptive environment where the decision and uncertainty set can change depending on earlier decisions. However, for certain problem classes such as chess or Atari games where a mathematical formulation is not present, ADP is the only method that can be applied.

All of the methods presented in this work have tuning parameters. PH has tuning parameters of ρ, ϵ and the size of the scenario tree. Increasing the size of the scenario tree gives a better representation of the uncertainty but comes at the cost of higher computational effort. Lower values of ϵ and ρ give better solutions but with higher effort. ADP has the tuning parameter λ to update the approximation, the sampling policy, total number of iterations (K) and number of samples per iteration (M). λ, K and sampling policy are heuristic and would need to be tuned based on the application. Generally more number of samples per iteration (M) gives better solutions but at higher effort. The only tuning parameter in SDDP is the number of samples (M) per iteration. More number of samples improves solution quality but at higher effort. Even very small sample sizes (M=1) are also used because they give reasonable solutions with low effort. We notice that SDDP has the lowest number of tuning

parameters, followed by PH and then ADP. PH and SDDP are also generally robust to the tuning parameters with ADP being the most sensitive.

Although SDDP has the least number of tuning parameters and is robust to these parameters, it has a strong assumption about stage-wise independence so that the cuts can be shared across all scenarios. Some recent work has been done to address the independence assumption [185] but this assumption can be invalid for many applications. The problem and stage-wise decomposition in ADP is potentially powerful when the sub-problem for every investment is computationally intensive. A numerical example of this in our analysis is when the number of weeks in the operations problem is greater than 12 weeks. In this scenarios, SDDP with only stage wise decomposition alone has higher effort than ADP with both problem and stage-wise decomposition.

ADP has the highest sample complexity compared to SDDP and PH. In our experiment, ADP uses 23000 samples compared to SDDP which uses 1500 samples at the maximum or PH which uses 5000. The high sample complexity in ADP has also been observed in other applications. The high sample complexity may not be suited for certain problem classes such as queuing or where obtaining large number of samples is not possible. For these problem classes, it maybe beneficial to use SDDP and PH compared to ADP.

4.7 Recommendations

In this section, we give recommendations of which of the methods described here are better suited for the different problem classes and the possible decision-maker constraints.

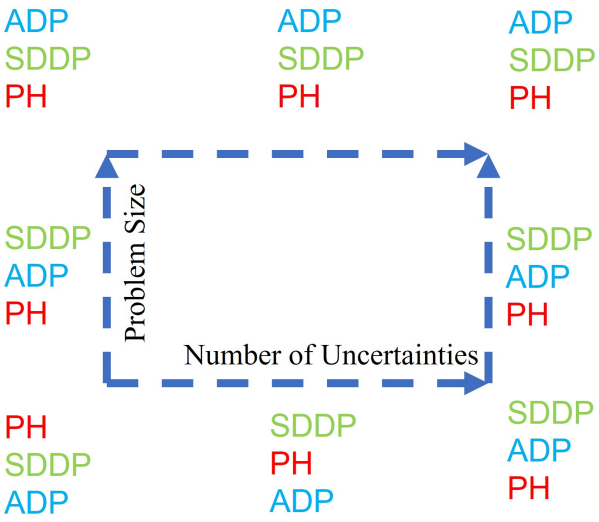
4.7.1 Problem Classes

Figure 4.8 show the framework of when to use each of the three methods depending upon the number of uncertainties, stages, and problem size. Problem size includes the number of variables in the optimization problem, which is determined by the number of hours/weeks in the operations problem and the number of technologies (for a capacity expansion problem). The interpretation of the figures is as follows: The left-hand bottom corner represents the smallest problem size and number of uncertainties. Then one can imagine increasing the number of uncertainties on the x-axis and the problem size on the y-axis or both simultaneously. At each node, the methods are described from best suited to least suited for the specific problem class. A mid-point node in the square along both the x and y-axis is shown because the recommendation could change at this location when the number of uncertainties and

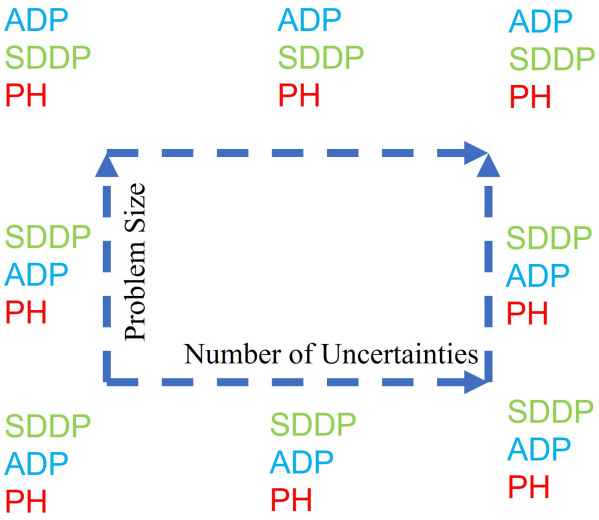
problem size are moderate. Figure 4.8(a) shows the recommendations when the number of decision stages is small (≤ 5), and figure 4.8(b) shows the recommendations when the number of stages is higher than 5.

- **Number of stages is less than or equal to 5**: We recommend that when the problem size and number of uncertainties is small, PH should be used because it can be easily applied and is quite robust to the tuning parameters. This is followed by SDDP and ADP being the least preferred. SDDP is better than PH when the number of uncertainties increases and the problem size is small. However, if the number of uncertainties is such that a scenario tree can represent the uncertainty reasonably and the problem size is small, PH would be better than ADP. This is because ADP has more tuning parameters and expertise is required to construct a low dimensional approximation. Once the number of uncertainties is large, and a scenario tree is not practical, then ADP is better than PH. For moderate problem sizes, SDDP is better suited than all three methods. ADP is better than PH for any number of uncertainties at this problem size because the computational effort with PH is too large. However at this problem size SDDP is still recommended over ADP because of the tuning parameters in ADP and the time required to obtain the low dimensional approximation. When the problem size is large, ADP's stagewise and problem decomposition framework makes it the best suited for these problems. When the problem size is large it can be beneficial to invest time in obtaining the low dimensional approximation for ADP to take advantage of its decomposition framework. SDDP can also be applied but at a considerably higher effort. PH becomes prohibitive at this time and is not recommended.

- **Number of stages is greater than 5**: When the number of stages is more than 5, PH should be the least preferred method even for a small number of uncertainties and problem sizes. This is because the number of scenarios at each stage to be represented will be small to keep computational tractability. Hence the representation of the uncertainty space would be poor. In these cases, for small and moderate problem sizes, SDDP is the best method, followed by ADP and then PH. ADP would be the recommended method for large problem sizes, followed by SDDP and then PH because of its decomposition framework. This framework would ensure that the problem is tractable and has lower effort than SDDP.



(a) Number of stages is less than or equal to 5



(b) Number of stages is greater than 5

Figure 4.8: 4.8(a) is a recommendation of the three methods for different number of uncertainties and problem sizes when the number of decision stages is less than or equal to 5. 4.8(b) is a recommendation of the three methods for different number of uncertainties and problem sizes when the number of decision stages is greater than 5.

4.7.2 Availability of high-performance computing

Although the availability of high-performance computing facilities has increased over the last decade, not all decision-makers have access to these resources because of cost constraints or the technical skills required to use them. In these cases when high-performance computing is not easily available, it may be beneficial to use ADP over SDDP and PH. The stagewise and problem decomposition structure in ADP lowers the computational burden significantly compared to PH and SDDP. When high-performance computing facilities are available, the recommendation of section 4.7.1 can be used to determine which method is better applicable.

4.7.3 Sample complexity

Through this analysis, we saw that ADP has the highest sample complexity, followed by PH and then SDDP. For problems where uncertainty samples are difficult to obtain such as queuing, it would be better to apply SDDP and PH compared to ADP.

4.7.4 Mathematical formulation

Certain problems which do not have an explicit mathematical formulation such as chess or Atari games cannot use PH and SDDP, both of which require an explicit mathematical optimization model. ADP is the only method that can be applied to these problem classes. Problems that have a mathematical formulation can be solved by all three methods. A low dimensional approximation of the objective function is required in ADP, and it can be time-consuming to obtain this approximation. In these situations, the recommendations of section 4.7.1 can be used to determine which method is better applicable.

4.7.5 Policy

It can sometimes be beneficial to obtain a policy than a solution for some problems. The policy is a mapping of the uncertainties to the decisions taken in those uncertainties. It is not necessary for the algorithm to experience all the uncertainties to get this mapping. ADP and SDDP give a policy, whereas PH gives a solution to a scenario tree for the uncertainties that are represented. With some post-processing after the algorithm is terminated, ADP and SDDP can also give a solution to the scenario tree that is used in PH. Problems that require policies should consider only SDDP and ADP, whereas, for a solution, the recommendations of section 4.7.1 can be used to determine which method is better applicable.

4.8 Discussion

In this chapter, we have compared three broad methods which can be applied to solve large-scale multistage stochastic optimization for which deterministic equivalent solutions can be computationally intractable. The three methods are progressive hedging, approximate dynamic programming, and stochastic dual dynamic programming. The comparison was made for computational effort and solution quality on multiple decisions, uncertainties, problem sizes, and stages where decisions have to be made. When considering a small number of uncertainties along with a smaller problem size, PH has good performance in terms of solution quality and computational time. However, as problem sizes increase, the computational effort increases exponentially. This is one area where the decomposition approaches of ADP and SDDP can provide solutions with less computational effort. SDDP consistently has good solution quality compared to ADP when increasing problem size or investment decisions with few uncertainties. ADP has the poorest solution quality amongst the three for a small number of uncertainties. Still, its stage-wise and problem decomposition approach can provide good approximate solutions with considerably less computational effort, especially for problems with more weeks in the dispatch. As the number of uncertainties increases, no method outperforms the other, and all of them provide comparable solutions. The stage-wise and problem decomposition in ADP offers an approach to obtain comparable solutions in the lowest computational effort. A similar pattern is observed as the number of investment stages increases in terms of computational effort. For solution quality, PH has the best solution quality with the increasing number of stages, but these results could be biased considering a small number of scenarios are used at each stage for computational tractability.

ADP and SDDP generally have higher solution variability than PH, with the standard deviation by mean percentage being as large as 5%. However, the PH solution was evaluated on the same scenario tree as it was optimized, and the results could potentially be biased. Furthermore, PH provides a solution to a scenario tree, whereas ADP and SDDP provide policies for which Monte-Carlo simulation can be performed. Monte-Carlo simulation helps obtain accurate estimates of variance and providing theoretical guarantees for the specific method. We observe that the Monte-Carlo simulation can reduce variance as the same set of sample paths are used across replications rather than a different scenario tree across replications.

In addition to the numerical results, we also outline the abstract advantages and disadvantages of each of the three methods. We also provide recommendations on which of these

methods is better suited based on the problem class and the constraints of the decision maker. The demonstration of the benefits of the stage-wise and problem decomposition approach in ADP has been shown by considering an entire year of dispatch in the capacity expansion planning problem. Although this work used a linear economic dispatch, the decomposition framework in ADP can easily include unit commitment constraints for the dispatch. Solving an investment and unit commitment for multiple weeks together can be computationally very intensive for every scenario, even using the stage-wise decomposition approach in SDDP. Still, with the stage-wise and problem decomposition approach in ADP, it would be possible to solve multiple weeks of unit commitment in parallel for an investment decision, thus reducing the effort substantially. This is potentially an area of future work with many applications, considering the increasing penetration of renewable energy in the grid, thereby requiring operations to be accurately represented.

Chapter 5
Conclusions

This chapter gives an overview of the major findings in this research and possible future work. This research developed some of the tools required to do long-term planning under uncertainty and with sufficient temporal and spatial complexity in the power grid to obtain valid estimates of impacts of decisions. The objective is that these methods can guide managers to make investment decisions that are more reliable, economical, and resilient to climate change-induced extreme events, thereby benefiting society.

5.1 Complexity in Power Systems

Traditionally power system operations could be approximated using a merit order approach with the plants coming online from cheapest to most expensive depending upon the demand and renewable energy generation. This approach was feasible because of low renewable energy penetration, and the number of extreme events in the system was low. Recently with a rise of extreme events such as colder winters and higher temperatures, managers of power systems need to consider the effects of other physical systems such as land, water, and economy on the power grid. One such effect is the rising temperatures of water which could potentially affect power plants that require water for cooling. Evaluating these impacts on power system models with low complexity in operations or spatially could indicate higher flexibility than what is present. This work compares the effects of using power systems with different spatial and operations complexity and shows the impact of water stress on the power grid and the economy. We also look at which of the spatial and operational constraints provide maximum enhancement to the model by looking at the power system impacts, economic impacts and computational effort.

5.2 Exploration in Q Learning

Long-term system planning is adaptive decision-making under uncertainty problem. Instead of a two-stage stochastic programming approach with a here and now and wait and see decision, it makes decisions based on how the uncertainty evolves. These are generally called multi-stage stochastic optimization problems, and one approach to this problem is using approximate dynamic programming or reinforcement learning. ADP or RL samples uncertainties and investments and uses a low dimensional approximation of the objective function, which is updated using stochastic approximation methods. As with any sampling-based optimization method, it suffers from the 'explore vs. exploit' problem where the 'explore' sample helps understand the less visited areas of the decision space, whereas the 'exploit' sample improves the approximation with the best value for the current approximation. We propose to address this problem using an importance sampling-based approach in which decisions are sampled proportional to the current function approximation. We call this method Q-Importance Sampling (QIS) because it can be only done within a Q Learning framework and uses importance sampling to disproportionately sample areas of the decision space.

5.3 Multi Stage Stochastic Optimization Schemes

Multi-stage stochastic adaptive planning is a computationally intensive problem, and its deterministic equivalent with enough scenarios can be hard to solve in a reasonable amount of time. Hence decomposition schemes have been developed to address this computational intractability. They can be broadly classified as stagewise decomposition methods such as stochastic dual dynamic programming and nested benders decomposition and scenario decomposition methods such as progressive hedging and lagrangian relaxation. Another type of decomposition method which has been widely applied is stagewise and problem decomposition schemes such as approximate dynamic programming or reinforcement learning. We compare these different decomposition schemes for solution quality and computational effort when applied to a multi-stage generation expansion planning problem with an entire year of simulated dispatch. Our results indicate the ADP can provide reasonable solutions with lower effort than the other methods. However, if the focus is on solution quality with a reasonable effort, SDDP offers a good approach. Progressive hedging is quite expensive and provides a solution to a specific scenario tree instead of a policy, and can be difficult to

generalize. SDDP is especially useful if scenario generation is hard, as in cases of queuing because it has the lowest sample complexity. ADP has the highest sample complexity as it requires many samples to update the approximation. Future work could include using ADP with a lower number of samples or efficiently reusing samples.

5.4 Future Work

5.4.1 Short Term Operations in GEP

ADP/RL provides an approach to problem decomposition, which could be exploited when doing long-term planning under uncertainty and including short-term operations of power plants. Short-term operations of power plants are represented through production cost models developed in Chapter 1 of this research. These models contain physical constraints of generators such as ramping and uptime/downtime and system-wide reserve requirements. These short-term operational constraints could potentially change the investment decisions, especially under higher renewable energy penetration, and have been observed in deterministic models. Although outside the scope of the dissertation, preliminary results indicate that including these short-term constraints within a multi-stage stochastic optimization framework using ADP can affect investment decisions.

5.4.2 Short Term Uncertainty in Long Term Planning

This thesis has focused on long-term uncertainties present in generation expansion planning such as gas prices, cost of carbon, and investment costs reduction in power plants. Another uncertainty present is the ability of the investment decisions to handle short-term uncertainty in the system, such as variability of demand and renewable generation. Although this work did not include these uncertainties, it provided an approach to handling short- and long-term uncertainties. The short-term uncertainties can be modeled using an SDDP approach using only linear models or with unit commitment constraints using a recently developed method called Stochastic Dual Dynamic Integer Programming (SDDIP) within an ADP/RL method focused on long term planning. The problem decomposition structure in ADP does not require us to solve the stochastic version of the short-term uncertainty with the investment decision, making it less computationally intensive.

5.4.3 Risk Averse Generation Expansion Planning

The methods focused in this work are primarily risk-neutral schemes applied to capacity planning. Risk-neutral methods focus on minimizing the average cost for the entire horizon. However, some situations require that the decision-maker be risk-averse or a risk-taker. The former is common in generation expansion planning or with heavy-tailed distributions problems as in finance. ADP using a risk-averse approach has been focused on small problems or a lookup table approach but has not been generalized using function approximations. The QIS method developed could potentially be modified to consider a risk-averse criterion while selecting investment decisions.

5.4.4 Value of Multi-Stage decisions

Multi-stage Generation Expansion Planning is computationally intensive, and the decomposition methods presented in this thesis provide one approach to address the computational intractability. Other simplifications commonly used are rolling 2 stage optimization, reduced scenario trees, planning for a single target year, planning for an average scenario, etc. It would be beneficial to show the value of using an adaptive multi-stage stochastic optimization approach to these other approaches present in the literature for solution quality and decision biases.

5.4.5 Long Term Planning with Economic Impacts

Long-term generation planning is generally modeled using either a central decision-maker or a bi-level approach to maximize profits for a single entity. An exciting area of future work will be to do long-term planning with detailed operations constraints and the economic model presented in Chapter 2. The economic model would improve the welfare of the entire society compared to the profit maximization of a single entity. Similar work has been considered with merit order dispatch in the operations in [192].

Appendix A

A.1 Application Overview and Details

This section goes over the parameters applied to the generator expansion planning problem described in Chapter 4. The demand, renewable energy data, generator capital cost, fixed cost, variable operating and maintenance cost, heat rate, and emission rate are obtained from the Western Electricity Coordinating Council. The current capacity for each generator type is shown in Table A.1.

Fuel Type	**Existing Capacity**	**Fuel Type**	**Existing Capacity**
Natural Gas Combustion Turbine	9760	Coal with CCS	5950
Combined Cycle Gas Turbine	12260	CCGT with CCS	13325
Coal	9260	Wind	15620
Nuclear	8260	Solar	1440

Table A.1: Existing capacity for each generator type at $t = 0$.

The model has three-seven investment stages, with the stages assumed to be a number of years apart as shown in Table A.2. Depending upon the number of uncertainties for the three-stage problem, the uncertain parameters upper and lower bounds (LB) are shown in Table A.3 and A.4. The investment cost uncertainty is represented as a fraction of the investment cost in stage 1.

Stages	Years between stages	Stages	Years between stages
3 Stages	20	6 Stages	8
4 Stages	13	7 Stages	7
5 Stages	10		

Table A.2: Years between investment stages for stages three-eight.

	Gas Price		Carbon Price		Nuclear Price	
Stage	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
Stage 1	3.2	3.2	50	50	0.8	0.8
Stage 2	3	7	0	100	0.4	3
Stage 3	3	11	100	300	0.4	5

Table A.3: Uncertainty bounds for natural gas price, carbon price and nuclear price for three stages.

	Nuclear		Wind		Solar		GT	
Stage	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
Stage 2	0.7	1.3	0.6	1	0.6	1	0.7	1.3
Stage 3	0.4	1.6	0.4	0.8	0.4	0.8	0.7	1.6

Table A.4: Range of investment costs [fraction to Stage 1] for three stages

A.2 Multiple Technologies

This section shows the figures of 8, 12, and 24 weeks in the dispatch problem when considering multiple technologies of Chapter 4. Figure A.1(a), A.1(c), A.1(e) is the median cost across 10 replications for ADP, SDDP and PH for 8, 12 and 24 weeks respectively with 4-8 technologies. Figure A.1(b), A.1(d), A.1(f) is the average time across 10 replications for ADP, SDDP and PH for 8, 12 and 24 weeks respectively with 4-8 technologies.

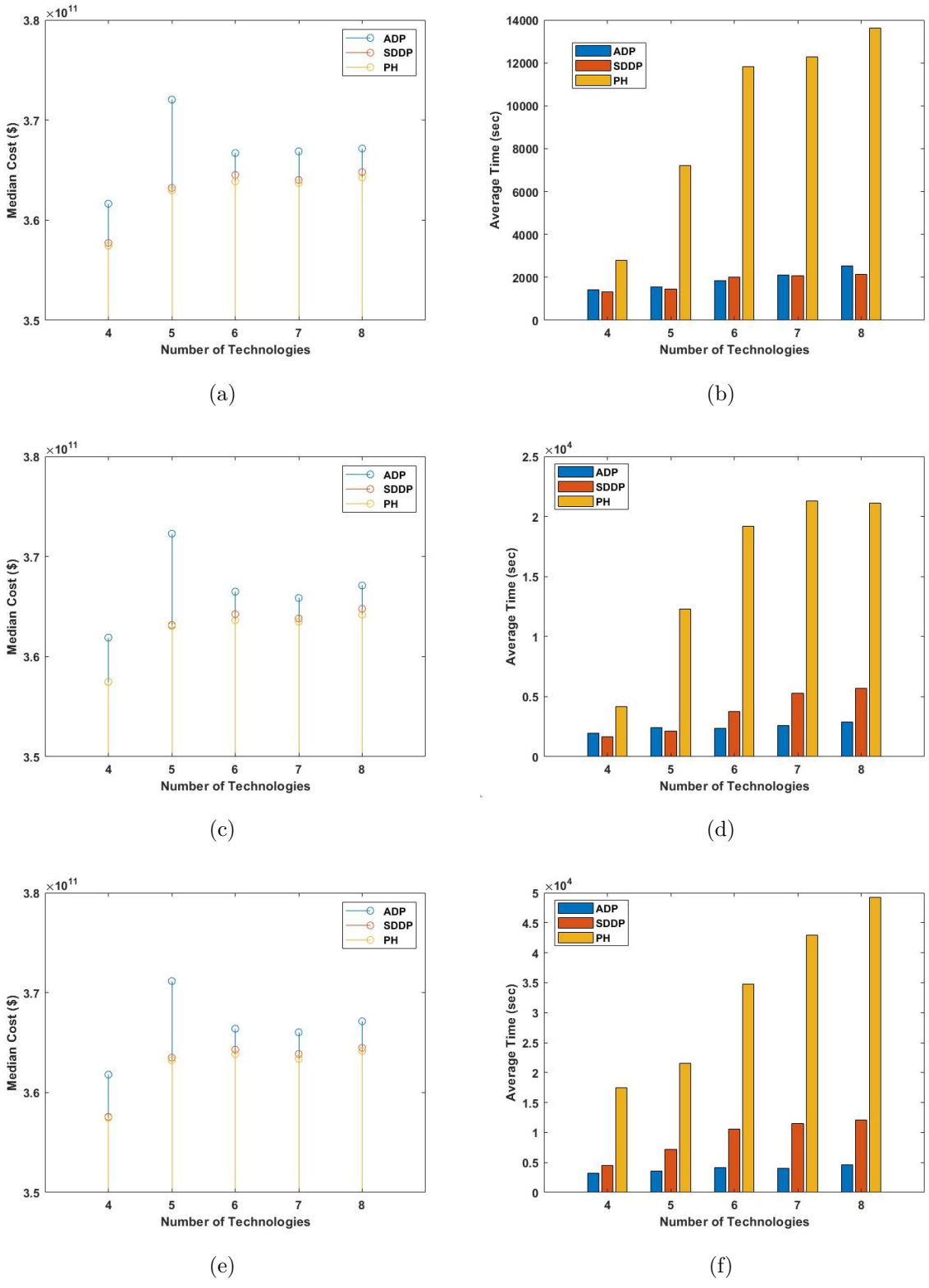


Figure A.1: **A.1(a)**, **A.1(c)** and **A.1(e)** is the median simulated cost across 10 replications for ADP, SDDP and PH with 8 weeks, 12 weeks and 24 weeks in the dispatch problem respectively for 4, 5, 6, 7 and 8 technologies. **A.1(b)**, **A.1(d)** and **A.1(f)** is the average time across 10 replications for ADP, SDDP and PH with 8 weeks, 12 weeks and 24 weeks in the dispatch problem respectively for 4, 5, 6, 7 and 8 technologies. Please note change in Y axis scale for **A.1(b)**, **A.1(d)** and **A.1(f)**.

A.3 Multiple Uncertainties

This section shows the figures of 8, 12, and 52 weeks in the dispatch problem when considering multiple uncertainties of Chapter 4. Figure A.2(a), A.2(c), A.2(e) is the median cost across 10 replications for ADP, SDDP, and PH for 8, 12, and 52 weeks, respectively with 2-7 uncertainties and 8 investment decisions. Figure A.2(b), A.2(d), A.2(f) is the average time across 10 replications for ADP, SDDP and PH for 8, 12 and 52 weeks respectively with 2-7 uncertainties and 8 investment decisions.

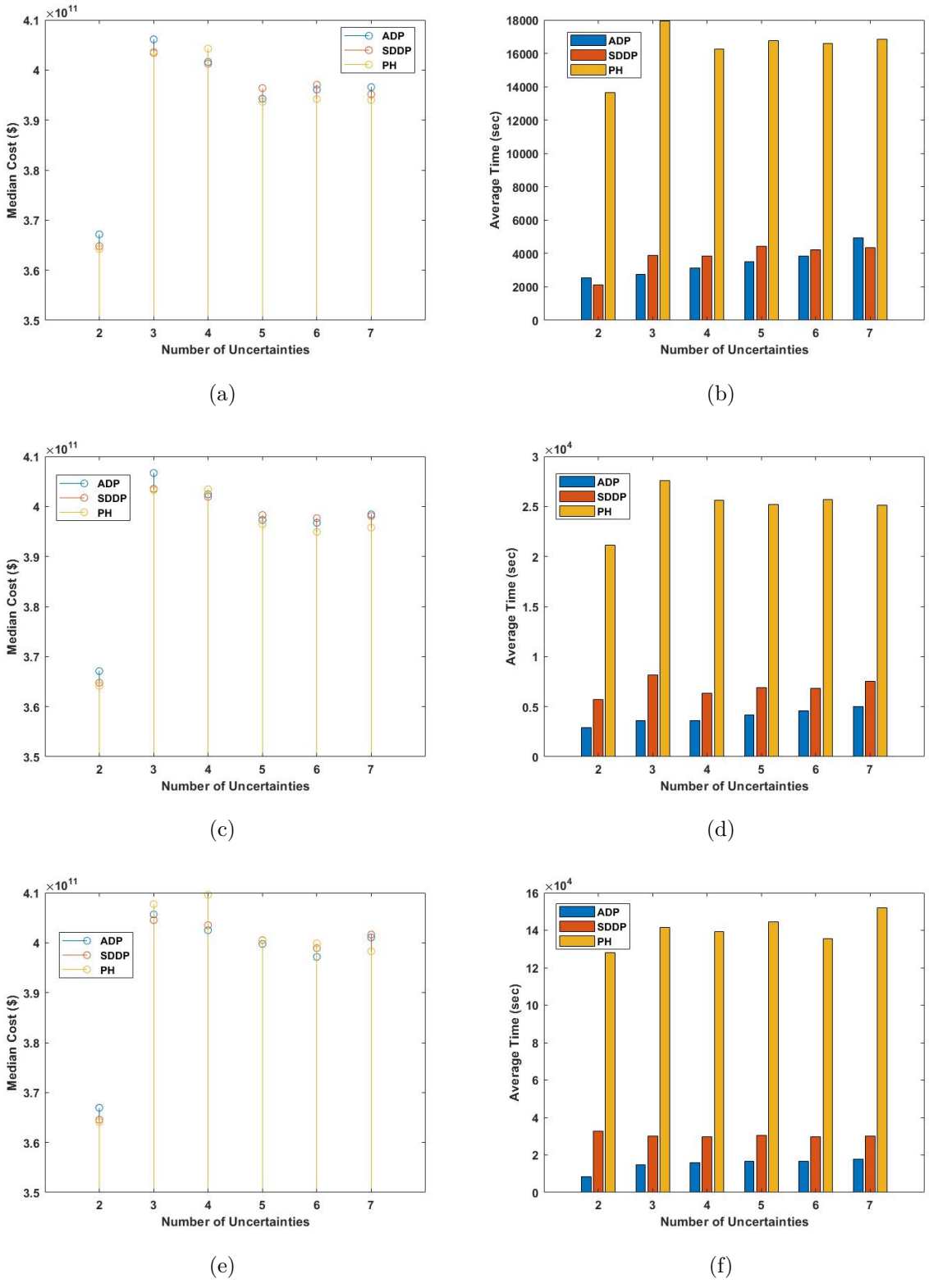


Figure A.2: **A.2(a)** and **A.2(c)** is the median simulated cost across 10 replications for ADP, SDDP and PH with 8 and 12 weeks in the dispatch problem respectively for 2, 3, 4, 5, 6, 7 uncertainties. **A.2(e)** is the median simulated cost across 10 replications for ADP and SDDP with 52 weeks in the dispatch problem respectively for 2-7 uncertainties. Because of the high computational effort the PH cost with 52 weeks is shown for one replication only. **A.2(b)** and **A.2(d)** is the average time across 10 replications for ADP, SDDP and PH with 8 and 12 weeks in the dispatch problem respectively for 2, 3, 4, 5, 6, 7 uncertainties. **A.2(f)** is the average time across 10 replications for ADP and SDDP with 52 weeks in the dispatch problem respectively for 2-7 uncertainties.

Appendix B

This appendix gives some additional outputs from the coupled water-energy-economy coupling model.

B.1 Outages from water scenarios

Table B.1 gives the number of hours with atleast one hour of outage, the average capacity outage over these hours and the maximum outage across these hours for GFDL-CM3 2041-2099 water scenarios from WBM.

MODEL	YEAR	HOURS OUTAGE	AVG OUTAGE	Max OUTAGE
GFDL-CM3	2041	1488	4324	19517
GFDL-CM3	2042	1800	4741	17532
GFDL-CM3	2043	1512	1731	6647
GFDL-CM3	2044	408	1393	3675
GFDL-CM3	2045	1560	2467	10490
GFDL-CM3	2046	648	1755	8059
GFDL-CM3	2047	1320	2950	11222
GFDL-CM3	2048	1344	2934	13176
GFDL-CM3	2049	1824	4804	17209
GFDL-CM3	2050	1632	3176	14350
GFDL-CM3	2051	1296	3137	18367
GFDL-CM3	2052	2040	4023	19932
GFDL-CM3	2053	1104	1619	7917
GFDL-CM3	2054	1560	1929	9100

GFDL-CM3	2055	1752	3672	14760
GFDL-CM3	2056	1800	4086	16654
GFDL-CM3	2057	1776	3664	15015
GFDL-CM3	2058	696	1700	7800
GFDL-CM3	2059	1680	3937	20649
GFDL-CM3	2060	1824	2270	13599
GFDL-CM3	2061	2016	3571	13554
GFDL-CM3	2062	1416	2672	10761
GFDL-CM3	2063	1872	4479	16408
GFDL-CM3	2064	2256	7773	22543
GFDL-CM3	2065	1704	6749	25205
GFDL-CM3	2066	1824	4945	21095
GFDL-CM3	2067	1224	2187	8908
GFDL-CM3	2068	1488	3143	15956
GFDL-CM3	2069	2472	6321	25101
GFDL-CM3	2070	2256	6917	24855
GFDL-CM3	2071	1584	4240	14916
GFDL-CM3	2072	2112	6482	20649
GFDL-CM3	2073	2304	6223	20499
GFDL-CM3	2074	1752	6143	27533
GFDL-CM3	2075	2352	11090	38085
GFDL-CM3	2076	1944	7204	27492
GFDL-CM3	2077	1896	4890	16297
GFDL-CM3	2078	2448	8932	33771
GFDL-CM3	2079	1944	4479	19354
GFDL-CM3	2080	2304	7572	20392
GFDL-CM3	2081	2616	7823	25385
GFDL-CM3	2082	2352	8784	29922
GFDL-CM3	2083	2112	9397	26576
GFDL-CM3	2084	2232	6301	25201
GFDL-CM3	2085	2040	7224	23083
GFDL-CM3	2086	2352	9504	37361
GFDL-CM3	2087	1968	7172	25838

GFDL-CM3	2088	2304	8742	29714
GFDL-CM3	2089	2496	8943	27857
GFDL-CM3	2090	3312	10311	39432
GFDL-CM3	2091	2136	8702	24729
GFDL-CM3	2092	3312	9087	32679
GFDL-CM3	2093	2160	9309	26298
GFDL-CM3	2094	3312	9841	38831
GFDL-CM3	2095	1656	2828	10381
GFDL-CM3	2096	2472	9592	29882
GFDL-CM3	2097	2832	12733	43766
GFDL-CM3	2098	2448	9587	31441
GFDL-CM3	2099	3072	11037	36511

Table B.1: The number of hours with non-zero generation capacity unavailable(HOURS OUTAGE), the capacity unavailable averaged over all hours for which this is non-zero(AVG OUTAGE), and the maximum capacity unavailable in any one hour(MAX OUTAGE) for GFDL-CM3 2041-2099 water scenarios from WBM.

B.2 Economic Impacts

This section gives information on the detailed economic impacts by sector for ED, UC, OPF and UC+OPF after equilibrium with the economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GFDL-CM3	2050	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
GFDL-CM3	2061	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
GFDL-CM3	2075	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
GFDL-CM3	2077	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2078	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2081	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2083	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2085	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2086	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2089	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
GFDL-CM3	2090	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
GFDL-CM3	2092	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
GFDL-CM3	2094	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
GFDL-CM3	2096	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
GFDL-CM3	2097	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00
GFDL-CM3	2098	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
GFDL-CM3	2099	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00

Table B.2: Percentage change by sector for California for ED after equilibrium between ED and economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	0.00	0.00	0.00	0.00	0.00	-0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2050	0.00	0.00	0.00	0.00	0.00	-0.06	0.00	0.00	0.00	0.00
GFDL-CM3	2061	0.00	0.00	0.00	0.00	0.00	-0.04	0.00	0.00	0.00	0.00
GFDL-CM3	2075	0.00	-0.01	0.01	0.00	-0.01	-0.22	0.00	0.00	0.00	0.00
GFDL-CM3	2077	0.00	-0.01	0.01	0.00	0.00	-0.18	0.00	0.00	0.00	0.00
GFDL-CM3	2078	0.00	-0.01	0.01	0.00	-0.01	-0.20	0.00	0.00	0.00	0.00
GFDL-CM3	2081	0.00	-0.01	0.01	0.00	-0.01	-0.18	0.00	0.00	0.00	0.00
GFDL-CM3	2083	0.00	-0.01	0.01	0.00	0.00	-0.17	0.00	0.00	0.00	0.00
GFDL-CM3	2085	0.00	-0.01	0.01	0.00	0.00	-0.16	0.00	0.00	0.00	0.00
GFDL-CM3	2086	0.00	-0.01	0.01	0.00	-0.01	-0.19	0.00	0.00	0.00	0.00
GFDL-CM3	2089	0.00	-0.01	0.01	0.00	-0.01	-0.23	0.00	0.00	0.00	0.00
GFDL-CM3	2090	0.00	-0.02	0.02	0.00	-0.01	-0.28	0.01	0.00	0.00	0.00
GFDL-CM3	2092	0.00	-0.02	0.02	0.00	-0.01	-0.27	0.01	0.00	0.00	0.00
GFDL-CM3	2094	0.00	-0.02	0.02	0.00	-0.01	-0.29	0.01	0.00	0.00	0.00
GFDL-CM3	2096	0.00	-0.01	0.01	0.00	-0.01	-0.24	0.00	0.00	0.00	0.00
GFDL-CM3	2097	0.00	-0.02	0.02	0.00	-0.01	-0.33	0.01	0.00	0.00	0.00
GFDL-CM3	2098	0.00	-0.01	0.01	0.00	-0.01	-0.24	0.00	0.00	0.00	0.00
GFDL-CM3	2099	0.00	-0.01	0.01	0.00	-0.01	-0.25	0.01	0.00	0.00	0.00

Table B.3: Percentage change by sector for Rest of Wecc after equilibrium between ED and economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	0.00	-0.10	0.08	0.00	-0.01	-2.29	0.02	0.00	0.00	0.00
GFDL-CM3	2050	0.00	-0.13	0.10	0.01	-0.01	-3.10	0.03	0.01	0.00	0.00
GFDL-CM3	2061	0.00	-0.12	0.10	0.00	-0.01	-3.00	0.03	0.01	0.00	0.00
GFDL-CM3	2075	0.00	-0.02	0.02	0.00	0.00	-0.47	0.00	0.00	0.00	0.00
GFDL-CM3	2077	0.00	-0.21	0.17	0.01	-0.01	-5.12	0.05	0.01	0.00	-0.01
GFDL-CM3	2078	0.00	-0.11	0.09	0.00	-0.01	-2.73	0.03	0.01	0.00	0.00
GFDL-CM3	2081	-0.01	-0.13	0.12	0.00	-0.01	-2.98	0.02	0.00	-0.01	-0.01
GFDL-CM3	2083	-0.01	0.04	0.00	0.00	0.00	1.04	-0.02	-0.01	-0.01	-0.01
GFDL-CM3	2085	0.00	-0.10	0.08	0.00	-0.01	-2.28	0.02	0.00	0.00	0.00
GFDL-CM3	2086	-0.01	-0.21	0.19	0.01	-0.01	-5.15	0.05	0.01	-0.01	-0.01
GFDL-CM3	2089	0.00	-0.23	0.19	0.01	-0.01	-5.62	0.06	0.01	0.00	-0.01
GFDL-CM3	2090	0.00	-0.13	0.10	0.01	-0.01	-3.12	0.03	0.01	0.00	0.00
GFDL-CM3	2092	0.00	-0.04	0.03	0.00	0.00	-1.00	0.01	0.00	0.00	0.00
GFDL-CM3	2094	0.00	-0.03	0.02	0.00	0.00	-0.63	0.01	0.00	0.00	0.00
GFDL-CM3	2096	-0.01	0.01	0.01	0.00	0.00	0.33	-0.01	-0.01	-0.01	0.00
GFDL-CM3	2097	0.00	-0.06	0.05	0.00	0.00	-1.44	0.01	0.00	0.00	0.00
GFDL-CM3	2098	0.00	-0.02	0.02	0.00	0.00	-0.49	0.00	0.00	0.00	0.00
GFDL-CM3	2099	0.00	-0.03	0.02	0.00	0.00	-0.60	0.01	0.00	0.00	0.00

Table B.4: Percentage change by sector for California after equilibrium between UC and economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	0.00	-0.01	0.02	0.00	-0.02	-0.13	0.00	0.00	0.00	0.00
GFDL-CM3	2050	0.00	0.01	0.00	0.00	-0.01	0.29	-0.01	0.00	0.00	0.00
GFDL-CM3	2061	-0.01	0.02	-0.02	0.00	0.00	0.56	-0.02	-0.01	-0.01	0.00
GFDL-CM3	2075	0.00	-0.01	0.01	0.00	-0.01	-0.20	0.00	0.00	0.00	0.00
GFDL-CM3	2077	-0.01	0.00	0.00	0.00	-0.02	0.32	-0.01	-0.01	-0.01	0.00
GFDL-CM3	2078	-0.01	0.06	-0.05	-0.01	0.02	1.08	-0.03	-0.01	-0.01	0.00
GFDL-CM3	2081	0.00	-0.17	0.17	0.02	-0.09	-2.66	0.05	0.02	0.01	-0.01
GFDL-CM3	2083	0.01	-0.35	0.35	0.04	-0.16	-5.89	0.12	0.05	0.03	-0.02
GFDL-CM3	2085	0.00	0.02	-0.01	0.00	0.00	0.39	-0.01	0.00	0.00	0.00
GFDL-CM3	2086	0.00	-0.18	0.18	0.02	-0.10	-2.67	0.05	0.02	0.01	-0.01
GFDL-CM3	2089	-0.01	0.00	0.01	0.00	-0.02	0.27	-0.01	-0.01	-0.01	-0.01
GFDL-CM3	2090	-0.01	0.05	-0.04	-0.01	0.01	1.00	-0.02	-0.01	-0.01	0.00
GFDL-CM3	2092	0.00	-0.01	0.01	0.00	-0.01	-0.14	0.00	0.00	0.00	0.00
GFDL-CM3	2094	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2096	0.01	-0.21	0.21	0.02	-0.10	-3.51	0.07	0.03	0.02	-0.01
GFDL-CM3	2097	0.00	-0.04	0.04	0.00	-0.02	-0.51	0.01	0.00	0.00	0.00
GFDL-CM3	2098	0.00	-0.02	0.02	0.00	-0.01	-0.27	0.00	0.00	0.00	0.00
GFDL-CM3	2099	0.00	0.04	-0.03	0.00	0.01	0.62	-0.01	-0.01	0.00	0.00

Table B.5: Percentage change by sector for Rest of Wecc after equilibrium between UC and economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	0.00	0.00	0.00	0.00	0.00	-0.09	0.00	0.00	0.00	0.00
GFDL-CM3	2050	0.00	0.00	0.01	0.00	0.00	-0.10	0.00	0.00	0.00	0.00
GFDL-CM3	2061	0.00	-0.01	0.01	0.00	0.00	-0.12	0.00	0.00	0.00	0.00
GFDL-CM3	2075	0.00	-0.03	0.03	0.00	0.00	-0.58	0.00	0.00	0.00	0.00
GFDL-CM3	2077	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
GFDL-CM3	2078	0.00	-0.02	0.02	0.00	0.00	-0.40	0.00	0.00	0.00	0.00
GFDL-CM3	2081	0.00	-0.02	0.02	0.00	0.00	-0.44	0.00	0.00	0.00	0.00
GFDL-CM3	2083	0.00	-0.02	0.02	0.00	0.00	-0.48	0.00	0.00	0.00	0.00
GFDL-CM3	2085	0.00	-0.01	0.01	0.00	0.00	-0.30	0.00	0.00	0.00	0.00
GFDL-CM3	2086	0.00	-0.02	0.02	0.00	0.00	-0.56	0.00	0.00	0.00	0.00
GFDL-CM3	2089	0.00	-0.02	0.02	0.00	0.00	-0.50	0.00	0.00	0.00	0.00
GFDL-CM3	2090	0.00	-0.04	0.04	0.00	0.00	-0.98	0.01	0.00	0.00	0.00
GFDL-CM3	2092	0.00	-0.03	0.03	0.00	0.00	-0.79	0.01	0.00	0.00	0.00
GFDL-CM3	2094	0.00	-0.04	0.04	0.00	0.00	-0.92	0.01	0.00	0.00	0.00
GFDL-CM3	2096	0.00	-0.03	0.03	0.00	0.00	-0.67	0.00	0.00	0.00	0.00
GFDL-CM3	2097	0.00	-0.05	0.05	0.00	0.00	-1.16	0.01	0.00	0.00	0.00
GFDL-CM3	2098	0.00	-0.02	0.02	0.00	0.00	-0.56	0.00	0.00	0.00	0.00
GFDL-CM3	2099	0.00	-0.05	0.04	0.00	0.00	-1.12	0.01	0.00	0.00	0.00

Table B.6: Percentage change by sector for California after equilibrium between OPF and economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	0.00	-0.01	0.01	0.00	0.00	-0.12	0.00	0.00	0.00	0.00
GFDL-CM3	2050	0.00	-0.02	0.02	0.00	-0.01	-0.30	0.01	0.00	0.00	0.00
GFDL-CM3	2061	0.00	-0.02	0.02	0.00	-0.01	-0.29	0.01	0.00	0.00	0.00
GFDL-CM3	2075	0.00	-0.07	0.08	0.01	-0.05	-1.09	0.02	0.01	0.00	0.00
GFDL-CM3	2077	0.00	-0.05	0.05	0.01	-0.02	-0.78	0.02	0.01	0.00	0.00
GFDL-CM3	2078	0.00	-0.06	0.06	0.01	-0.03	-0.95	0.02	0.01	0.00	0.00
GFDL-CM3	2081	0.00	-0.06	0.06	0.01	-0.03	-0.96	0.02	0.01	0.00	0.00
GFDL-CM3	2083	0.00	-0.05	0.05	0.01	-0.03	-0.78	0.02	0.01	0.00	0.00
GFDL-CM3	2085	0.00	-0.05	0.05	0.01	-0.02	-0.76	0.01	0.01	0.00	0.00
GFDL-CM3	2086	0.00	-0.06	0.06	0.01	-0.03	-0.90	0.02	0.01	0.00	0.00
GFDL-CM3	2089	0.00	-0.06	0.06	0.01	-0.03	-0.96	0.02	0.01	0.00	0.00
GFDL-CM3	2090	0.00	-0.09	0.09	0.01	-0.05	-1.41	0.03	0.01	0.01	0.00
GFDL-CM3	2092	0.00	-0.08	0.09	0.01	-0.05	-1.29	0.03	0.01	0.00	0.00
GFDL-CM3	2094	0.00	-0.08	0.09	0.01	-0.05	-1.27	0.02	0.01	0.00	0.00
GFDL-CM3	2096	0.00	-0.06	0.06	0.01	-0.03	-0.94	0.02	0.01	0.00	0.00
GFDL-CM3	2097	0.00	-0.09	0.12	0.01	-0.08	-1.47	0.03	0.01	0.01	-0.01
GFDL-CM3	2098	0.00	-0.06	0.06	0.01	-0.03	-0.95	0.02	0.01	0.00	0.00
GFDL-CM3	2099	0.00	-0.08	0.08	0.01	-0.04	-1.19	0.02	0.01	0.00	0.00

Table B.7: Percentage change by sector for Rest of WECC after equilibrium between OPF and economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	-0.01	-0.22	0.21	0.01	-0.01	-5.50	0.05	0.01	-0.01	-0.01
GFDL-CM3	2050	-0.01	-0.22	0.20	0.01	-0.01	-5.38	0.05	0.01	-0.01	-0.01
GFDL-CM3	2061	-0.01	-0.22	0.20	0.01	-0.01	-5.44	0.05	0.01	-0.01	-0.01
GFDL-CM3	2075	-0.01	-0.25	0.22	0.01	-0.01	-5.92	0.05	0.01	-0.01	-0.02
GFDL-CM3	2077	-0.01	-0.21	0.20	0.01	-0.01	-5.10	0.04	0.00	-0.01	-0.01
GFDL-CM3	2078	-0.01	-0.23	0.21	0.01	-0.01	-5.60	0.05	0.00	-0.01	-0.02
GFDL-CM3	2081	-0.01	-0.23	0.22	0.01	-0.01	-5.66	0.05	0.01	-0.01	-0.02
GFDL-CM3	2083	-0.01	-0.23	0.22	0.01	-0.01	-5.66	0.05	0.01	-0.01	-0.02
GFDL-CM3	2085	-0.01	-0.22	0.21	0.01	-0.01	-5.38	0.05	0.01	-0.01	-0.02
GFDL-CM3	2086	-0.01	-0.24	0.22	0.01	-0.01	-5.71	0.05	0.01	-0.01	-0.02
GFDL-CM3	2089	-0.01	-0.23	0.21	0.01	-0.01	-5.45	0.04	0.00	-0.01	-0.02
GFDL-CM3	2090	-0.02	-0.28	0.26	0.01	0.00	-6.82	0.06	0.01	-0.01	-0.02
GFDL-CM3	2092	-0.02	-0.25	0.22	0.01	0.00	-5.89	0.05	0.00	-0.01	-0.02
GFDL-CM3	2094	-0.02	-0.30	0.26	0.01	0.00	-7.15	0.06	0.01	-0.01	-0.02
GFDL-CM3	2096	-0.01	-0.24	0.21	0.01	-0.01	-5.64	0.05	0.01	-0.01	-0.02
GFDL-CM3	2097	-0.02	-0.33	0.28	0.01	0.00	-7.78	0.06	0.01	-0.01	-0.02
GFDL-CM3	2098	-0.01	-0.23	0.22	0.01	-0.01	-5.64	0.05	0.01	-0.01	-0.02
GFDL-CM3	2099	-0.02	-0.30	0.26	0.01	0.00	-7.16	0.06	0.01	-0.01	-0.02

Table B.8: Percentage change by sector for California after equilibrium between UC+OPF and economic model.

MODEL	YEAR	Financial	Agric.	Mining	Construction	Manuf.	Elec.	Utility	Retail	Services	Public
GFDL-CM3	2045	0.00	-0.24	0.24	0.03	-0.13	-3.63	0.07	0.03	0.01	-0.02
GFDL-CM3	2050	0.00	-0.25	0.26	0.03	-0.14	-3.87	0.07	0.03	0.01	-0.02
GFDL-CM3	2061	0.00	-0.25	0.29	0.03	-0.18	-3.85	0.07	0.03	0.01	-0.02
GFDL-CM3	2075	-0.01	-0.34	0.57	0.02	-0.40	-5.33	0.11	0.03	0.03	-0.02
GFDL-CM3	2077	0.00	-0.29	0.36	0.03	-0.22	-4.60	0.09	0.03	0.02	-0.02
GFDL-CM3	2078	-0.01	-0.32	0.43	0.03	-0.28	-4.99	0.10	0.03	0.02	-0.02
GFDL-CM3	2081	-0.01	-0.32	0.43	0.03	-0.27	-4.97	0.10	0.03	0.02	-0.02
GFDL-CM3	2083	0.00	-0.30	0.34	0.03	-0.20	-4.66	0.09	0.03	0.02	-0.02
GFDL-CM3	2085	0.00	-0.29	0.39	0.03	-0.25	-4.58	0.09	0.03	0.02	-0.02
GFDL-CM3	2086	-0.01	-0.31	0.52	0.02	-0.36	-4.94	0.10	0.03	0.02	-0.02
GFDL-CM3	2089	-0.01	-0.31	0.50	0.02	-0.35	-4.84	0.10	0.03	0.02	-0.02
GFDL-CM3	2090	-0.02	-0.37	0.64	0.02	-0.45	-5.87	0.12	0.03	0.03	-0.03
GFDL-CM3	2092	-0.02	-0.35	0.66	0.02	-0.48	-5.59	0.12	0.03	0.03	-0.02
GFDL-CM3	2094	-0.03	-0.36	0.76	0.01	-0.57	-5.67	0.13	0.02	0.03	-0.03
GFDL-CM3	2096	-0.01	-0.30	0.49	0.02	-0.34	-4.74	0.10	0.03	0.02	-0.02
GFDL-CM3	2097	-0.04	-0.39	0.96	0.00	-0.75	-6.23	0.15	0.02	0.04	-0.03
GFDL-CM3	2098	-0.01	-0.32	0.45	0.03	-0.30	-4.96	0.10	0.03	0.02	-0.02
GFDL-CM3	2099	-0.03	-0.35	0.77	0.01	-0.59	-5.51	0.12	0.02	0.03	-0.03

Table B.9: Percentage change by sector for Rest of WECC after equilibrium between UC+OPF and economic model.

Bibliography

[1] Craig, M. T., S. Cohen, J. Macknick, C. Draxl, O. J. Guerra, M. Sengupta, S. E. Haupt, B.-M. Hodge, and C. Brancucci (2018) "A review of the potential impacts of climate change on bulk power system planning and operations in the United States," *Renewable and Sustainable Energy Reviews*, **98**, pp. 255–267.

[2] Miara, A., J. E. Macknick, C. J. Vörösmarty, V. C. Tidwell, R. Newmark, and B. Fekete (2017) "Climate and water resource change impacts and adaptation potential for US power supply," *Nature Climate Change*, **7**(11), pp. 793–798.

[3] Van Vliet, M. T., J. Sheffield, D. Wiberg, and E. F. Wood (2016) "Impacts of recent drought and warm years on water resources and electricity supply worldwide," *Environmental Research Letters*, **11**(12), p. 124021.

[4] Bartos, M. D. and M. V. Chester (2015) "Impacts of climate change on electric power supply in the Western United States," *Nature Climate Change*, **5**(8), pp. 748–752.

[5] Ramseur, J. (2019) "US Carbon Dioxide Emissions in the Electricity Sector: Factors, Trends, and Projections," *Congressional Research Service Report*.

[6] Feldman, D., V. Ramasamy, R. Fu, A. Ramdas, J. Desai, and R. Margolis (2021) *US Solar Photovoltaic System and Energy Storage Cost Benchmark: Q1 2020*, *Tech. rep.*, National Renewable Energy Lab.(NREL), Golden, CO (United States).

[7] "EIA battery storage cost," `https://www.eia.gov/todayinenergy/detail.php?id=45596`.

[8] Cole, W. J. and A. Frazier (2019) *Cost projections for utility-scale battery storage*, *Tech. rep.*, National Renewable Energy Lab.(NREL), Golden, CO (United States).

[9] Masse, P. and R. Gibrat (1957) "Application of linear programming to investments in the electric power industry," *Management Science*, **3**(2), pp. 149–166.

[10] Anderson, D. (1972) "Models for determining least-cost investments in electricity supply," *The Bell Journal of Economics and Management Science*, pp. 267–299.

[11] Schaeffer, P. V. and L. J. Cherene (1989) "The inclusion of 'spinning reserves' in investmetn and simulation models for electricity generation," *European journal of operational research*, **42**(2), pp. 178–189.

[12] Booth, R. (1972) "Optimal generation planning considering uncertainty," *IEEE Transactions on Power Apparatus and Systems*, (1), pp. 70–77.

[13] Sanghvi, A. P. and I. H. Shavel (1986) "Investment planning for hydro-thermal power system expansion: Stochastic programming employing the Dantzig-Wolfe decomposition principle," *IEEE transactions on power systems*, **1**(2), pp. 115–121.

[14] Gorenstin, B., N. Campodonico, J. Costa, and M. Pereira (1993) "Power system expansion planning under uncertainty," *IEEE Transactions on Power Systems*, **8**(1), pp. 129–136.

[15] Mo, B., J. Hegge, and I. Wangensteen (1991) "Stochastic generation expansion planning by means of stochastic dynamic programming," *IEEE Transactions on Power Systems*, **6**(2), pp. 662–668.

[16] David, A. and R.-D. Zhao (1989) "Integrating expert systems with dynamic programming in generation expansion planning," *IEEE Transactions on Power Systems*, **4**(3), pp. 1095–1101.

[17] Nakamura, S. (1984) "A review of electric production simulation and capacity expansion planning programs," *International journal of energy research*, **8**(3), pp. 231–240.

[18] Marwali, M. and S. Shahidehpour (1998) "A deterministic approach to generation and transmission maintenance scheduling with network constraints," *Electric power systems research*, **47**(2), pp. 101–113.

[19] Chen, Q., C. Kang, Q. Xia, and J. Zhong (2010) "Power generation expansion planning model towards low-carbon economy and its application in China," *IEEE Transactions on Power Systems*, **25**(2), pp. 1117–1125.

[20] Meza, J. L. C., M. B. Yildirim, and A. S. Masud (2007) "A model for the multiperiod multiobjective power generation expansion problem," *IEEE Transactions on Power Systems*, **22**(2), pp. 871–878.

[21] Aghaei, J., M. A. Akbari, A. Roosta, and A. Baharvandi (2013) "Multiobjective generation expansion planning considering power system adequacy," *Electric power systems research*, **102**, pp. 8–19.

[22] Kannan, S., S. M. R. Slochanal, P. Subbaraj, and N. P. Padhy (2004) "Application of particle swarm optimization technique and its variants to generation expansion planning problem," *Electric Power Systems Research*, **70**(3), pp. 203–210.

[23] Sirikum, J., A. Techanitisawad, and V. Kachitvichyanukul (2007) "A new efficient GA-benders' decomposition method: For power generation expansion planning with emission controls," *IEEE Transactions on Power Systems*, **22**(3), pp. 1092–1100.

[24] Bakirtzis, G. A., P. N. Biskas, and V. Chatziathanasiou (2012) "Generation expansion planning by MILP considering mid-term scheduling decisions," *Electric Power Systems Research*, **86**, pp. 98–112.

[25] Kamalinia, S. and M. Shahidehpour (2010) "Generation expansion planning in wind-thermal power systems," *IET generation, transmission & distribution*, **4**(8), pp. 940–951.

[26] Aghaei, J., N. Amjady, A. Baharvandi, and M.-A. Akbari (2014) "Generation and transmission expansion planning: MILP–based probabilistic model," *IEEE Transactions on Power Systems*, **29**(4), pp. 1592–1601.

[27] Manso, L. and A. L. Da Silva (2004) "Probabilistic criteria for power system expansion planning," *Electric Power Systems Research*, **69**(1), pp. 51–58.

[28] Hobbs, B. F., S. Kasina, Q. Xu, S. W. Park, J. Ouyang, J. L. Ho, and P. E. Donohoo-Vallett (2016) "What is the Benefit of Including Uncertainty in Transmission Planning? A WECC Case Study," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, IEEE, pp. 2364–2371.

[29] Munoz, F. D., B. F. Hobbs, J. L. Ho, and S. Kasina (2013) "An engineering-economic approach to transmission planning under market and regulatory uncertainties: WECC case study," *IEEE Transactions on Power Systems*, **29**(1), pp. 307–317.

[30] Nagl, S., M. Fursch, and D. Lindenberger (2013) "The costs of electricity systems with a high share of fluctuating renewables: a stochastic investment and dispatch optimization model for Europe," *The Energy Journal*, **34**(4).

[31] Jin, S., S. M. Ryan, J.-P. Watson, and D. L. Woodruff (2011) "Modeling and solving a large-scale generation expansion planning problem under uncertainty," *Energy Systems*, **2**(3-4), pp. 209–242.

[32] Zhan, Y., Q. P. Zheng, J. Wang, and P. Pinson (2016) "Generation expansion planning with large amounts of wind power via decision-dependent stochastic programming," *IEEE Transactions on Power Systems*, **32**(4), pp. 3015–3026.

[33] Zou, J., S. Ahmed, and X. A. Sun (2018) "Partially adaptive stochastic optimization for electric power generation expansion planning," *INFORMS journal on computing*, **30**(2), pp. 388–401.

[34] Park, H. and R. Baldick (2016) "Multi-year stochastic generation capacity expansion planning under environmental energy policy," *Applied energy*, **183**, pp. 737–745.

[35] Liu, Y., R. Sioshansi, and A. J. Conejo (2017) "Multistage stochastic investment planning with multiscale representation of uncertainties and decisions," *IEEE Transactions on Power Systems*, **33**(1), pp. 781–791.

[36] Malcolm, S. and S. Zenios (1994) "Robust optimization for power capacity expansion planning," *Journal of the Operational Research Society*, **45**(9), pp. 1040–1049.

[37] Mulvey, J. M., R. J. Vanderbei, and S. A. Zenios (1995) "Robust optimization of large-scale systems," *Operations research*, **43**(2), pp. 264–281.

[38] Ben-Tal, A., L. El Ghaoui, and A. Nemirovski (2009) *Robust optimization*, vol. 28, Princeton University Press.

[39] Parkinson, S. C. and N. Djilali (2015) "Robust response to hydro-climatic change in electricity generation planning," *Climatic Change*, **130**(4), pp. 475–489.

[40] Franco, J. F., M. J. Rider, and R. Romero (2015) "Robust multi-stage substation expansion planning considering stochastic demand," *IEEE Transactions on Power Systems*, **31**(3), pp. 2125–2134.

[41] Verástegui, F., Á. Lorca, D. E. Olivares, M. Negrete-Pincetic, and P. Gazmuri (2019) "An adaptive robust optimization model for power systems planning with operational uncertainty," *IEEE Transactions on Power Systems*, **34**(6), pp. 4606–4616.

[42] Li, J., Z. Li, F. Liu, H. Ye, X. Zhang, S. Mei, and N. Chang (2017) "Robust coordinated transmission and generation expansion planning considering ramping requirements and construction periods," *IEEE Transactions on Power Systems*, **33**(1), pp. 268–280.

[43] Baringo, L. and A. Baringo (2017) "A stochastic adaptive robust optimization approach for the generation and transmission expansion planning," *IEEE Transactions on Power Systems*, **33**(1), pp. 792–802.

[44] Dehghan, S., N. Amjady, and A. J. Conejo (2015) "Reliability-constrained robust power system expansion planning," *IEEE Transactions on Power Systems*, **31**(3), pp. 2383–2392.

[45] Palmintier, B. and M. Webster (2011) "Impact of unit commitment constraints on generation expansion planning with renewables," in *2011 IEEE power and energy society general meeting*, IEEE, pp. 1–7.

[46] Jin, S., A. Botterud, and S. M. Ryan (2014) "Temporal versus stochastic granularity in thermal generation capacity planning with wind power," *IEEE Transactions on Power Systems*, **29**(5), pp. 2033–2041.

[47] Hua, B., R. Baldick, and J. Wang (2017) "Representing operational flexibility in generation expansion planning through convex relaxation of unit commitment," *IEEE Transactions on Power Systems*, **33**(2), pp. 2272–2281.

[48] Koltsaklis, N. E. and M. C. Georgiadis (2015) "A multi-period, multi-regional generation expansion planning model incorporating unit commitment constraints," *Applied energy*, **158**, pp. 310–331.

[49] Palmintier, B. S. and M. D. Webster (2015) "Impact of operational flexibility on electricity generation planning with renewable and carbon targets," *IEEE Transactions on Sustainable Energy*, **7**(2), pp. 672–684.

[50] Lara, C. L., J. D. Siirola, and I. E. Grossmann (2020) "Electric power infrastructure planning under uncertainty: stochastic dual dynamic integer programming (SDDiP) and parallelization scheme," *Optimization and Engineering*, **21**(4), pp. 1243–1281.

[51] Hobbs, B. F. (1995) "Optimization methods for electric utility resource planning," *European Journal of Operational Research*, **83**(1), pp. 1–20.

[52] Dagoumas, A. S. and N. E. Koltsaklis (2019) "Review of models for integrating renewable energy in the generation expansion planning," *Applied Energy*, **242**, pp. 1573–1587.

[53] Sadeghi, H., M. Rashidinejad, and A. Abdollahi (2017) "A comprehensive sequential review study through the generation expansion planning," *Renewable and Sustainable Energy Reviews*, **67**, pp. 1369–1394.

[54] Koltsaklis, N. E. and A. S. Dagoumas (2018) "State-of-the-art generation expansion planning: A review," *Applied energy*, **230**, pp. 563–589.

[55] Oree, V., S. Z. S. Hassen, and P. J. Fleming (2017) "Generation expansion planning optimisation with renewable energy integration: A review," *Renewable and Sustainable Energy Reviews*, **69**, pp. 790–803.

[56] Yalew, S. G., M. T. van Vliet, D. E. Gernaat, F. Ludwig, A. Miara, C. Park, E. Byers, E. De Cian, F. Piontek, G. Iyer, et al. (2020) "Impacts of climate change on energy systems in global and regional scenarios," *Nature Energy*, **5**(10), pp. 794–802.

[57] Fisher-Vanden, K., M. Webster, R. Lammers, V. Kumar, and J. Perla (2021) "Thresholds, Timing, and Teleconnections: A coupled water-power-economic analysis," .

[58] Kumar, V., K. Fisher-Vanden, M. Webster, R. Lammers, and J. Perla (2020), "Quantifying Extreme Event Impacts using a coupled electricity-economy model," .

[59] Xu, Q. and B. F. Hobbs (2019) "Value of model enhancements: quantifying the benefit of improved transmission planning models," *IET Generation, Transmission & Distribution*, **13**(13), pp. 2836–2845.

[60] Bukenberger, J. P. and M. D. Webster (2019) "Approximate Latent Factor Algorithm for Scenario Selection and Weighting in Transmission Expansion Planning," *IEEE Transactions on Power Systems*, **35**(2), pp. 1099–1108.

[61] Munoz, F. D., B. F. Hobbs, and J.-P. Watson (2016) "New bounding and decomposition approaches for MILP investment problems: Multi-area transmission and generation planning under policy constraints," *European Journal of Operational Research*, **248**(3), pp. 888–898.

[62] Liu, Y., R. Sioshansi, and A. J. Conejo (2017) "Hierarchical clustering to find representative operating periods for capacity-expansion modeling," *IEEE Transactions on Power Systems*, **33**(3), pp. 3029–3039.

[63] "PLEXOS software," `https://energyexemplar.com/solutions/plexos/`.

[64] Cole, W., M. Brown, K. Eurek, J. Becker, I. Chernyakhovskiy, S. Cohen, A. Frazier, P. Gagnon, N. Gates, D. Greer, et al. (2020) *Regional Energy Deployment System (ReEDS) Model Documentation: Version 2019*, *Tech. rep.*, National Renewable Energy Lab.(NREL), Golden, CO (United States).

[65] Papavasiliou, A. and S. S. Oren (2013) "Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network," *Operations research*, **61**(3), pp. 578–592.

[66] Kim, K., A. Botterud, and F. Qiu (2018) "Temporal decomposition for improved unit commitment in power system production cost modeling," *IEEE Transactions on Power Systems*, **33**(5), pp. 5276–5287.

[67] "PSO software," `http://psopt.com/`.

[68] Krishnan, V. and W. Cole (2016) "Evaluating the value of high spatial resolution in national capacity expansion models using ReEDS," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, IEEE, pp. 1–5.

[69] Ruiz, C. and A. J. Conejo (2015) "Robust transmission expansion planning," *European Journal of Operational Research*, **242**(2), pp. 390–401.

[70] Zhang, X. and A. J. Conejo (2017) "Robust transmission expansion planning representing long-and short-term uncertainty," *IEEE Transactions on Power Systems*, **33**(2), pp. 1329–1338.

[71] Hobbs, B. F., Q. Xu, J. Ho, P. Donohoo, S. Kasina, J. Ouyang, S. W. Park, J. Eto, and V. Satyal (2016) "Adaptive transmission planning: implementing a new paradigm for managing economic risks in grid expansion," *IEEE Power and Energy Magazine*, **14**(4), pp. 30–40.

[72] Lumbreras, S., A. Ramos, F. Banez-Chicharro, L. Olmos, P. Panciatici, C. Pache, and J. Maeght (2017) "Large-scale transmission expansion planning: from zonal results to a nodal expansion plan," *IET Generation, Transmission & Distribution*, **11**(11), pp. 2778–2786.

[73] Ahlhaus, P. and P. Stursberg (2013) "Transmission capacity expansion: an improved transport model," in *IEEE PES ISGT Europe 2013*, IEEE, pp. 1–5.

[74] Tuohy, A., P. Meibom, E. Denny, and M. O'Malley (2009) "Unit commitment for systems with significant wind penetration," *IEEE Transactions on power systems*, **24**(2), pp. 592–601.

[75] Chen, B. and L. Wang (2016) "Robust transmission planning under uncertain generation investment and retirement," *IEEE Transactions on Power Systems*, **31**(6), pp. 5144–5152.

[76] Castillo, A., C. Laird, C. A. Silva-Monroy, J.-P. Watson, and R. P. O'Neill (2016) "The unit commitment problem with AC optimal power flow constraints," *IEEE Transactions on Power Systems*, **31**(6), pp. 4853–4866.

[77] Bahrami, S. and V. W. Wong (2017) "Security-constrained unit commitment for ac-dc grids with generation and load uncertainty," *IEEE Transactions on Power Systems*, **33**(3), pp. 2717–2732.

[78] Zhang, H., G. T. Heydt, V. Vittal, and J. Quintero (2013) "An improved network model for transmission expansion planning considering reactive power and network losses," *IEEE Transactions on Power Systems*, **28**(3), pp. 3471–3479.

[79] Torbaghan, S. S., M. Gibescu, B. G. Rawn, and M. Van der Meijden (2014) "A market-based transmission planning for HVDC grid—case study of the North Sea," *IEEE Transactions on power systems*, **30**(2), pp. 784–794.

[80] Özdemir, Ö., F. D. Munoz, J. L. Ho, and B. F. Hobbs (2015) "Economic analysis of transmission expansion planning with price-responsive demand and quadratic losses by successive LP," *IEEE Transactions on Power Systems*, **31**(2), pp. 1096–1107.

[81] Cole, W., D. Greer, J. Ho, and R. Margolis (2020) "Considerations for maintaining resource adequacy of electricity systems with high penetrations of PV and storage," *Applied Energy*, **279**, p. 115795.

[82] Romero, R., A. Monticelli, A. Garcia, and S. Haffner (2002) "Test systems and mathematical models for transmission network expansion planning," *IEE Proceedings-Generation, Transmission and Distribution*, **149**(1), pp. 27–36.

[83] De Nooij, M., C. Koopmans, and C. Bijvoet (2007) "The value of supply security: The costs of power interruptions: Economic input for damage reduction and investment in networks," *Energy Economics*, **29**(2), pp. 277–295.

[84] Sullivan, M. J., M. Mercurio, and J. Schellenberg (2009) *Estimated value of service reliability for electric utility customers in the United States*, *Tech. rep.*, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States).

[85] Sanstad, A. H., Q. Zhu, B. Leibowicz, P. H. Larsen, and J. H. Eto (2020) "Case Studies of the Economic Impacts of Power Interruptions and Damage to Electricity System Infrastructure from Extreme Events," .

[86] Caron, J., S. M. Cohen, M. Brown, and J. M. Reilly (2018) "Exploring the impacts of a national US CO2 tax and revenue recycling options with a coupled electricity-economy model," *Climate Change Economics*, **9**(01), p. 1840015.

[87] Rausch, S. and M. Mowers (2014) "Distributional and efficiency impacts of clean and renewable energy standards for electricity," *Resource and Energy Economics*, **36**(2), pp. 556–585.

[88] Cohen, S. M. and J. Caron (2018) "The economic impacts of high wind penetration scenarios in the United States," *Energy Economics*, **76**, pp. 558–573.

[89] Böhringer, C. and T. F. Rutherford (2009) "Integrated assessment of energy policies: Decomposing top-down and bottom-up," *Journal of Economic Dynamics and Control*, **33**(9), pp. 1648–1661.

[90] Morales-España, G., J. M. Latorre, and A. Ramos (2013) "Tight and compact MILP formulation for the thermal unit commitment problem," *IEEE Transactions on Power Systems*, **28**(4), pp. 4897–4908.

[91] Munoz, F. D., E. E. Sauma, and B. F. Hobbs (2013) "Approximations in power transmission planning: implications for the cost and performance of renewable portfolio standards," *Journal of Regulatory Economics*, **43**(3), pp. 305–338.

[92] Grogan, D. S. (2016) *Global and regional assessments of unsustainable groundwater use in irrigated agriculture*, Ph.D. thesis, University of New Hampshire.

[93] Wisser, D., B. M. Fekete, C. Vörösmarty, and A. Schumann (2010) "Reconstructing 20th century global hydrography: a contribution to the Global Terrestrial Network-Hydrology (GTN-H)," *Hydrology and Earth System Sciences*, **14**(1), pp. 1–24.

[94] Stewart, R. J., W. M. Wollheim, A. Miara, C. J. Vörösmarty, B. Fekete, R. B. Lammers, and B. Rosenzweig (2013) "Horizontal cooling towers: riverine ecosystem services and the fate of thermoelectric heat in the contemporary Northeast US," *Environmental Research Letters*, **8**(2), p. 025010.

[95] Dingman, S. L. (1972) "Equilibrium temperatures of water surfaces as related to air temperature and solar radiation," *Water Resources Research*, **8**(1), pp. 42–49.

[96] Rienecker, M. M., M. J. Suarez, R. Gelaro, R. Todling, J. Bacmeister, E. Liu, M. G. Bosilovich, S. D. Schubert, L. Takacs, G.-K. Kim, et al. (2011) "MERRA: NASA's modern-era retrospective analysis for research and applications," *Journal of climate*, **24**(14), pp. 3624–3648.

[97] Böhringer, C., T. F. Rutherford, and W. Wiegard (2003) "Computable general equilibrium analysis: Opening a black box," .

[98] Bergman, L. (1991) "General equilibrium effects of environmental policy: a CGE-modeling approach," *Environmental and Resource Economics*, **1**(1), pp. 43–61.

[99] Armington, P. S. (1969) "A theory of demand for products distinguished by place of production," *Staff Papers*, **16**(1), pp. 159–178.

[100] Park, S., Q. Xu, and B. F. Hobbs (2019) "Comparing scenario reduction methods for stochastic transmission planning," *IET Generation, Transmission & Distribution*, **13**(7), pp. 1005–1013.

[101] Shayesteh, E., B. F. Hobbs, and M. Amelin (2016) "Scenario reduction, network aggregation, and DC linearisation: which simplifications matter most in operations and planning optimisation?" *IET Generation, Transmission & Distribution*, **10**(11), pp. 2748–2755.

[102] Shi, D. (2012) *Power system network reduction for engineering and economic analysis*, Arizona State University.

[103] "System adequacy planning datasets. Western Electricity Coordinating Council. (2012)," `https://www.wecc.biz:443/SystemAdequacyPlanning/Pages/Datasets.aspx`.

[104] "2019 integrated study program. Technical report, WECC Reliability Assessment Committee, Western Electricity Coordinating Council (2019)." `https://www.wecc.org/Reliability/2019\%20Integrated\%20Study\%20Program.pdf`.

[105] "3. Emissions generation resource integrated database. U.S. Environmental Protection Agency. (2020);," `https://www.epa.gov/egrid/emissions-generation-resource-integrated-database-egrid`.

[106] "IMPLAN Group, LLC. IMPLAN 2010. Huntersville, NC (2010," `https://implan.com/`.

[107] Birge, J. R. and F. Louveaux (2011) *Introduction to Stochastic Programming, Second Edition*, Springer.

[108] Van Slyke, R. M. and R. Wets (1969) "L-shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal on Applied Mathematics*, **17**(4), pp. 638–663.

[109] Rockafellar, R. T. and R. J.-B. Wets (1991) "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of operations research*, **16**(1), pp. 119–147.

[110] Zou, J., S. Ahmed, and X. A. Sun (2019) "Stochastic dual dynamic integer programming," *Mathematical Programming*, **175**(1-2), pp. 461–502.

[111] Bellman, R. E. and S. E. Dreyfus (2015) *Applied dynamic programming*, Princeton university press.

[112] Puterman, M. L. (2014) *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons.

[113] Lewis, F. L., D. Vrabie, and V. L. Syrmos (2012) *Optimal control*, John Wiley & Sons.

[114] Powell, W. B. (2007) *Approximate Dynamic Programming: Solving the curses of dimensionality*, vol. 703, John Wiley & Sons.

[115] Sutton, R. S. and A. G. Barto (2018) *Reinforcement learning: An introduction*, MIT press.

[116] Watkins, C. J. and P. Dayan (1992) "Q-learning," *Machine learning*, **8**(3-4), pp. 279–292.

[117] Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996) "Reinforcement learning: A survey," *Journal of artificial intelligence research*, **4**, pp. 237–285.

[118] Bertsekas, D. P. (2012) *Approximate dynamic programming*, Athena scientific Belmont.

[119] Watkins, C. J. C. H. (1989) "Learning from delayed rewards," .

[120] Rosenbluth, M. N. and A. W. Rosenbluth (1955) "Monte Carlo calculation of the average extension of molecular chains," *The Journal of Chemical Physics*, **23**(2), pp. 356–359.

[121] Hammersley, J. M. and K. W. Morton (1954) "Poor man's monte carlo," *Journal of the Royal Statistical Society: Series B (Methodological)*, **16**(1), pp. 23–38.

[122] Infanger, G. (1992) "Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs," *Annals of Operations Research*, **39**(1), pp. 69–95.

[123] Parpas, P., B. Ustun, M. Webster, and Q. K. Tran (2015) "Importance sampling in stochastic programming: A Markov chain Monte Carlo approach," *INFORMS Journal on Computing*, **27**(2), pp. 358–377.

[124] Precup, D. (2000) "Eligibility traces for off-policy policy evaluation," *Computer Science Department Faculty Publication Series*, p. 80.

[125] Precup, D., R. S. Sutton, and S. Dasgupta (2001) "Off-policy temporal-difference learning with function approximation," in *ICML*, pp. 417–424.

[126] Brafman, R. I. and M. Tennenholtz (2002) "R-max-a general polynomial time algorithm for near-optimal reinforcement learning," *Journal of Machine Learning Research*, **3**(Oct), pp. 213–231.

[127] Kearns, M. and S. Singh (2002) "Near-optimal reinforcement learning in polynomial time," *Machine learning*, **49**(2-3), pp. 209–232.

[128] Gehring, C. and D. Precup (2013) "Smart exploration in reinforcement learning using absolute temporal difference errors," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1037–1044.

[129] Strehl, A. L. and M. L. Littman (2008) "An analysis of model-based interval estimation for Markov decision processes," *Journal of Computer and System Sciences*, **74**(8), pp. 1309–1331.

[130] Singh, S., R. L. Lewis, A. G. Barto, and J. Sorg (2010) "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development*, **2**(2), pp. 70–82.

[131] Bellemare, M., S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos (2016) "Unifying count-based exploration and intrinsic motivation," in *Advances in neural information processing systems*, pp. 1471–1479.

[132] Martin, J., S. N. Sasikumar, T. Everitt, and M. Hutter (2017) "Count-based exploration in feature space for reinforcement learning," *arXiv preprint arXiv:1706.08090*.

[133] Ostrovski, G., M. G. Bellemare, A. v. d. Oord, and R. Munos (2017) "Count-based exploration with neural density models," *arXiv preprint arXiv:1703.01310*.

[134] Tang, H., R. Houthooft, D. Foote, A. Stooke, O. X. Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel (2017) "# exploration: A study of count-based exploration for deep reinforcement learning," in *Advances in neural information processing systems*, pp. 2753–2762.

[135] Silver, E. A. (1963) *Markovian decision processes with uncertain transition probabilities or rewards*, *Tech. rep.*, MASSACHUSETTS INST OF TECH CAMBRIDGE OPERATIONS RESEARCH CENTER.

[136] Duff, M. O. and A. G. Barto (1997) "Local bandit approximation for optimal learning problems," in *Advances in Neural Information Processing Systems*, pp. 1019–1025.

[137] Dearden, R., N. Friedman, and S. Russell (1998) "Bayesian Q-learning," in *Aaai/iaai*, pp. 761–768.

[138] Ryzhov, I. O. and W. B. Powell (2010) "Approximate dynamic programming with correlated bayesian beliefs," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, pp. 1360–1367.

[139] Ryzhov, I. O., M. R. Mes, W. B. Powell, and G. van den Berg (2019) "Bayesian exploration for approximate dynamic programming," *Operations research*, **67**(1), pp. 198–214.

[140] Sutton, R. S., D. A. McAllester, S. P. Singh, and Y. Mansour (2000) "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, pp. 1057–1063.

[141] Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller (2014) "Deterministic policy gradient algorithms," .

[142] Simmons-Edler, R., B. Eisner, E. Mitchell, S. Seung, and D. Lee (2019) "Q-learning for continuous actions with cross-entropy guided policies," *arXiv preprint arXiv:1903.10605*.

[143] Kalashnikov, D., A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. (2018) "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*.

[144] Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015) "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*.

[145] Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine (2018) "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*.

[146] Gu, S., T. Lillicrap, I. Sutskever, and S. Levine (2016) "Continuous deep q-learning with model-based acceleration," in *International Conference on Machine Learning*, pp. 2829–2838.

[147] Amos, B., L. Xu, and J. Z. Kolter (2017) "Input convex neural networks," in *International Conference on Machine Learning*, PMLR, pp. 146–155.

[148] Ryu, M., Y. Chow, R. Anderson, C. Tjandraatmadja, and C. Boutilier (2019) "Caql: Continuous action q-learning," *arXiv preprint arXiv:1909.12397*.

[149] Mahmood, A. R., H. P. van Hasselt, and R. S. Sutton (2014) "Weighted importance sampling for off-policy learning with linear function approximation," in *Advances in Neural Information Processing Systems*, pp. 3014–3022.

[150] Hachiya, H., T. Akiyama, M. Sugiayma, and J. Peters (2009) "Adaptive importance sampling for value function approximation in off-policy reinforcement learning," *Neural Networks*, **22**(10), pp. 1399–1410.

[151] Sutton, R. S., A. R. Mahmood, and M. White (2016) "An emphatic approach to the problem of off-policy temporal-difference learning," *The Journal of Machine Learning Research*, **17**(1), pp. 2603–2631.

[152] Jiang, N. and L. Li (2016) "Doubly robust off-policy value evaluation for reinforcement learning," in *International Conference on Machine Learning*, PMLR, pp. 652–661.

[153] Rowland, M., A. Harutyunyan, H. Hasselt, D. Borsa, T. Schaul, R. Munos, and W. Dabney (2020) "Conditional importance sampling for off-policy learning," in *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 45–55.

[154] Guo, Z., P. S. Thomas, and E. Brunskill (2017) "Using options and covariance testing for long horizon off-policy policy evaluation," in *Advances in Neural Information Processing Systems*, pp. 2492–2501.

[155] Xie, T., U. Amehrst, Y.-X. Wang, Y. Ma, and A. Amazon (2018) "Marginalized Off-Policy Evaluation for Reinforcement Learning," in *NeurIPS 2018 Workshop on Causal Learning*.

[156] Gelada, C. and M. G. Bellemare (2019) "Off-policy deep reinforcement learning by bootstrapping the covariate shift," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3647–3655.

[157] Hanna, J., S. Niekum, and P. Stone (2019) "Importance sampling policy evaluation with an estimated behavior policy," in *International Conference on Machine Learning*, PMLR, pp. 2605–2613.

[158] Xie, T., Y. Ma, and Y.-X. Wang (2019) "Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling," in *Advances in Neural Information Processing Systems*, pp. 9668–9678.

[159] Mahmood, A. R., H. Yu, and R. S. Sutton (2017) "Multi-step off-policy learning without importance sampling ratios," *arXiv preprint arXiv:1702.03006*.

[160] Kroese, D., T. Taimre, and Z. Botev (2011) *Handbook of Monte Carlo Methods*.

[161] Ho, J., B. F. Hobbs, P. Donohoo-Vallett, Q. Xu, S. Kasina, S. Park, and Y. Ouyang (2016) "Planning transmission for uncertainty: Applications and lessons for the western interconnection," *The Western Electricity Coordinating Council*.

[162] Singh, S., T. Jaakkola, M. L. Littman, and C. Szepesvári (2000) "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine learning*, **38**(3), pp. 287–308.

[163] Bellman, R. (1966) "Dynamic programming," *Science*, **153**(3731), pp. 34–37.

[164] Olsen, P. (1976) "Multistage stochastic programming with recourse: The equivalent deterministic problem," *SIAM Journal on Control and Optimization*, **14**(3), pp. 495–517.

[165] Wets, R. (1972) "Stochastic programs with recourse: A basic theorem for multistage problems," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, **21**(3), pp. 201–206.

[166] Birge, J. R. (1985) "Decomposition and partitioning methods for multistage stochastic linear programs," *Operations research*, **33**(5), pp. 989–1007.

[167] Wets, R. J. (1989) "The aggregation principle in scenario analysis and stochastic optimization," in *Algorithms and model formulations in mathematical programming*, Springer, pp. 91–113.

[168] Bertsekas, D. P. (1976) "Multiplier methods: A survey," *Automatica*, **12**(2), pp. 133–145.

[169] Rockafellar, R. T. (1976) "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of operations research*, **1**(2), pp. 97–116.

[170] Pereira, M. V. and L. M. Pinto (1991) "Multi-stage stochastic optimization applied to energy planning," *Mathematical programming*, **52**(1), pp. 359–375.

[171] Shapiro, A. (2011) "Analysis of stochastic dual dynamic programming method," *European Journal of Operational Research*, **209**(1), pp. 63–72.

[172] Rebennack, S., B. Flach, M. V. Pereira, and P. M. Pardalos (2011) "Stochastic Hydro-Thermal Scheduling Under CO2 Emissions Constraints," *IEEE Transactions on Power Systems*, **27**(1), pp. 58–68.

[173] Cerisola, S., J. M. Latorre, and A. Ramos (2012) "Stochastic dual dynamic programming applied to nonconvex hydrothermal models," *European Journal of Operational Research*, **218**(3), pp. 687–697.

[174] Bareilles, G., Y. Laguel, D. Grishchenko, F. Iutzeler, and J. Malick (2020) "Randomized Progressive Hedging methods for multi-stage stochastic programming," *Annals of Operations Research*, **295**(2), pp. 535–560.

[175] Gul, S., B. T. Denton, and J. W. Fowler (2015) "A progressive hedging approach for surgery planning under uncertainty," *INFORMS Journal on Computing*, **27**(4), pp. 755–772.

[176] Astaraky, D. and J. Patrick (2015) "A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling," *European Journal of Operational Research*, **245**(1), pp. 309–319.

[177] Bukenberger, J. and B. Palmintier (2018) "Stochastic generation capacity expansion planning with approximate dynamic programming," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, IEEE, pp. 1–5.

[178] Soares, M. P., A. Street, and D. M. Valladão (2017) "On the solution variability reduction of stochastic dual dynamic programming applied to energy planning," *European Journal of Operational Research*, **258**(2), pp. 743–760.

[179] Zhang, D. and D. Adelman (2009) "An approximate dynamic programming approach to network revenue management with customer choice," *Transportation Science*, **43**(3), pp. 381–394.

[180] Ryan, S. M., R. J.-B. Wets, D. L. Woodruff, C. Silva-Monroy, and J.-P. Watson (2013) "Toward scalable, parallel progressive hedging for stochastic unit commitment," in *2013 IEEE Power & Energy Society General Meeting*, IEEE, pp. 1–5.

[181] Guo, G., G. Hackebeil, S. M. Ryan, J.-P. Watson, and D. L. Woodruff (2015) "Integration of progressive hedging and dual decomposition in stochastic integer programs," *Operations Research Letters*, **43**(3), pp. 311–316.

[182] Helgason, T. and S. W. Wallace (1991) "Approximate scenario solutions in the progressive hedging algorithm," *annals of Operations Research*, **31**(1), pp. 425–444.

[183] Crainic, T. G., M. Hewitt, and W. Rei (2014) "Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design," *Computers & Operations Research*, **43**, pp. 90–99.

[184] Haugen, K. K., A. Løkketangen, and D. L. Woodruff (2001) "Progressive hedging as a meta-heuristic applied to stochastic lot-sizing," *European Journal of Operational Research*, **132**(1), pp. 116–122.

[185] Rebennack, S. (2016) "Combining sampling-based and scenario-based nested Benders decomposition methods: application to stochastic dual dynamic programming," *Mathematical Programming*, **156**(1-2), pp. 343–389.

[186] De Matos, V. L., A. B. Philpott, and E. C. Finardi (2015) "Improving the performance of stochastic dual dynamic programming," *Journal of Computational and Applied Mathematics*, **290**, pp. 196–208.

[187] Guigues, V. (2020) "Inexact cuts in stochastic dual dynamic programming," *SIAM Journal on Optimization*, **30**(1), pp. 407–438.

[188] De Farias, D. P. and B. Van Roy (2004) "On constraint sampling in the linear programming approach to approximate dynamic programming," *Mathematics of operations research*, **29**(3), pp. 462–478.

[189] Desai, V. V., V. F. Farias, and C. C. Moallemi (2012) "Approximate dynamic programming via a smoothed linear program," *Operations Research*, **60**(3), pp. 655–674.

[190] Kumar, V. and M. Webster (2021) "Importance Sampling based Exploration in Q Learning," *arXiv preprint arXiv:2107.00602*.

[191] Shapiro, A., W. Tekaya, J. P. da Costa, and M. P. Soares (2011), "Report for technical cooperation between georgia institute of technology and ons-operador nacional do sistema elétrico," .

[192] Morris, J., V. Srikrishnan, M. Webster, and J. Reilly (2018) "Hedging strategies: electricity investment decisions under policy uncertainty," *The Energy Journal*, **39**(1).

Vita

Vijay Kumar

Vijay Kumar was born and raised in Pune, India. He completed his bachelor's in mechanical engineering from Visvesvaraya National Institute of Technology, Nagpur, in 2014. After completing his bachelor's, he worked in Bajaj Auto limited, an automobile company, as a Senior Engineering in the Planning and Procurement Department. He is currently a Ph.D. candidate in the Industrial Engineering and Operations Research department at Pennsylvania State University. His primary interest lies in developing tools for emerging applications with a focus on optimization and data science.