THESIS

STOCHASTIC OPTIMIZATION ON HYBRID RENEWABLE ENERGY SYSTEMS

Submitted by

Nouralden Mohammed Jadalla Mohammed

School of Computer Science and Applied Mathematics

In fulfillment of the requirements

For the Degree of Doctor of Philosophy

University of the Witwatersrand,

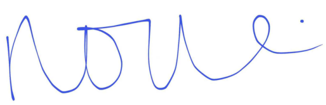
Johannesburg, South Africa

March 2023

Supervisor: Professor M Montaz Ali

DECLARATION

As the undersigned here, I declare that this thesis and its contents are my own original work, except where citations are made. It is being submitted for the degree of Doctor of Philosophy at the University of the Witwatersrand, Johannesburg. This work has never been submitted before to any other university for an academic degree or examination.



Nouralden Mohammed Jadalla Mohammed

May 26, 2023.

ABSTRACT

STOCHASTIC OPTIMIZATION ON HYBRID RENEWABLE ENERGY SYSTEMS

This research explores the use of stochastic optimization in finding the optimal size of hybrid renewable energy systems that combine two or more renewable energy sources. The goal is to achieve an optimal balance between system size, cost, and reliability. We have used mathematical models and algorithms to find the optimal sizing of the system by minimizing costs while ensuring reliability. Specifically, we propose an optimal-sized hybrid renewable energy system that uses stochastic programming to minimize the cost of the system based on the reliability percentage. We have formulated the hybrid renewable energy sizing problem using two-stage stochastic programming and developed an algorithm using the three-block Alternating Direction Method of Multipliers (ADMM) to solve this problem. The theoretical convergence of the proposed algorithm was established, and the results were compared with state-of-the-art methods in the literature, demonstrating the superior performance of our proposed approach. Also, we have applied the proposed algorithm to a rural area in South Africa- as a case study- to find the optimal sizing of the hybrid renewable system for the geographic location. Moreover, we have developed a differential equations-based model for the deterministic version of the stochastic programming problem for optimal sizing. This model enables the design of the renewable system as we progress in time instead of building the entire system at the beginning of the project. The dynamical-based model has shown superiority over the three-block ADMM in terms of capital cost, where it decreased the capital costs of the renewable components by a significant amount. Finally, we have incorporated machine learning techniques to model wind turbine power curves and solar power predictions using climatic data. We have found that ensemble models outperformed parametric models by having better accuracy. Overall, this research provides new insights into the optimal sizing of hybrid re-

newable energy systems and presents innovative algorithms and techniques for their design and optimization.

Keywords: Hybrid Renewable Energy, Two-stage Stochastic Programming, Alternating Direction Method of Multipliers, Progressive Hedging, Pattern Search, Differential Equations-based Model

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Allah for providing me with the strength, guidance, and blessings to complete my Ph.D. thesis. The journey was challenging and at times arduous, but with divine support, I was able to overcome the obstacles and reach this remarkable accomplishment.

I would also like to extend my heartfelt appreciation by expressing my deep and sincere gratitude to my supervisor, Professor Montaz Ali, who was more than just a mentor, but a source of inspiration and support throughout my Ph.D. journey. His unwavering generosity, insightful recommendations, and invaluable guidance played a crucial role in the successful completion of this thesis. Professor Ali's dedication to his students and passion for his work is truly remarkable, and I am honored to have had the opportunity to work under his supervision.

My sincere gratitude also goes to my parents, wife, brothers, and sister for their unwavering support and encouragement. Their love, understanding, and belief in me have been a constant source of motivation, and I cannot thank them enough for their tremendous support over the past few years. It would not have been possible for me to complete this journey without their love and support.

Last but not least, I would like to acknowledge the tremendous support of Wits University. The university provided me with the facilities and resources that I needed to conduct my research and created an academic atmosphere that was not only challenging but also motivating. The opportunities and experiences I gained during my time at Wits University will be invaluable to me in the future, and I am grateful for the support they provided me throughout my Ph.D. journey.

Dedicated to my parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SP Stochastic Programming

LP Linear Programming

SMPS Stochastic Mathematical Programming System

ADMM Alternating Direction Method of Multipliers

PHA Progressive Hedging Algorithm

kW Kilowatt

PV Photovoltaic

FC Fuel Cell

SoC SoCState of Charge

LPSP Loss of Power Supply Probability

LLP Loss of Load Probability

MSE Mean Squared Error

RMSE Rooted Mean Squared Error

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

CV Cross-Vaidation

RNN Recurrent Neural Networks

LSTM Long Short-Term Memory

XGB Extreme Gradient Boosting

GBM Gradient Boosting Regression

DTR Decision Tree Regressor

RF Random Forest

SDG Sustainable Development Goal

CPU Central Processing Unit

GA Aenetic Algorithm

ACO Ant Colony Optimization

PSO Paricale Swarm Optimization

LASSO Least Absolute Shrinkage and Selection Operator

SVM Support Vector Machine

SGD Stochastic Gradient Descent

KL Kurdyka-Lojasiewicz

LCE Levelized Cost of Energy

HOMER Hybrid Optimization of Multiple Resources

m Meter

s Second

°**C** Celsius

STC Standard Temperature Conditions

Chapter 1

Introduction

Electricity is a fundamental necessity of modern life, and its importance cannot be overstated. It is a crucial aspect of daily life in developed and developing countries, powering everything from residential homes to gigantic industrial and governmental facilities. The availability of reliable electricity is often considered a hallmark of modern society and a crucial aspect of speeding up economic growth and development.

The significance of electricity is evident in its widespread use across various sectors. It powers homes, businesses, and industries, enabling people to carry out their daily activities with ease. From communication systems to transportation networks, electricity is a crucial part of modern life. In industries, it plays a crucial role in powering machinery and equipment, enabling companies to operate efficiently and produce goods and services that meet the demands of consumers.

However, as the world's population continues to grow, so does the electricity demand. This rapid growth of populations and industries creates an increased demand for energy, which puts pressure on the existing power infrastructure. Countries worldwide are investing in renewable energy sources to provide cleaner and more sustainable energy. Additionally, they are investing in research and development of new technologies that can improve the efficiency and reliability of energy generation and distribution in response to the growing demand for energy, particularly with the increase in population and economic growth. With its increasing importance, countries must continue to invest in their energy infrastructure to ensure that their citizens have access to reliable and sustainable sources of electricity.

Generally, generating electricity can be achieved through two main methods: conventional and renewable energy. The first type, fuel-based energy generation, relies on using finite resources such as coal, natural gas, and oil to generate electricity. It is popular because of its deterministic nature and the ability to meet the power demand. The fact that capacities of conventional generators are well known to electricity companies. This reality allows companies to accurately calculate the percentage of the load that fuel-based generators can satisfy, reducing the risk of power shortages.

However, conventional energy generation has several significant drawbacks. The primary disadvantage is its dependence on finite fossil fuels, which are not only depleting but also unevenly distributed around the world. This burden creates a dependency on foreign sources of fuel and increases the risk of price volatility and supply interruptions. In addition, conventional energy generation contributes to the emission of greenhouse gases (GHGs), which have a detrimental effect on the environment and climate change.

On the other hand, renewable energy relies on natural environmental resources such as the sun, wind, water, and geothermal energy to generate electricity. This method is becoming increasingly popular due to its ability to reduce dependency on finite fossil fuels and its potential to reduce greenhouse gas emissions.

The main advantage of renewable energy is that it uses renewable resources that are widely distributed, making it a more sustainable solution to energy generation. However, renewable energy also has its limitations. For example, the availability of renewable resources can depend on local weather conditions, which might make electricity generation less predictable. In addition, renewable energy sources can also be more expensive to install and maintain than conventional energy sources, making the switch to renewables a more significant investment for electricity companies. Both conventional and renewable energy methods have their advantages and disadvantages, and the choice of method will depend on many factors, such as local resources, cost, and environmental impact.

GHG emissions from fossil fuels are the primary cause of global warming and its associated environmental impacts. Conventional power generation methods using fossil fuels are highly reliable in meeting the global electricity demand but have been responsible for a significant increase in GHG emissions. The emissions of these toxic gases have continued to rise, and according to the Intergovernmental Panel on Climate Change (IPCC), they increased by 70% from 1970 to 2004 [7]. The increased concentration of these gases in the atmosphere has led to the warming of the

Earth's surface, causing significant environmental problems such as ice melting, forest fires, and heat waves.

In addition to the negative environmental impacts, conventional power generation methods also contribute to air pollution, which has serious health consequences for humans and wildlife. Burning fossil fuels emits harmful particles and chemicals into the atmosphere, which can lead to respiratory problems and raise the likelihood of heart and lung diseases. Given these consequences, there is a growing need to transition to more sustainable and renewable energy sources, such as solar and wind power, to reduce the dependence on fossil fuels and mitigate the impact of GHG emissions on the environment. While renewable sources may not yet be as reliable or cost-effective as conventional power generation methods, advances in technology and investment in infrastructure will make it possible to meet the increasing demand for energy while reducing the negative impact on the planet.

Fossil fuels contribute to GHG emissions, and the emission varies from method to method due to the dependency on the chemical reaction on the generator side. The emissions vary depending on the specific method of power generation and the type of fossil fuel used. For example, coal-fired power plants emit the most GHGs, followed by natural gas-fired power plants, and oil-fired power plants emit the least. This variability is due to the different chemical reactions that occur during electricity generation. These emissions are known to be harmful to the environment and contribute to global warming, air pollution, and ocean acidification. The emissions that occur from chemical reactions are considered environmentally harmful.

Although the initial construction expenses for conventional fossil fuel power plants are reasonable from an economic perspective, the operating costs are relatively high compared to renewable energy alternatives. This is due to the constant need for fuel to keep the generator functioning. The fuel cost can be unpredictable and can vary greatly, leading to higher operating costs over time. In contrast, renewable energy sources, such as wind and solar power, do not emit GHGs and have much lower operating costs. They also have the potential to reduce dependence on finite fossil fuels and help to mitigate climate change. Switching to renewable energy sources is becoming increasingly economically viable, and many countries are setting targets to phase out fossil fuels and transition to cleaner forms of energy.

South Africa, with its expanding economy and population, has experienced a steady rise in GHG emissions over the past three decades [8]. This increase is reflected in the GHG emissions shown in Figure 1.1, where the horizontal axis represents the years, and the vertical axis denotes the kiloton (kt) of CO_2 equivalent and the annual percentage of change. From 1990 to 2019, the country's total GHG emissions rose from 331 million metric tons of CO2 equivalent to 555 million metric tons of CO2 equivalent, representing a 67% increase. This increase in emissions has significant implications for the environment and the global effort to mitigate the effects of climate change in South Africa.

The South African government has committed to addressing this issue by transitioning the country's energy sector toward a low-carbon economy. This transition involves a focus on renewable energy sources such as wind and solar power, as well as a decrease in the use of coal, which is a significant contributor to GHG emissions. By reducing the country's reliance on fossil fuels, the government hopes to reduce GHG emissions and contribute to the global effort to mitigate the effects of climate change.

The increase in GHG emissions in South Africa over the past 30 years is a big concern, and the government's commitment to transitioning the country's energy sector towards a low-carbon economy is a critical step in the right direction. To achieve these goals locally and globally, the optimal sizing of hybrid renewable energy systems is essential. This refers to the process of determining the most efficient combination of different renewable energy sources to meet energy demands in an environmentally responsible and cost-effective manner. By carefully considering the relative strengths and limitations of different renewable energy technologies, governments, and investors can ensure that it is making the most of the available resources and minimizing the impact on the environment.

Under these circumstances, scientists and engineers have been working hard to find alternative methods to generate electricity effectively and reliably to meet the demand and keep the envi-

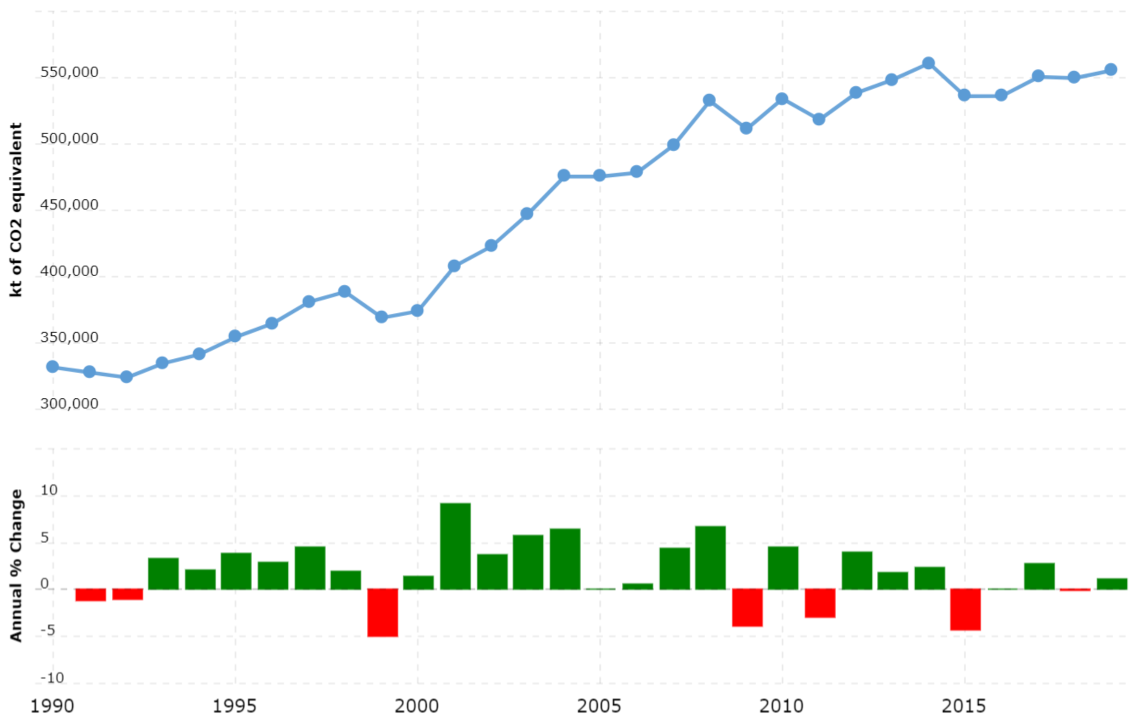


Figure 1.1: South Africa Greenhouse Gas (GHG) Emissions 1990-2019

ronment unharmed. The innovative approaches should minimize greenhouse gas emissions to an acceptable level and accommodate the future growth of populations and, thus, related human activities.

Renewable sources, like solar, wind, and hydro, are considered a well-suited solution to power generation since they enjoy desirable features. Renewable energy generation methods use free natural resources that are not depletable. Especially, Africa is regarded as the Sun continent; because it receives, on average, the highest amount of daily sunlight than other parts of the world. Although there is enormous solar potential in Africa, the development of solar power in Africa is deficient [9].

The generated power by renewable sources is clean and environmentally friendly because there is no contribution of greenhouse gas emissions which ensures the 7th goal of Sustainable Development Goals (SDGs) —environmental sustainability. The installation costs of renewable systems are considered high, but the operating costs are much less than the conventional methods in long-term projects. The main factors that affect electricity production by renewable sources are weather conditions and seasonal cycles. For example, solar power is not produced effectively during cloudy days and none at night. Renewable methods can fully replace conventional methods in the short term for the domestic generation with the cooperation of backup generators.

Many countries are preparing to achieve 100% renewable energy in the near future. For example, Iceland gets all of its electricity from renewable energy sources such as geothermal and hydroelectric. Costa Rica is also paving the route for 100% renewable power production. For 300 days consecutively, Costa Rica satisfied all its electrical demands with renewable energy sources such as hydropower, geothermal, biomass, wind, and solar. Kenya, Namibia, Norway, Ethiopia, and Uruguay are among the countries that now generate more than 90% of their power from renewable sources [10].

Using one renewable energy source in generating electricity may affect system reliability due to weather fluctuation. Thus, hybridizing two or more sources, possibly with backup sources like a conventional generator and batteries, will enhance system reliability and reduce dependency on storage capacity. This study aims to design an economic and optimal integrated hybrid renewable energy system model that will be reliable and environmentally sustainable.

We have taken two different approaches to solve the problem. The first approach to tackling the problem involves modeling it as a two-stage stochastic programming problem. Upon formulating the stochastic programming problem, we transform the deterministic version of the model into a finite-dimensional optimization problem. This step involves identifying the appropriate set of constraints and variables that will be used to create an objective function that optimizes a particular outcome.

We have solved the model using the state-of-art alternating direction method of multipliers (ADMM) and applied it to a case study in a rural area in South Africa, Limpopo province, Ga-Noane. The proposed ADMM algorithm allows parallel computation in finding the optimal solution. Parallel computation is crucial in solving real-world problems instead of serial computing to save time and space. We have compared our procedure to an energy problem solved by Sakalauskas and Zilinskas [5] using two-stage Stochastic Programming (SP) with the Monte Carlo method. The comparison has shown that our approach is superior in terms of CPU time and the number of iterations. Also, we have proved the theoretical convergence of the three-block ADMM approach to the optimal solution under certain conditions.

Our second strategy involves transforming the classical finite-dimensional optimization problem into a dynamic optimization problem based on differential equations. By using this approach, we can gradually construct an optimal sizing solution over time. We have compared the results of this approach with those of the first strategy.

1.1 Problem Statement and Motivation

Hybrid renewable energy systems are considered an adequate means for energy solutions in the future, especially for rural areas where expanding the grid is too costly. Most of the current energy systems depend heavily on fuel consumption, which generates greenhouse gas emissions that cause severe environmental impact. Renewable energy is environmentally friendly that uses natural resources to generate electricity and power. Most hybrid systems use solar and wind resources that rely on uncertain weather conditions, thus, uncertain power generation. Further, the load demand is also uncertain, and it is difficult to meet using fluctuating resources without proper sizing of the system's components. Since most of these types of systems are standalone systems used to electrify rural off-grid areas, thus the design of such systems must be in optimal conditions.

The research on modeling optimal sizing mathematical algorithms has used commercial software intensively to size hybrid renewable systems. Few researchers have utilized detailed algorithms to solve those particular types of problems. To the best of our best knowledge, no research has been conducted using the three-block Alternating Direction Method of Multiplier (ADMM) to solve two-stage Stochastic Programming (SP) in hybrid renewable energy system sizing.

The problem, therefore, is to design an optimal sizing system of hybrid renewable energy that ensures system reliability using SP based on three-block ADMM and the concept of Loss of Power Supply Probability (LPSP). Such eco-friendly systems seek to minimize the effect of greenhouse gas emissions by using entire renewable sources. Figure 1.2 represents a possible flow chart of the optimal sizing process for the hybrid renewable energy system (HRES) components.

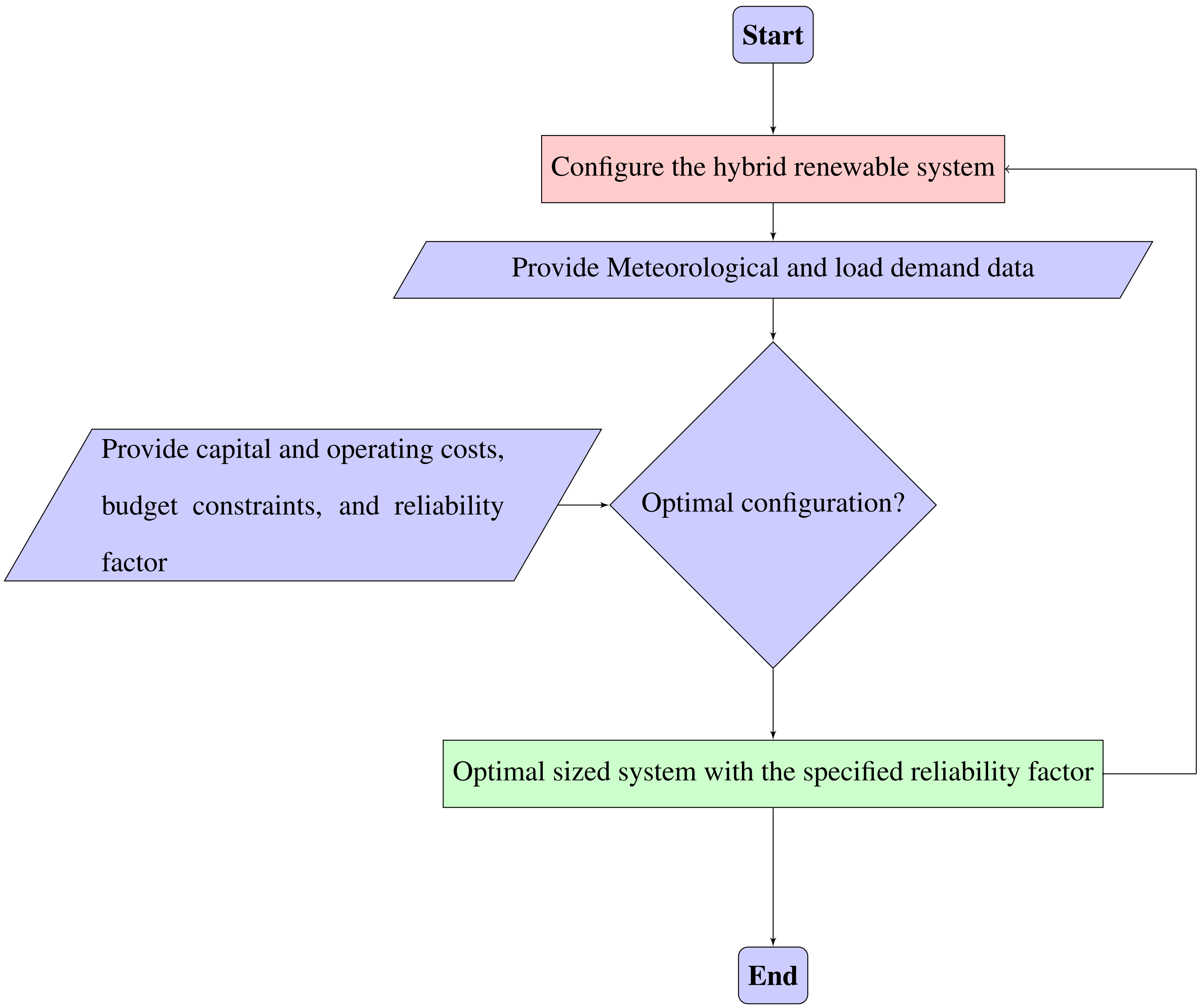


Figure 1.2: Flow chart of optimized process of HRES

Our second model is constructed on differential equations (DE) based dynamical systems, which means that solving DEs is necessary for evaluating the objective function. We have used a pattern search optimization technique after modeling the problem using DEs to obtain the optimal sizing and associated costs for the hybrid renewable system.

1.2 Aim and Objectives

This research aims to design an integrated hybrid renewable energy system in rural areas taking into account reliability and environmental sustainability.

Achieving the above aim can be done by the following objectives:

1. Developing a machine learning scheme that predicts solar and wind power output based on climatic data, enabling assessment of potential power output in a specific geographic location.

2. Developing a two-stage stochastic programming mathematical model for sizing hybrid renewable energy systems, ensuring optimal system design.

3. Designing a three-block ADMM-based optimization algorithm for solving the two-stage stochastic problem in the mathematical model, facilitating efficient and accurate solutions.

4. Providing a convergence proof for the three-block ADMM algorithm, subject to suitable assumptions, validating its reliability and effectiveness.

5. Modeling a differential equation-based dynamical optimization model for solving the two-stage stochastic problem using pattern search, offering an alternative approach to the ADMM-based method.

6. Comparing the results will be obtained from the differential equation-based model with those will be obtained from the ADMM algorithm, assessing their performance, and identifying potential advantages or limitations.

7. Assessing the numerical comparison of the performance of the three-block ADMM algorithm with existing methods in the literature, using both benchmark and practical problems, testing its competitiveness and superiority in solving similar problems.

1.3 Contributions

The current study aims to provide notable contributions to the renewable energy area by attempting to fill several gaps. First, the study focused on building an optimization model using two-stage stochastic programming (SP) to size the hybrid renewable energy system components. Our study is among the first to consider stochastic optimization based on the three-block Alternating Direction Method of Multipliers (ADMM) in renewable energy applications. We have reformulated the two-stage SP in the form where ADMM can be applied. The benefit of doing this is that it allows parallel computations. To the best of our knowledge, this study was not carried out before. We have shown the convergence of the three-block ADMM algorithm theoretically and numerically by implementing it to a number of benchmark problems.

Second, we have developed a mathematical model that allows the decision of optimal sizing of renewable energy systems to be chosen over a number of years based on budget restrictions. The benefit of this approach is that it enables us to incrementally develop the renewable system as the project progresses rather than having to create the entire renewable system upfront for the whole duration of the project. The model is based on a system of differential equations. We consider this a major contribution to the research topic, as no such work has been reported in the literature. We addressed these gaps in the literature, which have the potential to provide new and innovative solutions for renewable energy systems.

Third, we have studied the suitability of the Machine Learning (ML) approach, including Long-Short Term Memory (LSTM), to predict wind power curves and solar power output. Thus, the ML provides a tool for a feasibility study to assess the potential renewable power in specific geographic locations.

1.4 Outline of the Thesis

This thesis comprises nine chapters. Each chapter focuses on a distinct concept. As a result, each chapter has its own introduction at the beginning. We organized the thesis as follows. Chapter 2 introduces three types of renewable energy sources. Also, it overviews the two-stage stochastic programming model and some related optimization techniques used in this thesis. Chapter 3 presents machine learning approaches for prediction. It introduces the problem of predicting wind turbine and solar power output. Then, it discusses implementations and evaluations of machine learning models for power prediction. Chapter 4 introduces two-stage stochastic programming and its application in renewable energy systems. Also, it develops the two-stage stochastic programming model into the three-block ADMM. Chapter 5 presents a detailed analysis of the convergence properties of the three-block ADMM algorithm. Chapter 6 illustrates the application of three-block ADMM optimization to a hybrid renewable energy system. It overviews existing optimization methods for hybrid systems. Then, it develops a new optimization method for hybrid systems using two-stage stochastic programming and solves it by the three-block ADMM. It presents the numerical experiments and results by comparison with existing methods. Chapter 7 presents a new dynamic optimization model for the hybrid renewable energy system and its solution using the pattern search method. Chapter 8 discusses the applications of three-block ADMM to benchmark problems from the literature. Also, it compares the computational results of our proposed three-block ADMM with the existing methods. Chapter 9 summarizes the main contributions of the thesis. It discusses the implications and future directions of the research. Finally, it concludes and makes recommendations for further work.

Chapter 2

Literature Survey

2.1 Introduction

Recently, a significant amount of literature has focused on renewable energy production due to its cost-effectiveness and its potential to reduce greenhouse gas emissions compared to conventional methods. Researchers concentrate on designing new systems using efficient renewable energy as an alternative solution to traditional energy. Part of these studies focuses on the technical side of manufacturing equipment and improving its efficiencies and other engineering-related issues. Some studies have focused on building mathematical models to produce economical and reliable energy systems that can meet the load demand. Some suggested mathematical models in the literature apply to solve energy applications [11, 12]. These models vary in performance and computational capabilities in solving energy problems.

Energy optimization problems have large sizes and require efficient algorithms to solve them within a reasonable time. Our focus in this research will be on modeling a stochastic programming mathematical method and its implementation. The intent is to find the optimal sizes of hybrid renewable energy systems comprising solar, wind, and hydrogen power generation with backup batteries.

Renewable energy systems rely on natural resources affected by weather fluctuations, which makes the generated power unpredictable. The reliability of such systems is a big concern because of the nondeterministic nature of the resources. A widely recognized measure of reliability is the Loss of Power Supply Probability (LPSP) [11], which calculates the probability of a power outage.

Many research studies have employed iterative techniques to optimize hybrid solar-wind energy systems using the LPSP concept. Chedid and Rahman [13] presented a method for designing and analyzing a hybrid wind-solar power system. Their solution employs a linear programming concept to minimize power costs while satisfying load demands reliably. They concluded that renewable energy is far more environmentally benign than diesel or grid alternatives. Their suggested methodology allows the user to investigate the interactions between economic, operational, and environmental issues, making it a beneficial tool for planning and assessing hybrid solar-wind power generating systems.

Chedid et al. [14] presented a multi-objective optimization-based technique to design a renewable energy system within a specified time frame. They examined the minimum emissions objective, which seeks to reduce greenhouse gas emissions and aims to achieve the lowest possible cost while meeting energy demand and reliability.

A different iterative algorithm, known as RAPSODY, employs the dynamic programming approach devised by De and Musgrove [15] to analyze optimal operating techniques for a hybrid electricity generation system. The researchers designed a model that incorporated a diesel generator, wind, and solar power sources, and a backup battery (if required) to evaluate the most efficient operating strategies. The approach they developed considers various factors such as capital expenses, maintenance and operating costs, and fuel expenditures to determine the most effective strategy for operating the system. Unfortunately, these iterative methods cannot handle problems with large dimensions due to their sequential processing.

Many researchers have shown an increasing interest in meta-heuristics algorithms for renewable energy applications. Heuristic and meta-heuristic approaches are popular techniques used in optimization problems that require finding an approximate solution when the exact one is not feasible. These approaches are based on searching through a large solution space to find the optimal solution. The main advantage of these techniques is that they are fast and require less computational resources compared to exact optimization methods. For example, Yang et al. [16, 17] presented a sizing approach for hybrid solar-wind systems based on a genetic algorithm (GA). They used the model to determine the best system design for achieving the LPSP target while minimizing the system's yearly cost. They used their technology to create a hybrid solar-wind system to power a communications relay station on a remote island in China.

Also, Koutroulis et al. [18] provided a technique for sizing a standalone solar and wind system using GA. Their approach finds the best number of units, guaranteeing that it minimizes the overall system cost over 20 years, subject to the restriction that they satisfy load energy requirements. They used their suggested method to model a power generation system to supply a residential home. They showed that hybrid solar-wind systems have lower system costs than either wind power or only solar sources.

Mohammadi et al. [19] offered an optimum microgrid design in a distribution system using a genetic algorithm. They utilized a hybrid power model comprising various components such as solar panels, hydrogen fuel cells, battery banks, and multiple diesel-generating units. They implemented a GA-based optimization technique to determine an optimal power output and pricing strategy for their proposed microgrid. The objective function evaluates the overall net present worth, and GA was used to determine the microgrid's optimal net present worth. The outcomes of their modeling demonstrated that employing diesel generators and microgrids resulted in a lower total net present worth. In contrast, Xu et al. [20] investigated a standalone hybrid wind and PV power system using GA to minimize total capital costs while taking into account the LPSP.

Hakimi et al. [21] used a particle swarm optimization (PSO) meta-heuristic approach in sizing a hybrid system. They used wind turbines, fuel cells, an electrolyzer, and hydrogen tanks. The biggest concern with renewable energy sources is their reliance on environmental conditions, which means they may not meet demand. Therefore, they used hydrogen tanks as storage components to solve the problem by supplying power demand when renewable producers are deficient. The hybrid wind-fuel cell system is suited for their case study location because of the proper wind speed throughout the year and the high costs of fuel transition.

Also, Boonbumroong et al. [22] proposed a technique for enhancing the configuration of a conventional AC-coupling hybrid power system that operates independently by utilizing PSO. Their method involved formulating the design as an optimization problem, which led to identifying the optimal hybrid system configuration resulting in reduced costs over the system's lifespan. To validate their system component models, the researchers employed a hybrid power system comprising

PV, wind, and diesel generators located in Thailand as a benchmark. Their findings revealed that particle swarm optimization (PSO) produced the best overall cost to the system. Also, [23, 24] presented two applications that use PSO in hybrid systems.

La Terra et al. [25] proposed a fuzzy logic-based multi-objective optimization technique for sizing a hybrid solar-wind power plant. Their optimization procedure aims to achieve a design that balances the long-term average performance of the generating system with its overall cost. The objective is to seek out an optimal balance between these two factors without compromising them.

Khatib et al. [26] proposed optimizing a PV/wind microgrid for rural house electrification. They combined an iterative method with a genetic algorithm to determine the optimal sizes for photovoltaic panels, wind turbines, and batteries. The main issue with the heuristics and meta-heuristics approaches is that the accuracy of the solution obtained through these techniques cannot be judged mathematically. This limitation makes it difficult to assess the quality of the solution obtained and the level of confidence that can be placed in it.

Yang et al. [27] developed an optimization approach for determining the sizes of hybrid solar and wind power generation systems in Guangdong, China. They used the Levelized Cost of Energy (LCE) [28] for economic analysis and the LPSP for reliability assessment. The study found that there is a strong correlation between LCE and LPSP. As LPSP decreases, the system cost (LCE) increases, and vice versa.

In a study conducted by Dursun et al. [29], a nursing home in Istanbul, Turkey was the focus of investigation for a hybrid renewable energy system. They utilized Hybrid Optimization of Multiple Resources (HOMER) software to design a standalone system that incorporated various combinations of wind turbines, PV panels, and fuel cells alongside hydrogen storage. The system cost was evaluated using both net present and total energy costs. The authors found that regions with higher wind speed and solar radiation had a significant impact on reducing energy costs.

The rest of this chapter will discuss various topics related to renewable energy sources and stochastic programming methods and is organized in the following way: In Section 2.2, we have discussed three types of renewable energy sources, namely, solar, wind, and hydrogen. This section aims to give readers a basic understanding of these renewable energy sources and their potential for providing sustainable energy. In Section 2.3, we have provided an overview of some of the stochastic programming methods that are used in the literature, which aims to provide readers with an understanding of the various stochastic programming methods available and which can be applied in the context of renewable energy systems. Finally, in Section 2.4, we have given an introductory overview of the pattern search method. This method is a derivative-free optimization method that can be used to solve optimization problems in the absence of analytical gradients. We have discussed the basic principles of the pattern search method, which will be used to solve optimization problems in the context of renewable energy systems.

2.2 Renewable Energy

Renewable energy depends totally on weather, which may affect system reliability. For any energy system that uses renewable energy sources, hybridizing two or more power generation methods with backup sources, like a conventional generator and batteries, will enhance system reliability [30]. In this section, we discussed various renewable sources that are used for power generation. We started with solar power, followed by wind power. Finally, we gave an overview of hydrogen and fuel cell technologies.

2.2.1 Solar System

A growing body of literature recognizes the importance of solar power, which is used commonly in electrifying rural areas where the national grid is difficult to reach these areas due to financial limitations and the sparse locations of rural areas [31].

Most modern solar energy applications supply houses with a few kilowatts (kW) of electrical power. The continuous increase in electricity prices motivates the development of these types of systems. In the domestic areas, where the national grid is connected but affected by long intervals of load-shedding, solar power might be a satisfactory solution.

Solar Photovoltaic (PV) cells are used in households to convert solar radiation into electricity using the photoelectric process. In the photoelectric process, the electrons are made excited and jump from the p-layer to the n-layer when the photons interact with the PV cell's surface, creating a voltage differential in the electrical circuit. The electrons then travel to keep the circuit in equilibrium [32].

PV cells are either linked in series or in parallel. A suitable combination of parallel and series of connections needs to be used according to the electrical appliance used by the users. The amount of solar radiation that reaches the earth is measured by kilowatt per meter-squared (kW/m^2). The variation of solar intensity depends on the geographic location; even during the day, the sunlight varies; it is most productive at noon and least early in the morning or none at night.

Constructing a solar farm requires choosing a suitable location that maintains a good potential for solar intensity. Performing a feasibility study for the location site requires at least one year of solar data to analyze the solar reliability in that location. The solar data depends on parameters like latitude, longitude, monthly average radiation, and daily horizontal radiation.

The technical and economic factors are very crucial when choosing a PV array. The technical issues are represented in the number of PV units, rated capacity per unit, the size of the panels, and their efficiencies. Also, economic factors like the lifetime of the PV cells, capital cost, replacement cost, and operation and maintenance costs are significant factors used in the feasibility study.

The power produced by a PV cell at a specified time t relates to the amount of solar irradiance received by the PV cell and the efficiency of the cell, and is given by

$$p_{pv}(t) = \frac{G(t)}{G_{STC}} \times p_{pv,rated} \times \eta_{pv}, \tag{2.1}$$

describes the relationship between the power output of the PV cell, $p_{pv}(t)$, and the solar irradiance received by the cell, $G(t)$, as well as the efficiency of the cell, η_{pv}. The variable $p_{pv,rated}$ is the power output of the PV cell under standard temperature conditions (STC), which are defined as a solar irradiance of 1 kW/m^2, an air temperature of 25 °C, and an air mass of 1.5. This value is used as a reference point to compare the actual power output of the cell under different conditions.

The efficiency factor η_{pv} of the PV cell is determined by many factors like the aging of the panel, deviation of the temperature from $25°$ C, and other factors. The variable $G(t)$ is the incident solar radiation on the solar panel, or the amount of solar energy received by cell per unit area, at time t. This value can vary depending on the time of day, weather conditions, and other factors. The variable G_{STC} is the solar irradiance under standard temperature conditions, which is used to normalize the actual solar irradiance. The total power produced by the PV system at time t is given by

$$P_{pv}(t) = N_{pv} \times p_{pv}(t),$$

where N_{pv} is the number of cells.

2.2.2 Wind Energy

Wind power is considered one of the most widely utilized and environmentally friendly sources of renewable energy. Due to continuous advancements and research, the development of wind turbines has proven to be highly effective in increasing wind power output. One of the key advantages of wind power is its ability to generate power at any time of the day or night. The generated wind power depends on the wind speed and the height of the turbine hub. The cut-in speed refers to the minimum wind speed required for the turbine to start operating, while the cut-out speed is the maximum wind speed at which the turbine can operate safely. The rated speed, which is the speed that produces the highest power output, lies between these two speeds [33].

Various elements determine wind turbine power, including wind speed, hub height, rotor radius, and generator efficiency. A feasibility analysis for constructing a wind turbine in a suitable location site needs at least one year of meteorological data.

Wind Resource

Meteorological data for wind speed is required to calculate the average power generated by a wind turbine. Data can be supplied by the user or retrieved from specialist websites, such as NASA.

The wind speed measured at the site's height -usually - is different from the one measured at the tower height. Given the wind speed v_0, measured at specific height h_0, if the height of the tower of the wind turbine was h, the wind speed v is given by

$$v = v_0 \left(\frac{h}{h_0}\right)^{\alpha},$$

where α is the power law exponent, a function of atmospheric stability. For open lands, α is considered to be $1/7$ [11].

The theoretical power curve of a wind turbine [11] illustrates the connection between the average wind speed and the average turbine's power output P_{avg}, and it is given by

$$P_{avg} = \frac{1}{2}\rho A c_p \bar{v}^3, \tag{2.2}$$

where the relationship depends on several parameters like ρ is the air density (kg/m^3); A is the swept area of the blade (m^2); c_p is power coefficient (dimensionless), and $\bar{v}$ is the average wind speed (m/s).

However, in real situations, it is difficult to have all the parameters presented in the equation, thus making the theoretical equation hard to use. To overcome this difficulty, researchers and engineers often use modeling techniques to estimate the power curve by utilizing a subset of the parameters.

Statistical distributions can be used to express the average wind speed. Usually, the two most frequently used statistical distributions for estimating wind power in a specific region are Weibull and Rayleigh. The probability density function (pdf) provides the probability of occurring at a specific wind speed. So, both Rayleigh and Weibull pdfs are used to test wind conditions at the selected site. The Weibull distribution's pdf is given by [34]

$$f(v;\ \lambda, k) = \frac{k}{\lambda}\left(\frac{v}{\lambda}\right)^{k-1} e^{-\left(\frac{v}{\lambda}\right)^k}, \quad v \geq 0,$$

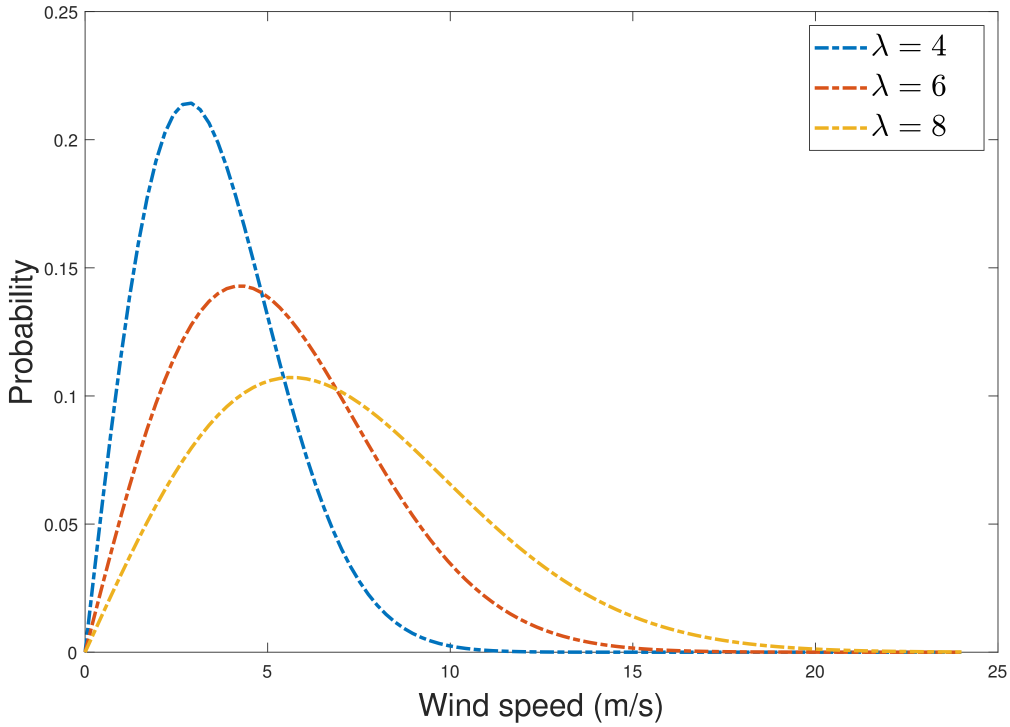


Figure 2.1: The Response of Rayleigh Distribution to Various Parameter Scales

where λ is called the scale parameter (m/s), and k is shape parameter which changes the look of the pdf. Usually, the value of k lies between 2 and 3. When the shape parameter is 2, the Weibull distribution is known as the Rayleigh distribution and is given by

$$f(v;\ \lambda) = \frac{2v}{\lambda^2} e^{-\left(\frac{v}{\lambda}\right)^2}, \quad v \geq 0.$$

Figure 2.1 shows the consequence of changing the scale parameter, where it displays that the higher values of λ shift the curve toward higher wind speeds.

The average wind speed $\bar{v}$ is directly related to the scale parameter of Rayleigh distribution, given by the following equation:

$$\bar{v} = \int_0^\infty v f(v,\lambda) dv = \int_0^\infty \frac{2v^2}{\lambda^2} e^{-\left(\frac{v}{\lambda}\right)^2} dv = \frac{\sqrt{\pi}}{2}\lambda,$$

or we can write it in the following way

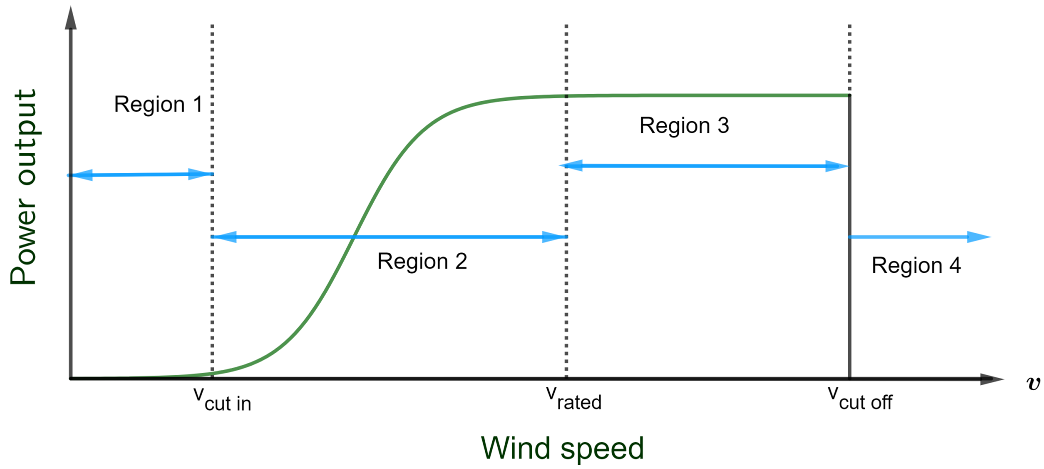


Figure 2.2: Wind Turbine Power Curve

$$\lambda = \frac{2}{\sqrt{\pi}}\bar{v},$$

then, the Rayleigh distribution takes the following form

$$f(v;\ \lambda) = \frac{\pi v}{2\bar{v}^2} e^{-\frac{\pi}{4}\left(\frac{v}{\lambda}\right)^2}, \quad v \geq 0.$$

Wind Turbine

The power produced by a wind turbine is directly proportional to the wind speed. Once the wind speed exceeds the minimum threshold, the turbine initiates power generation. On the other hand, if the wind speed surpasses the maximum limit, the turbine may become dangerous, and a brake system gets activated to halt the turbine.

The power curve of a wind turbine appears in Figure 2.2. The power curve depicts the connection between wind turbine power output and wind speed. At a time t, the amount of power produced by a wind turbine is given by [35]:

$$p_{wind}(t) = \begin{cases} q(v(t)) & v_{\text{cut in}} < v(t) < v_{\text{rated}}, \\ P_{rated} & v_{\text{rated}} \leq v(t) \leq v_{\text{cut out}}, \\ 0 & \text{otherwise}, \end{cases} \tag{2.3}$$

where P_{rated} is the rated (maximum) power of the turbine in kilowatt (kW); $v_{\text{cut in}}$ the cut-in speed (m/s); $v_{\text{cut out}}$ the cut-out speed (m/s); and v is wind speed (m/s) [36]. The function $q(v)$ describes the non-linear curve of Region 2 in Figure 2.2, which is approximated using various methods such as parametric and non-parametric. The total power produced by the wind turbine, at time t, is given by:

$$P_{wind}(t) = N_{wt} \times p_{wind}(t) \times A_{wt} \times \eta_{wt},$$

where N_{wt} is the number of turbines, A_{wt} is the total swept area of the wind turbine, and η_{wt} is the efficiency of the wind turbine.

A high altitude location away from houses and trees with an intensity of wind speed, like mountain gaps, is the best location for installing a wind farm. Estimating wind energy is based on wind speed and wind turbine power curves. The future forecast of power output is critical in sizing and cost-optimal analyses. The precise prediction model aids in preventing overestimation and underestimation, which minimizes needless expenses.

The manufacturer's power curve for the wind turbine is created under standard conditions, which might not comply with the real-time requirements of the site under consideration. So, modeling the power curve with actual wind speed data and power measured from the turbine will give a more sensible result. Such an approach has been used in this thesis; see next chapter.

2.2.3 Hydrogen Energy

In recent times, people have become increasingly intrigued by hydrogen and fuel cell technologies that are seen as clean, dependable sources of renewable energy. These fuel cells do not give off any toxic fumes when using hydrogen. Thus, these methods are considered a practical solution for the 21st century with zero greenhouse gas emissions.

Hydrogen is not an energy source, but it is an energy carrier. The hydrogen H_2 is the most abundant element in the universe [37, 38] since the sun consists of 70% of hydrogen and 28% of helium. Hydrogen enjoys some properties; it is much lighter than air and is a highly flammable and reactive substance. Also, it is a gas at the ordinary temperature and pressure, and it is transformed to liquid under $-235\ C^{\circ}$ only [39].

Hydrogen is found abundantly in the universe, but usually, it is part of a compound and rarely on its own. Consequently, a method of obtaining hydrogen that is clean and does not leave a negative environmental impact is water distillation. This process, known as water splitting, can be done through electrolysis or thermolysis. This process is depicted in Figure 2.3. The device used in an electrolysis operation is known as an electrolyzer.

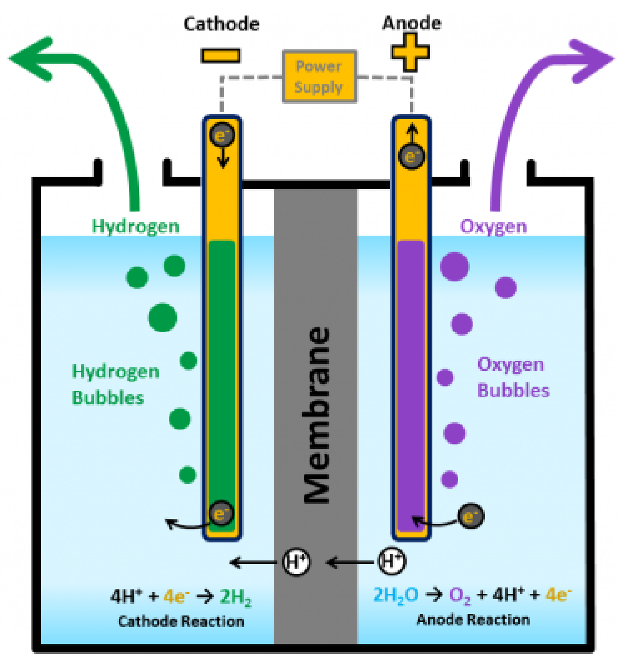


Figure 2.3: Producing Hydrogen by Electrolysis

The electrolyzer consists of three main parts: an anode, a cathode, and an electrolyte separating them. The most commonly used electrolyte is the Proton Exchange Membrane (PEM). The water reacts at the anode to produce oxygen, hydrogen protons (H^+), and electrons (e^-). The oxygen (O_2) will be released into the environment, and the hydrogen protons (H^+) will pass through the membrane to the cathode side, while the electrons (e^-) will be attracted by the positive charges in the anode rod. At the cathode, the hydrogen protons (H^+) react with the electrons (e^-) that come from the cathode to produce hydrogen gas (H_2). The hydrogen gas will then be transported through an external channel to be used or stored in tanks. The chemical reaction at the electrolyzer is expressed as

$$\text{Anode reaction:} \qquad 2H_2O \longrightarrow O_2 + 4H^+ + 4e^-$$
$$\text{Cathode reaction:} \qquad 4H^+ + 4e^- \longrightarrow 2H_2.$$

The generator that uses hydrogen as a fuel is called the Fuel Cell (FC); commercially, it is known as Proton Exchange Membrane Fuel Cell (PEMFC) [40]. The fuel cell produces electricity by converting the chemical energy to electrical energy using the stored hydrogen, and oxygen from the surrounding environment. Like an electrolyzer, a fuel cell consists of an anode and a cathode separated by an electrolyte. The PEMFC works by reversing the electrolyzer functionality, and it is expressed as:

$$\text{Anode reaction:} \qquad H_2 \longrightarrow 2H^+ + 2e^-$$
$$\text{Cathode reaction:} \qquad 2H^+ + \frac{1}{2}O_2 + 2e^- \longrightarrow 2H_2O.$$

When hydrogen enters the fuel cell through the anode, the catalyst breaks it down into protons (H^+) and electrons (e^-). The protons (H^+) will travel through the membrane to the oxygen on the opposite side at the cathode. Electrons move through an external circuit, generating an electrical current that serves as the power source. The hydrogen protons (H^+) will react with the oxygen protons (O_2) to generate water on the cathode side.

When the generated power exceeds the load demand, the remaining power is converted into hydrogen and transferred into the tank. The following equation describes the power transfer from the electrolyzer to the hydrogen tank [41]:

$$P_{el-tank} = P_{g-el} \times \eta_{el},$$

where $P_{el-tank}$ is the power transferred from the electrolyzer to the hydrogen tank, P_{g-el} is the generator's surplus power provided to the electrolyzer, and η_{el} is the electrolyzer efficiency. The energy of hydrogen contained in tanks at time t is calculated as follows:

$$E_{tank}(t) = E_{tank}(t-1) + P_{el-tank}(t) - P_{tank-fc}(t) \times \eta_{tank},$$

where $E_{tank}(t)$ stands for the amount of energy stored in the tank at time t, $P_{tank-fc}(t)$ the amount of power generated by fuel cells using the hydrogen from the tank at time t, and η_{tank} the storage efficiency.

2.3 Stochastic Programming

Many uncertainties in real-life problems are encoded mathematically in the problem's parameters. Uncertainties result from random experiments with specified outcomes modeled by ω (called a random variable). The set of all outcomes is called sample space and is denoted by the symbol Ω. A subset of outcomes in Ω is known as events. The set of all such events is denoted by $\mathcal{F}$. Each event $A \in \mathcal{F}$ can be measured by the probability function $P : \mathcal{F} \to [0, 1]$. The triple space $(\Omega, \mathcal{F}, P)$ is known as the probability space.

Stochastic programming is an area of optimization that deals with modeling and solving problems that have uncertain parameters. Most stochastic problems incorporate discrete decisions. The classification of the decision depends on the information that is known to the decision-maker. A well-known problem of stochastic optimization is called two stages stochastic program, which is given by [42]

$$
\begin{aligned}
&\min_{x} c^T x + \mathbb{E}_\omega Q(x, \omega), \qquad \text{subject to} \\
&Ax = b, \\
&x \geq 0,
\end{aligned} \tag{2.4}
$$

where

$$Q(x, \omega) = \min\{q_\omega^T y \mid W_\omega y = h_\omega - T_\omega x, y \geq 0\},$$

where ω is the random variable, x is called the first-stage decision variable, which needs to be calculated before revealing the future realization of the random event. The variable y is the second-stage decision variable corresponding to the decisions after the realization of a random event known as recourse decision [42]. For each realization of ω, we have $W_\omega, T_\omega, h_\omega$, and q_ω. The recourse decision is intended to compensate for any bad decisions caused by the first-stage decision variable.

The matrix W_ω is called the recourse matrix. If this matrix does not depend on the random variable ω, it is called a fixed recourse matrix. For simplicity and to illustrate the concepts, this section will only consider the case of a two-stage problem with a fixed recourse matrix.

2.3.1 L-Shaped Method

The L-shaped algorithm is used in stochastic optimization to solve two-stage SP. It is used to solve problems where the first-stage decision variables are deterministic while the second-stage decision variables are random. The algorithm combines the idea of deterministic equivalence and a heuristic method called cutting plane to solve the problem. It approximates the nonlinear function, Q, with linear functions [42] to avoid the complex evaluation that may arise in the calculation. The extensive form of the block structure to the L-shaped method is shown in Figure 2.4. The name of the L-Shaped method arises from this structure. The L-shaped algorithm starts by solving the deterministic equivalent of the problem, which is a linear program (LP) with the expected values of the second-stage random variables. Then, it iteratively adds the second-stage constraints to the LP and solves it again until finding a feasible solution.

The method assumes a finite number of realizations to the random variable ω. The algorithm involves finding the initial solution and then adding constraints to ensure it is feasible and optimal. Then, two types of constraints are added: (i) *Feasibility cuts*: ensuring that the solution of the first stage is not violating the second-stage constraints and (ii) *Optimality cuts*: which are linear approximations to the second-stage function. The algorithm terminates when the optimal solution is found or when a stopping criterion is met. The L-Shaped algorithm is beneficial when the first-stage decision problem has a small number of constraints and a small number of variables. One of the main disadvantages of the L-Shaped algorithm is that it does not provide any guarantee of global optimality, and it may get trapped in locally optimal solutions. Also, it can become problematic when the number of realizations increases, leading to a dimensionality explosion, and may not always converge to the optimal solution.

Consider Equation (2.4), where it has a finite number of scenarios. Let $k = 1, \ldots, N$ be the index of the finite scenarios of the random variable ω, and let p_k be the probabilities associated with each scenario. We can define large-scale (extensive form) linear programming as

$$
\begin{aligned}
&\min c^T x + \sum_{k=1}^{N} p_k q_k^T y_k \\
&\text{s.t.} \qquad Ax = b, \\
&\qquad\qquad T_k x + W y_k = h_k, \qquad k = 1, \ldots, N, \\
&\qquad\quad x \geq 0, \quad y_k \geq 0.
\end{aligned}
\tag{2.5}
$$

The formulation of Problem (2.5) is noteworthy from various perspectives. Firstly, it is an expanded version of a Linear Programming problem. The matrices T_k can be thought of as having the same dimensions as the matrix A in LP. In practical applications, the size of A is often substantial. As a result, when the number of realizations N is large, the size of Problem (2.5) can become very large.

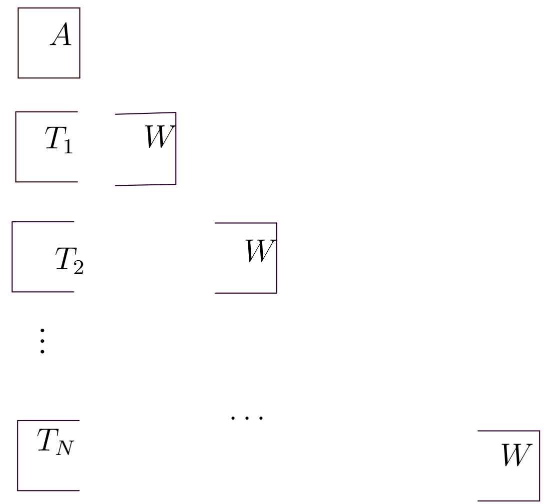


Figure 2.4: Block Structure of the Extensive Form of Two-stage Stochastic Programming

2.3.2 Progressive Hedging Algorithm

The Progressive Hedging Algorithm (PHA) [43] is a decomposition method used to solve two-stage stochastic programming problems. It is a type of regularized decomposition algorithm. Regularized algorithms can handle significant drawbacks of the cutting plane methods, like the degeneration that may happen at the end of the process. The basic idea behind the algorithm is to transform the original problem into a sequence of subproblems according to scenarios, each of which is a deterministic optimization problem. PHA is beneficial for solving problems with plentiful uncertain parameters and can be applied to a wide range of optimization problems, such as portfolio optimization, resource allocation, and risk management.

The PHA method starts by solving a deterministic equivalent problem, which is a relaxation of the original problem where the uncertain parameters are replaced by their expected values. Then, in each iteration, the algorithm solves a subproblem by fixing the value of one of the uncertain parameters, called a scenario. The subproblem is a linear program that is solved using a linear programming algorithm. Then, the solution of the subproblem is employed to update the dual variables of the problem, which in turn is utilized to update the solution of the deterministic equivalent problem. The algorithm continues to iterate until a stopping criterion is met, such as a certain level of accuracy or a maximum number of iterations. The final solution is a combination of the subproblems, and it is an approximation of the optimal solution of the original problem.

One of the main advantages of the PHA is that it can handle large-scale problems with numerous scenarios and decision variables. Also, the algorithm is relatively simple to implement and has parallel implementation. However, it requires a large number of computational resources, especially when the number of scenarios increases.

Problem (2.4) can be put into an extensive form as

$$\zeta^{SP} := \min_{x,y_s} \left\{ c^T x + \sum_{s\in S} p_s q_s^T y_s : (x, y_s) \in K_s, \forall s \in S \right\}, \tag{2.6}$$

where $K_s = \{(x, y_s) : W_s y_s = h_s - T_s x, x \in X, y_s \in Y_s\}$, and S is the set of a finite scenarios. Problem (2.6) has a decomposable structure by creating copies of the first-stage variable, x, for all scenarios. The new structure takes the form

$$\zeta^{SP} := \min_{x_s,y_s} \left\{ \sum_{s\in S} p_s \left(c^T x_s + q_s^T y_s \right) : (x_s, y_s) \in K_s, x_s = z, \forall s \in S, z \in \mathbb{R}^{n_x} \right\}. \tag{2.7}$$

The constraint $x_s = z, s \in S$, imposes that the first-stage decisions x_s must be the same for all scenarios $s \in S$ —*non-anticipativity constraints*. It ensures that the decision of one stage does not depend on the future stages' decisions. These constraints guarantee that decisions made at each step of the algorithm are consistent with a decision-making process that proceeds in a sequential, one-stage at-a-time manner. They secure that the algorithm solves the problem correctly and that the produced solutions are meaningful and interpretable. Non-anticipativity constraints create two copies of each variable; one representing the value before revealing the realization of some random variable, and the other representing the value it takes after the realization has become known. In reality, those values must be equal since we have to fix the variable value before disclosing the realization. Application of Lagrange relaxation to Equation (2.7) yields the non-anticipativity

Lagrange dual function [43]

$$\phi(\mu) = \min_{x,y,z} \left\{ \sum_{s \in S} \left[p_s \left(c^T x_s + q_s^T y_s \right) + \mu_s^T \left(x_s - z \right) \right] : (x, y_s) \in K_s, \forall s \in S, z \in \mathbf{R}^{n_x} \right\}, \quad (2.8)$$

where $\mu = \left(\mu_1, \cdots, \mu_{|S|} \right)^T \in \Pi_{s \in S} \mathbb{R}^{n_x}$ is the vector of Lagrange multipliers associated with the non-anticipativity constraints. Setting $\omega_s = \frac{1}{p_s} \mu_s$, Equation (2.8) may be written as [43]

$$\phi(\omega) = \min_{x,y,z} \left\{ \sum_{s \in S} p_s L_s(x_s, y_s, z, \omega_s) : (x_s, y_s) \in K_s, \forall s \in S, z \in \mathbf{R}^{n_x} \right\}, \quad (2.9)$$

where

$$L_s(x_s, y_s, z, \omega_s) := c^T x_s + q_s^T y_s + \omega_s^T \left(x_s - z \right).$$

The augmented Lagrange function is founded by adding the regularized term of the non-anticipativity constrain as follows

$$L^\rho(x, y, z, \omega) = \sum_{s \in S} p_s L_s(x_s, y_s, z, \omega_s),$$

with

$$L_s^\rho(x_s, y_s, z, \omega_s) := c^T x_s + q_s^T y_s + \omega_s^T \left(x_s - z \right) + \frac{\rho}{2} \|x_s - z\|^2,$$

where $\rho > 0$ is the penalty parameter. The PHA generates a sequence of iterations that converge to the solution if it exists. There is no restriction on the value of ρ, but a suitable choice of ρ will lead to fast convergence.

2.4 Pattern Search Algorithm

The pattern search method is a type of direct search optimization algorithm that is a variation of the coordinate search method [44]. It falls under the class of direct search methods. Also, it is referred to as the generalized pattern search (GPS) method [45]. The pattern search method generates a sequence of iterates $\{x^1, x^2, \cdots, x^k, \cdots\}$ where the objective function values are nonincreasing. The method consists of two steps: the SEARCH step and the POLL step. In the literature on this method [46], there is limited information provided on implementing the SEARCH step. Despite the lack of information, the results of the pattern method using only the POLL step have been reported in various studies. Audet et al. [45] reported that the SEARCH step can be a hindrance to convergence, which is why it is not always included in the implementation of the pattern search method.

In this thesis, we intend to focus solely on the POLL step in the pattern method. Before delving into the pattern search algorithm based on the POLL step, we would like to provide more in-depth information about the POLL step. The POLL step is crucial to the pattern method as it updates the current iterate and the step size based on the results of the previous iteration. It allows the algorithm to progress toward the optimal solution.

The POLL process starts by selecting a trial point, denoted as p_i, from the poll set P_k. The trial point is calculated as [47, 48]

$$p_i = \{x^k + \Delta_k d_i : d_i \in D, i = 1, \cdots, 2n\},$$

where

$$D = \{e_1, \cdots, e_n, -e_1, \cdots, -e_n\},$$

where $e_i \in \mathbb{R}^n$ is the i^{th} unit coordinate vector.

The current iterate x^k is used as the starting point for this calculation. The trial point, p_i, is then evaluated to determine if it is a better solution than the current iterate. If a better solution is found ($f(p_i) < f(x^k)$), the POLL process ends, and the next iterate, x^{k+1}, is updated to be the better solution $x^{k+1} = p_i$. If no better solution is found for all $d_i \in D$, the current iterate is retained $x^{k+1} = x^k$. In the case of a successful POLL step, the step size parameter, Δ_{k+1}, is increased to $2\Delta_k$ to enhance exploration. On the other hand, if the POLL step is unsuccessful, the step size parameter is decreased to $\frac{1}{2}\Delta_k$. Thus, the iterates are updated as follows [47, 48]:

$$
x^{k+1} = \begin{cases} p_i & \text{if } f(p_i) < f(x^k) \quad \text{for some } p_i \\ x^k & \text{if } f(p_i) \geq f(x^k), \quad \forall p_i \in P_k. \end{cases} \tag{2.10}
$$

Also, the updates for the step size is summarized as follows [47, 48]:

$$
\Delta_{k+1} = \begin{cases} 2\Delta_k & \text{if } f(p_i) < f(x^k) \quad \text{for some } p_i \\ \frac{1}{2}\Delta_k & \text{if } f(p_i) \geq f(x^k), \quad \forall p_i \in P_k. \end{cases} \tag{2.11}
$$

Therefore, the update of the iteration k links to the update of Δ_k. The pattern search method continues to iterate through the POLL step until the objective function values stop decreasing or meet the stopping criterion, i.e. $\Delta_k < \Delta_{tol}$. Algorithm 1 provide the implementation for the pattern search method using the POLL step.

Algorithm 1 Pattern Search Optimization Algorithm

Input: Initial feasible solution $x^0 \in \mathcal{X}$, positive spanning set D, initial step size $\Delta_0 > 0$, stopping tolerance $\Delta_{tol} > 0$
Output: Optimal solution x^*
Set counter $k \leftarrow 0$ and $i \leftarrow 1$
while $\Delta_k \geq \Delta_{tol}$ **do**
    Evaluate the objective function f at the trial point $p_i = (x^k + \Delta_k d_i) \in P_k, d_i \in D$
    if $f(p_i) < f(x^k)$ **then**
        Set $x^{k+1} \leftarrow p_i$
        Increase $k \leftarrow k + 1$
        Set $i \leftarrow 1$
        Expand the mesh by increasing $\Delta_{k+1} \leftarrow 2\Delta_k$
    else
        Increase $i \leftarrow i + 1$
        if $i > 2n$ **then**
            Set $x^{k+1} \leftarrow x^k$
            Increase $k \leftarrow k + 1$
            Set $i \leftarrow 1$
            Reduce the mesh by decreasing $\Delta_{k+1} \leftarrow \frac{1}{2}\Delta_k$
        end if
    end if
end while
Return $x^* = x^k$

Chapter 3

Predicting Wind Turbine and Solar Power Output Using Machine Learning

3.1 Introduction

Electricity is an indispensable requirement in modern industrialized and developing nations. With the population and industries growing at a rapid pace, the demand for power and energy is increasing every year. However, conventional resources are becoming scarce and are not equally available worldwide, creating tensions between nations. Fuel costs are unstable and vary due to various economic and political factors. Moreover, greenhouse gas emissions caused by conventional resources have increased substantially over the past 50 years, exacerbating the situation [7].

To address this issue, renewable energy sources such as solar, wind, hydrogen, and hydro have emerged as potential alternatives for power generation. These sources are ideal for reducing the carbon footprint. Moreover, by combining renewable energy systems with energy storage systems, excess energy generated during off-peak hours can be stored and used during peak hours, reducing the need to use fuel-based energy systems and thereby lowering greenhouse gas emissions. Governments, organizations, and individuals worldwide are actively pursuing ways to increase the use of renewable energy to address climate change.

Wind and solar power are fast-growing areas in renewable energy. However, their success is highly dependent on weather conditions, making it challenging to predict their output accurately. Wind and solar power rely on wind speed and solar radiation, respectively, which are intermittent resources. Therefore, accurate predictive models based on climatic data are required to size hybrid renewable systems and optimize the operation of wind turbines and solar panels. Predictions can help to plan future energy systems, ensure sufficient power to meet customer demands, and avoid blackouts. Additionally, predictions can improve energy management by forecasting energy needs during peak hours and energy surplus during off-peak hours, allowing for better control of energy storage systems and optimal use of renewable energy systems.

The power curve of a wind turbine plays a crucial role in monitoring its output and defines its response to varying wind speeds [49]. Typically, wind turbine manufacturers provide a datasheet with a power curve, which is created under standard weather conditions. However, this curve may not accurately represent the actual site conditions where the turbine operates. Therefore, predicting the power output of a wind turbine using climatic data is crucial to sizing hybrid systems and ensuring optimal performance.

Forecasting power output and performance monitoring are among the most crucial applications of the power curve. Accurately predicting the power output allows operators to schedule maintenance, plan for grid integration and make other operational decisions. The power curve is also helpful in identifying regions suitable for wind power construction. Two types of models are employed to estimate the power curve using machine learning: parametric and non-parametric models. Parametric models involve assuming a functional form for the power curve, such as polynomials, and estimating the values of the parameters from the available data. These models are typically simple and easy to implement, but they may not accurately capture the complex relationship between wind speed and power output.

Non-parametric models, on the other hand, do not assume any functional form for the power curve and instead use the available data directly to estimate the power curve. These models are more flexible and can better capture the complex relationship between wind speed and wind turbine power output, but they can be more computationally intensive and may require more data to achieve good results.

To accurately predict the power output of a wind turbine, we can leverage the power of machine learning techniques. The first step is to analyze the standard wind turbine power curve depicted in Figure 2.2, which is divided into four distinct regions. By understanding the unique characteristics of each region, we can train a machine-learning model that can accurately predict the power output. With a well-trained machine learning model, we can optimize the operation of the wind turbine by predicting its power output in real time based on environmental factors such as wind speed and direction. This prediction will enable us to maximize the power generation of the wind turbine and ensure the efficient utilization of renewable energy sources.

By analyzing the available wind speed data, we can identify areas with high wind speeds and better wind resources, which are more favorable for wind power development. In designing hybrid energy systems, the accurate prediction of the wind turbine power output is crucial for sizing and optimizing wind turbines during the design stage. Also, it helps estimate the turbine's capacity factor, which determines the average energy production over time [50].

On the other hand, the quantity of energy produced by a PV system is proportional to climatic factors such as cloud cover, solar intensity, and site-specific circumstances. The PV panels are most productive during summer. However, in the case of rain and windy conditions, the energy produced by PV panels is much less. It is crucial to consider weather forecasts when planning for a PV system to account for this variability in solar power. This forecast helps to ensure that the system can generate the required solar energy and seek backup resources when there are less favorable weather conditions.

Predicting the power output of wind turbines and solar panels is a crucial task in the renewable energy industry. Accurate predictions can optimize the operation of renewable energy systems, plan for future energy demand, and evaluate the potential of new projects.

Machine learning algorithms can analyze historical meteorological data to learn the relationship between weather and the power output of renewable energy systems. The algorithms can then use this knowledge to predict future power output, making it easier for energy providers to plan and manage their operations.

In predicting the power output of wind turbines, machine learning algorithms rely on meteorological data, such as wind speed, wind direction, and pressure. In contrast, predicting solar power output depends on temperature, solar radiation, and pressure.

Several machine learning algorithms, such as regression, decision trees, random forests, and neural networks, are available for predicting wind turbines and solar power output. The choice of algorithm depends on the specific problem and the data.

Machine learning-based predictions of renewable energy output can be more accurate than traditional methods using reasonable feature engineering, particularly when the relationship between weather and power output is complex and nonlinear. It is because machine learning algorithms can identify patterns in large datasets that humans may miss, enabling more accurate predictions.

One of the main challenges in renewable energy applications is the variability of wind speed and solar irradiance from one location to another. This variability imposes the design of the hybrid system to be for a particular range of weather conditions. Thus, an accurate wind turbine power output is essential in selecting turbines that match the chosen site's wind regime. Also, proper power output models for PV panels help to size the system components. Because of increased data availability and computer capacity, machine learning algorithms can make better predictions. Machine learning and time series models are used to predict renewable energy generation, which is critical for many sectors in the energy market.

In this chapter, multiple machine-learning approaches are applied to time series data. We found that employing time series models for renewable energy data is challenging due to their non-stationary nature. As a result, we realized that utilizing machine learning techniques is a more feasible option. We implemented machine learning algorithms to predict wind turbine and solar power output. The models used a dataset obtained from Wind Atlas for South Africa. We found that the Long short-term memory (LSTM) and Extreme Gradient Boosting Regression Trees outperform the other methods on average.

The rest of this chapter is structured as follows: Section 3.2 discusses some related works on wind and solar power output predictions. Section 3.3 describes the methodology and data used in the machine learning algorithms. It covers data processing, cross-validation of time series data, and the performance metric employed. The various machine-learning algorithms employed in the study are presented in Section 3.4. Section 3.5 covers the numerical experiments of the machine learning algorithms carried out for both wind and solar power. Finally, Section 3.6 summarizes the results of the wind and solar power predictions and highlights the algorithms utilized.

3.2 Related Works

The wide-scale development of renewable energy showed a substantial and expanding body of literature focused on wind turbine and solar power output prediction. This section summarizes some known prediction methods used. Inman et al. [51] gave a comprehensive evaluation of renewable energy forecasting theory. Their essay discusses several hypotheses, such as how to simulate irradiance, air masses, and clearness indexes. They offered a theoretical foundation for frequently used time series models and machine learning approaches. They also provided theories on irradiance and satellite photos to demonstrate how they may be applied. Their work provides directions for both theory and methodology in physics, statistics, and machine learning.

Hong et al. [52] presented an overview of several energy forecasting techniques. They highlighted the increased usage of time series and machine-learning approaches in various energy scenarios. A principal part of their research is based on an energy forecasting competition and assesses the methods implemented in the Global Energy Forecasting Competition 2014. They also anticipated that the research on solar power forecasting will grow significantly during the next decade.

Bacher et al. [53] proposed a technique for predicting power output from photovoltaic systems installed on rooftops in a Danish village using online forecasting. Their method used adaptive linear time series models that relied on autoregressive (AR) and AR models incorporating exogenous variables. However, it is worth noting that the models they have used may be insufficient to accurately predict non-stationary data, which is common in climatic data related to PV systems.

The power curve depicts the reaction of the power output of a wind turbine power to varying wind speeds. Ehsan [54] provided reasons for under and over-estimation due to manufacturers' power curves. As a result, it is critical to estimate the wind turbine power output using climatic data. Finding power output in this way can provide operational data reflecting real situations.

An accurate power prediction can avoid the problem of under and over-estimation. Also, it is used as an indicator of efficiency by calculating the capacity factor. The power curve also benefits in selecting the turbine's model by comparing different models and choosing the appropriate option that suits the specific geographical location.

Two types of modeling are well-known for the wind turbine power curve, namely, parametric and nonparametric methods. The parametric methods consider the pattern of the power curve, then try to create the relationship between the generated power and the wind speed data. Some parametric methods use functions that can be linear [55, 56], quadratic [57], cubic, and exponential [58]. The other parametric methods consider the inflection point where the curve changes its direction, like double exponential, 3PL, 4PL [59, 60] and 5PL [55]. On the other hand, nonparametric methods do not consider the shape of the curve; it is suitable to derive the power curve when it is hard to get a functional relationship using the parameters (such as temperature, humidity, and wind direction) on a given data.

Kusiak et al. [59] constructed power curves using historical wind turbine data for online monitoring of power curves. In their study, they used three approaches; two parametric methods and one nonparametric method. They used the least square and maximum likelihood as estimation methods to derive the power curves as a parametric approach. They reported that the least square beats the maximum likelihood one in terms of accuracy. On the other hand, they used a nonparametric model called the k-nearest neighbor (k-NN) algorithm. Also, in a different paper, Kusiak et al. [60] found that k-NN performs poorly when the condition of the wind farm is abnormal. Then, they combined an evolutionary method with a data mining approach to filter the original data when it has noisy points.

Thapar et al. [57] have done a critical analysis for mathematical modeling for the wind turbine using three commercial wind turbines. They reported that the models that use the fundamental power equation are not reflecting the actual wind turbine. They noted that the models based on the shape of the power curve lack the desired accuracy. They also concluded that using curve fitting techniques, such as least squares, and cubic spline methods for deriving the power curve gives a moderately accurate result for wind turbine power curves.

3.3 Methodology: Data Collection, Processing, and Validation

This section provides an overview of the climatic data used in our study and the correlation employed to identify the significant predictor variables. It will also discuss the preprocessing of the data and describe the cross-validation technique utilized in our machine learning (ML) algorithms for time-series data.

3.3.1 Data Collection

The climatic dataset used in this study was collected from Wind Atlas for South Africa (WASA) [61]. The dataset contains data for both wind power and solar power. We selected the dataset from October 2015 until February 2022 for the analysis as the 2022 data was incomplete for the entire year.

The collected dataset comprises a total of $297,318$ records, with numerical values of eight features: time, wind speed, direction codes, relative humidity, barometric pressure, wind turbine power output, solar irradiance, and PV solar output. In our solar power analysis, we have included two crucial features that were generated from the time feature: month and hour. These additional features provide significant insights into solar power generation patterns. The "hour" feature determines the duration when the solar panels are most productive throughout the day. This information is invaluable for identifying the ideal time to maximize electricity generation. For example, our analysis indicates that solar panels generate electricity most efficiently between 9 a.m. and 3 p.m. The "month" feature is essential for understanding the seasonal patterns of solar irradiance, and it also plays a crucial role in optimizing solar power generation.

To begin with, we performed the correlation analysis of the features mentioned above. We have used the built-in correlation function in Python's programming language. This was done to identify the most correlated features in the collected data for both wind turbine and solar power output. The results indicated that wind speed, relative humidity, and barometric pressure were the top three correlated parameters for the wind turbine power output, as shown in the last horizontal row in Figure 3.1, which shows the correlation heatmap between the numerical features in the dataset. Those top correlated features will be used as predictors in the ML algorithms to predict the wind turbine power output.

Similarly, solar irradiance, barometric pressure, relative humidity, and hour were the top correlated parameters for solar power output, as depicted in the last horizontal row in Figure 3.2; these will be used as predictors for the solar power output. Notably, it was observed from Figure 3.2 that barometric pressure and relative humidity were inversely proportional to PV power. These correlation results suggest the importance of considering multiple parameters when analyzing wind and solar power outputs, as they are significantly affected by multiple weather conditions.

The dataset we have used is time-series data, which has a sequential time index. The time-series-related tasks are well-known in real-world problems. We have used the time series cross-validator, which is a useful tool for dividing time series data samples that are seen at fixed time intervals into training and testing sets. The test dataset is used to assess the effectiveness of various prediction models in terms of their performance. The presented validation tool is a modified version of KFold that utilizes the initial k folds as training data and the $(k+1)$th fold as the test set for each iteration. Nonetheless, unlike conventional cross-validation methods, here the following training sets incorporate previous ones. By adopting the time series cross-validator, we can ensure that your model is properly trained and evaluated on suitable datasets.

The datasheet of the wind turbine Vestas V90 is obtained from [62], which is the manufacturer's datasheet. Vestas V90 is a wind turbine with a rated (maximum) power of 2000 kW. The cutting-in speed to start generating power is $4.0m/s$. The rated wind speed is $13.0m/s$, then the speed becomes constant until the wind speed reaches $35.0m/s$, so a barking system needs to be employed to bring the turbine to a stand-still situation.

Vestas V90 is a three blades wind turbine with two variants, either made from fiber or carbon. Each blade has a radius of $45m$, which makes the turbine swept area $6,362\ m^2$. The type of

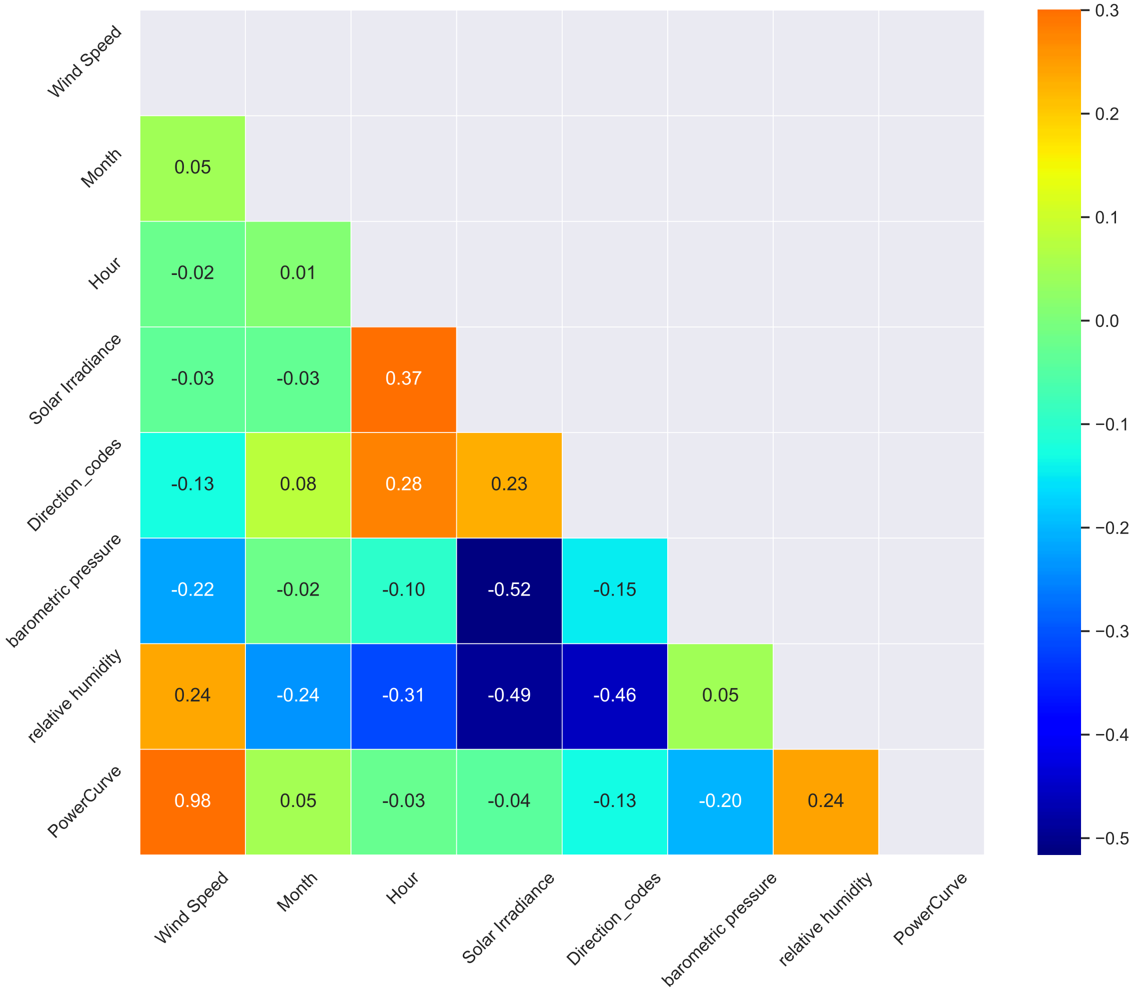


Figure 3.1: Correlation Analysis of Wind Turbine Power Features

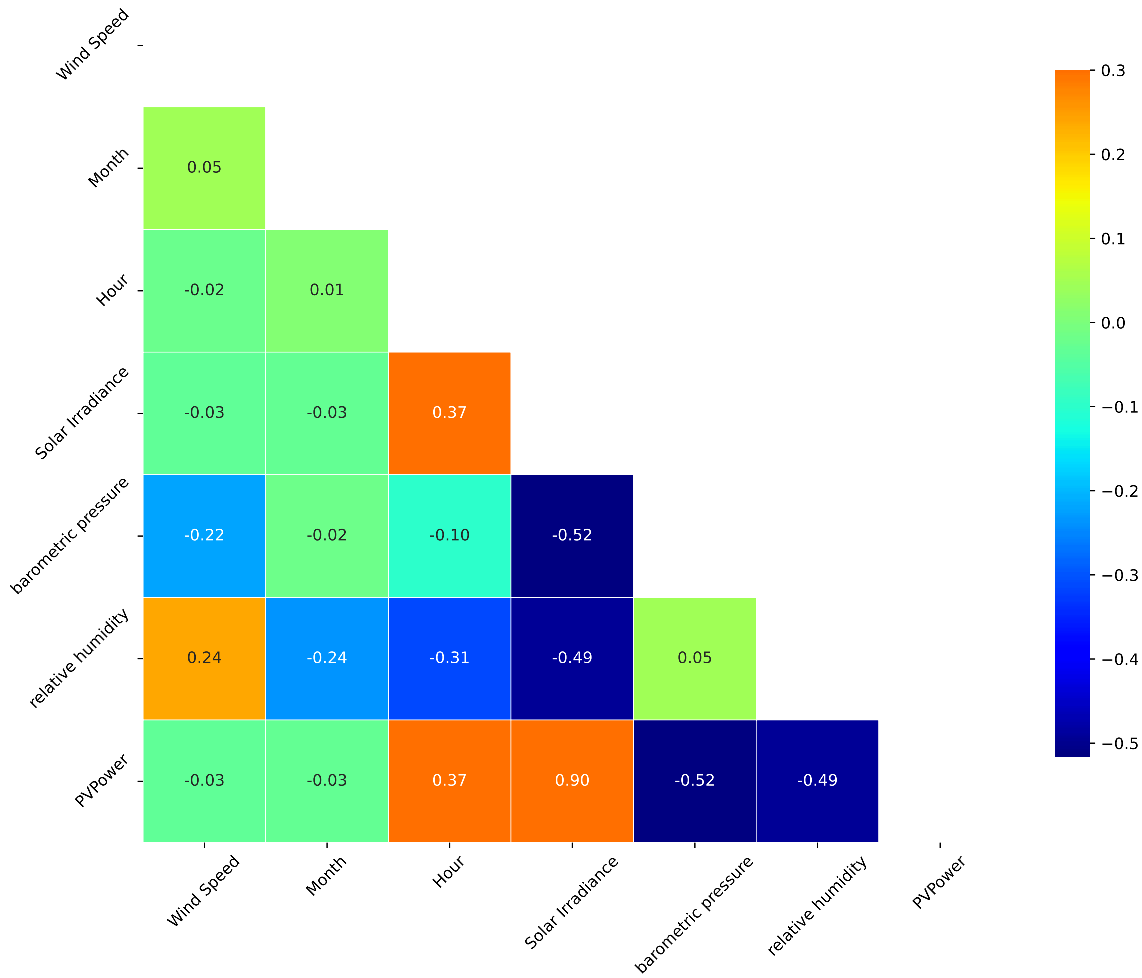


Figure 3.2: Correlation Analysis of Solar Power Features

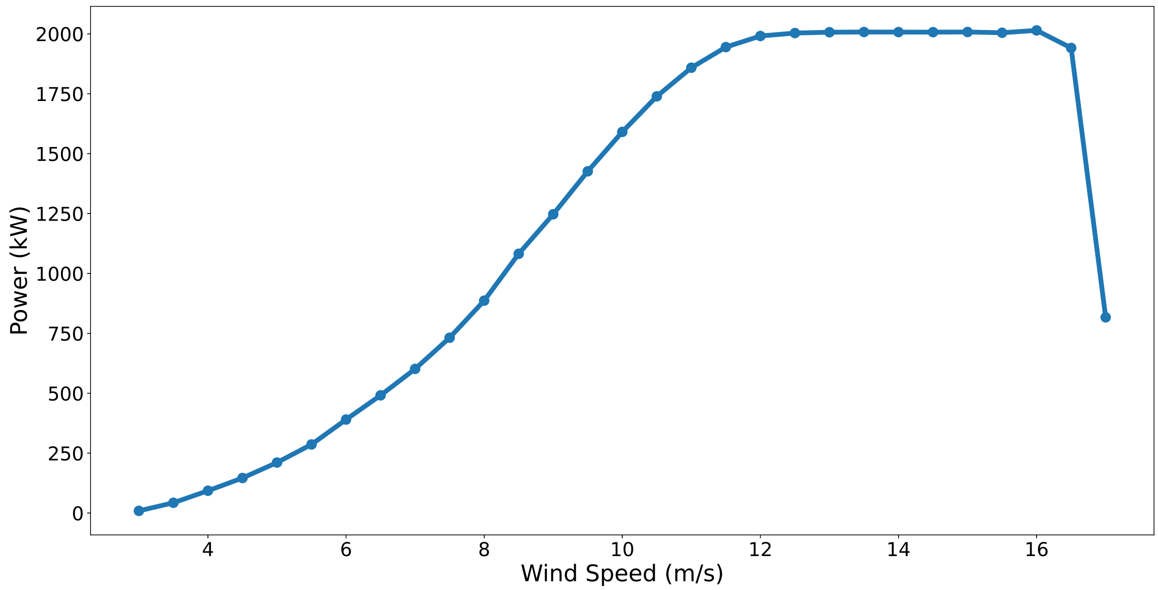


Figure 3.3: Power Curve of Vestas V90 Wind Turbine

generator for Vestas V90 is asynchronous with an output voltage of 690.0 V. Figure 3.3 depicts the power curve of Vestas V90, in which the data[62] used to plot the graph is provided by the manufacturer. The average wind speed per month for Ga-Nkoane, a rural area in South Africa, is presented in Figure 3.4, which will be used to assess the wind turbine power output.

The solar panel used is the Tenesol TE2200 - 220W [63]. This type of solar panel was developed for different applications, including rural electrification. As a result, it is an excellent choice for a solar panel in our analysis.

3.3.2 Data Processing

We have conducted all data handling and mathematical computation in this chapter using Python version 3.9.13 on Jupyter Notebook. We have chosen this programming language for its reliability and efficiency in data analysis and the built-in packages for the machine learning algorithms. It can perform a wide range of complex computations and visualize the results in an organized and understandable manner. Using this language also allows for reproducibility and transparency in our methods, as the Python code is open-source and accessible to others. All of the

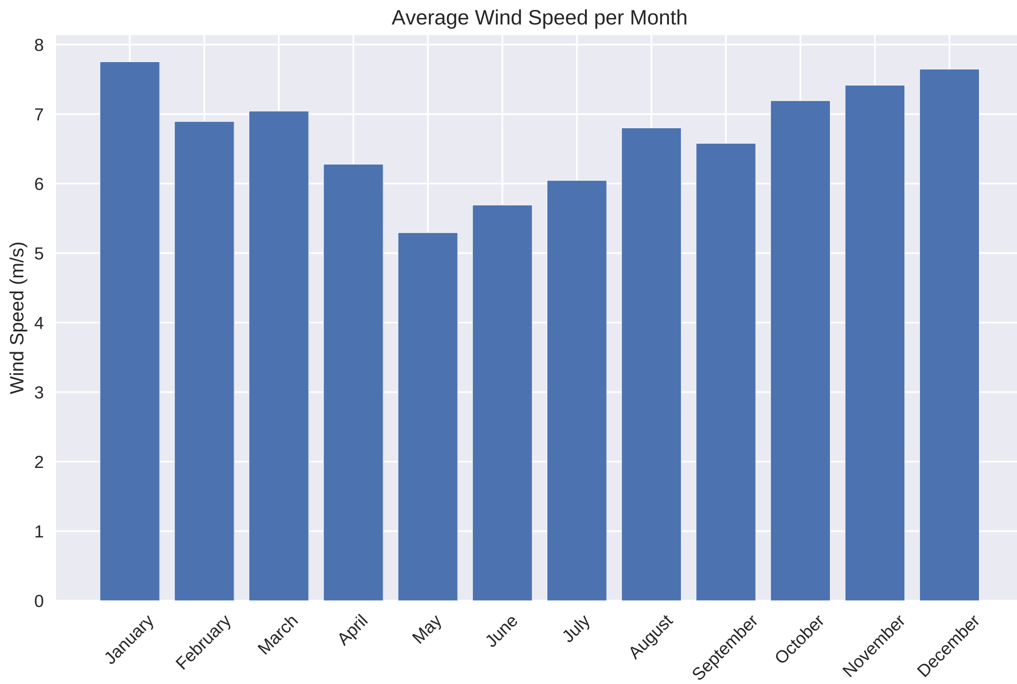


Figure 3.4: Average Wind Speed per Month for Ga-Nkoane

regression algorithms used in this study were sourced from scikit-learn [64], except LSTM, which is obtained from keras [65].

For preprocessing of time series data, we used the tsExtract library [66]. This library provides the conversion of time series data to a suitable form and then passes it to a standard machine learning regression algorithm.

We have excluded data from periods when solar panels are unlikely to generate power since they typically produce electricity between 9 a.m. to 3 p.m.

3.3.3 Cross-Validation

Cross-validation is a technique used in machine learning to evaluate the model's performance on unseen data. It is commonly used in time series data to assess the ability of a model to make accurate predictions on future time points based on past observations.

In time-series cross-validation, we split data into training and test sets. The training set is used to fit the model after preprocessing step, while the test set is used to evaluate its performance. The process is repeated with different partitions of the data, allowing for a more robust estimate of the model's performance.

One common approach for time-series cross-validation is called rolling window cross-validation. The rolling window cross-validation technique divides the data into fixed-size windows, and the model is trained using one window and tested using the following one. This procedure is carried out continuously until the whole dataset has been used for training and testing.

Another approach for time-series cross-validation is expanding window cross-validation. In this method, the model is trained on the earliest data points and tested on later data points, gradually increasing the size of the training set as more data becomes available. This method allows a more realistic evaluation of the model's capability for handling large amounts of data. It is worth noting that when working with time series data, it is essential to use cross-validation techniques that preserve the temporal ordering of the data. In other words, the training and test sets should be contiguous and non-overlapping time series data to avoid data leakage. The sequential structure of time-series data does not allow the data points to be mixed or shuffled. Thus, any splitting of time-series data should preserve sequentiality, so the data will not lose its regular structure.

For splitting the dataset, we have used a time series with ten-fold cross-validation. Cross-validation helps by estimating the model parameters automatically. A suitable loss function tells how closely the model approximates the data. We have used the negative mean squared error as a loss function. Then, by cross-validation, the chosen loss function evaluates the given model parameters by calculating their gradient. The model parameters keep adjusting until descending to the global minimum.

When performing cross-validation on time series data, a slightly different approach is necessary to avoid mixing the data. In our study, we utilized the expanding window cross-validation method. This procedure involves training the model on a small segment of data from the beginning of the dataset up until a certain point, denoted as s. The model then predicts data from $s + 1$ to $s + m$ and calculates the error. The training sample is then extended to include the data from time steps $s + m + 1$ to $s + 2 \times m$, with a step size of m, and the test segment is moved along the dataset

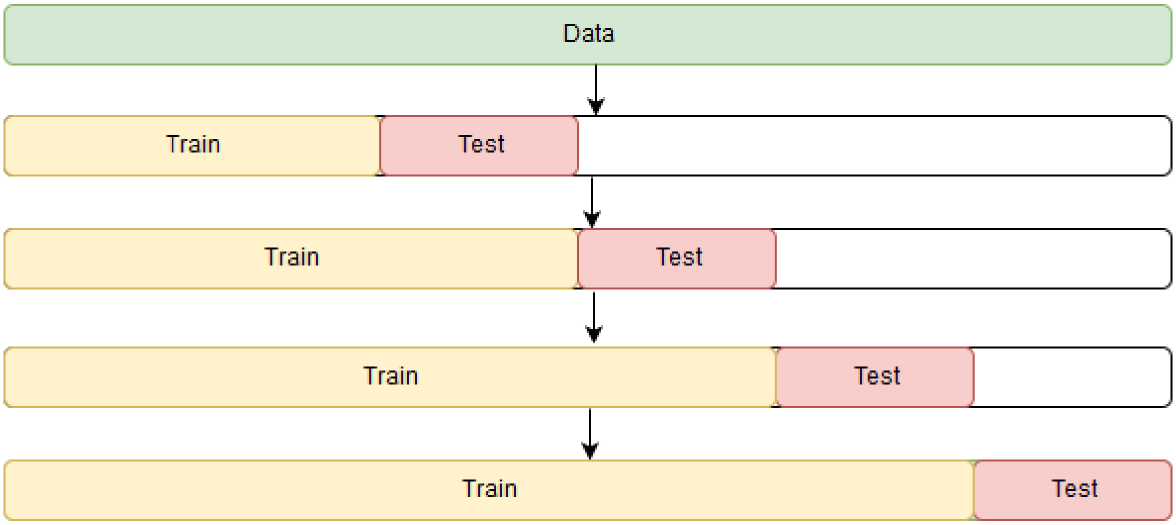


Figure 3.5: Time-Series Cross-Validation

until we reach the end, where each training and testing is based on K-fold. Figure 3.5 shows the Cross-Validation process of time-series data.

3.3.4 Performance Metrics

The objective of this section is to utilize machine learning techniques to predict the power output of wind turbines and solar energy systems. The study began with cleaning and preprocessing the collected data. Then, we trained various regression models and evaluated them using K-fold cross-validation on a time-series dataset. Four performance metrics, Mean Square Error (MSE), Rooted Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), are employed to assess the accuracy of the models. These metrics are used to evaluate the model's ability to predict the test data (actual data) denoted by the symbol y, compared to the predicted data denoted by $\hat{y}$. In this study, y and $\hat{y}$ represent the actual and predicted turbine power output, respectively. Also, they represent the actual and predicted solar power output. A smaller value of error indicates a better-performing model. Cross-validation is a prevalent method in time series data that is used to evaluate the efficiency of different machine learning models. It

Table 3.1: Performance Metrics used in Assessing Regression

Name	Expression
MSE	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
MAE	$\sum_{i=1}^{n}\frac{\lvert y_i - \hat{y}_i\rvert}{n}$
MAPE	$\frac{1}{n}\sum_{i=1}^{n}\frac{\lvert y_i - \hat{y}_i\rvert}{\lvert y_i\rvert}$

allows the model to be trained and tested on different subsets of the data. Doing so helps to prevent overfitting and gives a more accurate representation of the model's performance on unseen data.

The performance evaluation metrics are defined in Table 3.1, where n denotes the number of data used for training or testing.

3.4 Machine Learning Algorithms

In this chapter, we have used several machine learning algorithms for extensive comparative study. We utilized eleven regression techniques, five using mathematical models, the same number of ensemble and decision tree-based techniques, and one deep learning technique. All the linear response regressions are obtained by solving a version of the least-square problem. Some of these methods incorporate an additional regularization term to achieve desired solutions. The linear response model-based regression techniques that we used are simple linear regression (LR), Ridge regression, LASSO, Bayesian Ridge (BR), and Elastic Net (EN). On the other hand, the ensemble techniques based on decision trees used are the Bagging regressor, extreme gradient boosting (XGB) regressor, gradient boosting regressor (GBM), and random forest (RF) regressor. Also, we used a Decision tree regressor (DTR) as a decision tree; and Long Short-Term Memory (LSTM) as a deep learning algorithm.

3.4.1 Linear Response Models

1. **Simple Linear regression** trains a linear mathematical model for predictions. It constructs the best line(plane or hyper-plane) coefficients by minimizing the sum of the residual squared

between the observed values in the dataset and predicted values. The coefficients of the linear model are given by solving the following least square problem

$$\min_{w} \|Xw - y\|_2^2\,, \tag{3.1}$$

where $X \in \mathbb{R}^{n\times m}, w \in \mathbb{R}^m$, and $y \in \mathbb{R}^n$. The matrix X results from the substitutions of the values of predictor variables (features) in the response model, which is linear in this case. The solution to Equation (3.1) can be written in the following closed form

$$w = \left(X^T X\right)^{-1} X^T y.$$

In the wind turbine case, we used the top correlated features with the wind turbine power output in the linear regression, and they are wind speed (x_1), relative humidity (x_2), and barometric pressure (x_3). Then, the simple linear regression model for the wind turbine power output will be expressed as:

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + \epsilon,$$

where w_0, w_1, w_3 and w_3 are the coefficients, and ϵ is the error term.

Similarly, for the solar power output, we used the top correlated features with the solar power output, and they are solar irradiance (x_1), relative humidity (x_2), barometric pressure (x_3), and hour (x_4). Then, the simple linear regression model for the solar power output will be expressed as:

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + \epsilon.$$

The notation was the same for both cases, but the meaning of the variables was different, as we explained above.

Linear regression is a simple model that can easily be interpreted because of linearity. Also, it is computationally efficient and does not require complex calculations when the dataset is large. Despite the model's advantage, the linear regression is too simple. It can not capture the complexity of the real-world problem data, and the linear assumption between the variables can not always be guaranteed. Furthermore, its severely affected by the outliers resulting in a model that cannot capture the information of the dataset.

2. **Ridge regression** adds a penalty term or a regularization of L_2 norm to the minimization of the residuals in the simple linear regression in Equation (3.1). The penalty term is equal to the sum square of the coefficient. The penalty term is controlled by the coefficient parameter λ, which penalizes the large values of the regularized terms. The cost function of Ridge regression is given by

$$\min_{\hat{w}}(\|Xw - y\|_2^2 + \lambda \|w\|_2^2). \tag{3.2}$$

The benefit of using Ridge regression is to reduce the model complexity and multicollinearity, but it is not suitable for feature reduction.

3. **LASSO regression** is a version of linear regression, which differs from Equation (3.2) in its regularization term. It uses the concept of shrinkage due to the regularization used. In particular, the LASSO cost function is obtained by replacing the penalty term in the cost function of ridge regression with the absolute sum of the coefficient called L_1 norm, which is written as

$$\min_{\hat{w}}(\|Xw - y\|_2^2 + \lambda \|w\|_1). \tag{3.3}$$

This type of regression is well-suited for models with a high level of multicollinearity. Also, when automating some parts of the model, like variable selection. The shortcoming of using LASSO is that when there are two highly collinear variables, it selects one randomly, which makes the interpretation of the data difficult.

4. **Elastic Net Regression** is a type of linear regression that combines L_2 regularization in (3.2) and L_1 regularization in (3.3). This combination allows the models to inherit both properties of Lasso and Ridge regressions. The method is helpful if there are collinear features. Elastic Net tends to pick both correlated features instead of picking one randomly, like LASSO regression. The cost function is given by

$$\min_{\hat{w}} \left(\|Xw - y\|_2^2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2 \right). \tag{3.4}$$

5. **Bayesian Ridge Regression** is a powerful technique for analyzing poorly distributed data. Unlike traditional linear regression, which uses point estimators, Bayesian Ridge Regression employs probability distributions to form linear regressions with regularization parameters. This approach allows for recovering the entire range of inferential solutions rather than just a point estimate and a confidence interval. One key difference between Bayesian Ridge Regression and other types of regression is that the vector of coefficients in traditional regressions is fixed, while in this method, the coefficients are random variables following a normal distribution. The model has an explicit prior on the parameters and the choice of the prior results in a regularizing effect. After specifying the prior, the maximum posterior estimate is obtained by solving the maximum log-normal likelihood as follows

$$\begin{aligned}\hat{w} &= \arg\max_{w} \log \left[\exp\left(-\frac{1}{2\sigma^2}(Y - Xw)^T(Y - Xw) - \frac{1}{2\tau^2} w^T w \right) \right] \\ &= \arg\min_{w} \left(\frac{1}{\sigma^2} \|Y - Xw\|_2^2 + \frac{1}{\tau^2} \|w\|_2^2 \right).\end{aligned}$$

If we set $\lambda = \frac{\sigma^2}{\tau^2}$, then Bayesian approach is equivalent to the traditional ridge regression.

3.4.2 Polynomial Response Models

Equation (3.1) represents a linear model that is capable of preserving the linear relationships between predictor variables, also known as features, and the response variable. However, the relationship between features and the response variable may not be linear. In such cases, nonlinear models can also be used to model these relationships accurately.We note that the objective functions, i.e., (3.1)-(3.4), remain the same.

To study the performance of different models, we considered two non-linear models and evaluated their performances in various model-based regressions, including simple regression, Ridge, LASSO, Elastic Net, and Bayesian Ridge regression.

Quadratic models the relationship between the input variable x and the output variable y as a quadratic function. For example, the quadratic form for the wind turbine regression model is given by

$$y = w_{000} + w_{100}x_1 + w_{010}x_2 + w_{001}x_3 + \cdots w_{111}x_1x_2x_3 + \cdots + w_{200}x_1^2 + w_{020}x_2^2 + w_{002}x_3^2 + \epsilon, \quad (3.5)$$

where $w_{ijk}, 0 \leq i, j, k \leq 2$, are the coefficients, and ϵ is the error term. The quadratic regression for solar power will be represented by four sub-indices because there are four features.

Cubic models the relationship between the input variable x and the output variable y as a cubic function. For example, the cubic form for the wind turbine regression model is given in the following form:

$$y = w_{000} + w_{100}x_1 + w_{010}x_2 + w_{001}x_3 + \cdots w_{111}x_1x_2x_3 + \cdots + w_{300}x_1^3 + w_{030}x_2^3 + w_{003}x_3^3 + \epsilon, \quad (3.6)$$

where $w_{ijk}, 0 \leq i, j, k \leq 3$, are the coefficients, and ϵ is the error term. Similarly, the cubic regression for solar power will be represented by four sub-indices.

Quadratic and cubic models are more flexible than linear regression models, as they can capture more complex relationships between input and output variables. However, they are also more prone to overfitting, which is the tendency of the model to fit the training data too closely and perform poorly on new data.

3.4.3 Decision Tree and Ensemble Algorithms

1. **Decision Tree Regressor (DTR)** for regression problems works by recursively partitioning the input space into smaller and smaller regions [67]. In each partition, the tree assigns an output value, which is the predicted value of the target variable for that region. The goal of the decision tree algorithm is to find the partitioning that minimizes the sum of squared residuals between the predicted and actual values of the target variable.

   To understand the math behind decision trees for regression problems, let us consider the example of predicting wind power output based on wind speed, relative humidity, and barometric pressure. Suppose we have a dataset with n data points, where each data point i is represented by a vector of input features x^i = (wind speed$_i$, relative humidity$_i$, barometric pressure$_i$), and the target variable y_i is the wind power output. The goal is to find a function $f(x)$ that predicts the output of y for any input vector x. The algorithm starts with the entire input space and chooses a split point that divides the data into two subsets. The split point is chosen to minimize the sum of squared residuals between the predicted and actual values of the target variable for each subset. Suppose we choose to split the data based on the wind speed feature, as it has the highest correlation. We split the data into two subsets S_{left} and S_{right}. S_{left} contains all data points with wind speed less than or equal to a certain threshold value t, and S_{right} contains all data points with wind speed greater than t. The predicted value for the target variable in each subset is given by the mean value of the target variable in that subset:

   $$f_{\text{left}} = \text{mean}(y_i|x^i \in S_{\text{left}})$$
   $$f_{\text{right}} = \text{mean}(y_i|x^i \in S_{\text{right}}).$$

   The sum of squared residuals for this split is given by:

   $$RSS = \sum_{x^i \in S_{\text{left}}} (y_i - f_{\text{left}})^2 + \sum_{x^i \in S_{\text{right}}} (y_i - f_{\text{right}})^2.$$

The decision tree algorithm finds the threshold value t that minimizes RSS by trying all possible threshold values for the wind speed feature and selecting the one that results in the lowest RSS. Once the best threshold value for the wind speed feature is found, the algorithm repeats the process on each subset S_{left} and S_{right}. This process continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of data points in a leaf node.

At each leaf node, the predicted value for the target variable is given by the mean value of the target variable for the data points in that region:

$$f_{\text{leaf}} = \text{mean}(y_i | x^i \in \text{leaf node}).$$

The overall prediction function for the decision tree is given by the combination of all the predicted values for each leaf node: $f(x) = f_{\text{leaf}}$, if x belongs to leaf node.

The sum of squared residuals between the predicted and actual values of the target variable for the entire dataset is given by:

$$RSS = \sum (y_i - f(x^i))^2.$$

We will build the model by utilizing the DecisionTreeRegressor from the "sklearn" library. Once we have imported the necessary modules, we will train it on the train data using the fit function. After training, we will generate predicted values $\hat{y}$ by applying the predict function to the test data. This prediction will allow us to see how well our model can generalize to new data. To evaluate the performance of our model, we will measure the difference between the correct labels y and the predicted labels $\hat{y}$ using the metric specified in Table 3.1. By doing so, we can assess the accuracy and effectiveness of the model. We will follow the same procedure for the upcoming ensemble methods.

2. **Bagging** (Bootstrap Aggregating) is a machine learning technique that improves the accuracy and stability of a predictive model by combining the predictions of multiple models trained on different subsets of the training data [67]. In bagging regression, the algorithm creates a set of B bootstrap samples, where each sample contains r data points selected randomly with replacement from the original training data set.

   Again, let us explain it using the wind turbine power output. Suppose we have a dataset with n data points, where each data point i is represented by a vector of input features x^i = (wind speed$_i$, relative humidity$_i$, barometric pressure$_i$), and the target variable y_i is the wind power output. The goal is to find a function $f(x)$ that predicts the output of y for any input vector x. The bagging algorithm creates B different decision trees, each trained on a different bootstrap sample of the original data set. Each decision tree uses a random subset of the input features at each node, chosen independently for each split.

   For each tree b in the ensemble, the algorithm computes the predicted value of the target variable y_i for each data point i in the test set using the following formula:

$$\hat{y}_{i,b} = f_b(x^i),$$

   where $f_b(x^i)$ is the predicted value of the target variable for data point i in tree b. The final predicted value of the target variable for data point i is computed as the average of the predicted values of all trees in the ensemble:

$$\hat{y}_i(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{y}_{i,b}(x).$$

   The accuracy and stability of the bagging regression model depend on the number of trees in the ensemble and the variance of the individual trees. Bagging reduces variance by combining the predictions of multiple models trained on different subsets of the training data, but it does not reduce bias.

The accuracy and stability of the bagging regression model can be evaluated using various metrics as in Table 3.1. The model can be further improved by tuning the hyperparameters, such as the number of trees in the ensemble or the maximum depth of each tree.

3. **Random Forest (RF)** regressor is an ensemble machine learning algorithm used for regression tasks. It is based on DTR and builds multiple decision trees in parallel during the training process by selecting a random subset of the data and features for each tree [67]. It is an extension of the bagging algorithm we have discussed above. The difference between RF and bagging is that in RF, a random subset of features is selected at each node in each decision tree, while bagging uses all features. The final prediction is made by averaging the predictions of all the trees in the forest, which reduces the variance and improves the model's accuracy. The randomness in selecting the data and features helps to prevent overfitting and improves the generalization of the model. The RF regressor can manage linear and non-linear relationships and handle missing values and categorical variables. One of the main advantages of an RF regressor is that it is easy to interpret and understand due to its decision tree structure. In addition, it is also relatively fast to train and predict compared to other ensemble methods.

   To explain how RF work, suppose we have a dataset with n instances and $m = 3$ features: wind speed, relative humidity, and barometric pressure. Our target variable is the wind power output. We want to build an RF model to predict wind power output.

   1. We start by randomly sampling a subset of the data with replacement to create a new dataset of the same size as the original.

   2. We then build a decision tree on this new dataset but only consider a random subset of the features, say $k = 2$, at each node. For example, at the root node, we might consider wind speed and relative humidity but not barometric pressure. For the second tree, there are two features could be wind speed and barometric pressure. We split the

data based on the selected features and repeat this process for each subsequent node until we reach the leaf nodes.

3. We repeat steps 1 and 2 t times to build t decision trees.

4. To predict the output for a new instance, we traverse each decision tree down to a leaf node and record the prediction. We repeat this process for all t trees and take the average prediction as the final output.

The math behind RF involves calculating the prediction for each decision tree and then averaging them. Suppose we have t trees, and we want to predict the output for a new instance x. We denote the prediction for the i-th tree as $f_i(x)$. Then, the final prediction is calculated as follows

$$\hat{y}_i(x) = \frac{1}{t}\sum_{i=1}^{t} f_i(x),$$

where $\hat{y}_i(x)$ is the predicted output for the new instance x.

4. **Gradient Boosting Regressor (GBM)** is an ensemble learning algorithm that adds decision trees to the model iteratively, each one trained to predict the residuals (difference between actual and predicted values) of the previous tree.

Here is a step-by-step explanation of how gradient boosting regressor works, using the wind turbine example:

1. We start by initializing the model with a simple regression tree, which calculates the mean of the target variable for all samples in the training set.

2. We calculate the residuals of the initial prediction by subtracting the predicted value from the actual target values. The residuals represent the error of the initial model.

3. We create a new regression tree to predict the residuals. The tree is trained on the features (wind speed, relative humidity, and barometric pressure), and the residuals are calculated in step 2.

4. We use the new tree to predict the residuals for each sample in the training set. We then add these predicted residuals to the previous prediction to get an updated prediction for the target variable.

5. We repeat steps 2-4 by calculating the residuals of the updated prediction and creating a new regression tree to predict these residuals. We continue this process until the desired number of trees is reached or until the improvement in the model's performance is no longer significant.

6. We combine the predictions of all the trees in the model to get the final prediction for the target variable.

The GBM algorithm involves iteratively adding weak models to a regression ensemble to improve the overall prediction accuracy. At each iteration, a new weak model is fit to the negative gradient of the loss function with respect to the current prediction. The loss function is typically the MSE, defined as:

$$L(y, F) = \frac{1}{2}(y - F)^2,$$

where y is the true target value, and F is the current prediction. The negative gradient of the loss function with respect to F is:

$$r = -\partial \frac{L(y, F)}{\partial F} = y - F.$$

This residual is the difference between the true target value and the current prediction. The negative sign is added to make sure that the weak model is fit to the direction in which the loss function is decreasing.

The new weak model is typically a decision tree with a small number of leaves or depth, called a regression tree. Then, we fit a regression tree to the negative gradient values r_{it} and obtain the tree's prediction $h_t(x^i)$ for each training example, where r_{it} is the residual of the tree t for feature i. Then, we update current prediction $F_t(x)$ by:

$$F_t(x) = F_{t-1}(x) + \nu h_t(x),$$

where ν is the learning rate, which is a hyperparameter that controls the contribution of each new weak model to the overall prediction.

This process is repeated iteratively, with each new weak model fitting to the negative gradient of the current prediction, until a stopping criterion is met, such as the maximum number of iterations or a minimum improvement in the loss function.

5. **The eXtreme Gradient Boosting** regressor (XGB) is a variant of the gradient boosting algorithm that adds additional regularization and optimization techniques to improve performance and reduce overfitting [68]. The main differences between XGB and GBM are the objective function, regularization term, and the way the trees are constructed.

   XGB has a more general form of the loss function compared to GBM, which can be written as [69]:

$$L(y_i, \hat{y}_i) = l(y_i, \hat{y}_i) + \Omega(f)$$
$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\left\|w\right\|^2$$

   where $l(y_i, \hat{y}_i)$ is the loss function, e.g. $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, $\Omega(f)$ is a regularization term, f is the regression tree model, T is the number of leaves, and w is the leaf weights. The regularization term penalizes complex models by adding a penalty term that is a function of the number of leaves. This helps prevent overfitting by discouraging the model from learning noise or irrelevant features in the data.

   The objective function in XGB is a sum of the loss function and the regularization term over all the training samples, plus an additional term that helps to prevent overfitting and stabilize the learning process [68, 69]:

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y}i) + \sum_{k=1}^{K} \Omega(f_k)$$

where K is the number of trees, and γ is a hyperparameter that controls the complexity of the model. The penalty term helps to smooth the output of the trees and prevent them from fitting too closely to the training data.

To train the XGB model, we start with an initial prediction F_0 which is the mean of the target variable. At each iteration t, we fit a new regression tree $h_t(x)$ to the negative gradient of the objective function with respect to the current prediction:

$$r_{it} = -\frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i}\Big|_{\hat{y}i = F_{t-1}(x^i)}.$$

The negative gradient is the residual between the true target value and the current prediction, similar to GBM. The new tree is fit to the residuals using the following objective function [68, 69]:

$$Obj_t = \sum_{i=1}^{n} \left[l(y_i, F_{t-1}(x^i) + h_t(x^i)) + \Omega(h_t) \right].$$

Once the tree is trained, we update the current prediction using the following formula:

$$F_t(x) = F_{t-1}(x) + \eta h_t(x),$$

where η is the learning rate, which controls the contribution of each new tree to the overall prediction. The learning rate helps to slow down the learning process and prevent overfitting.

We repeat this process for a fixed number of iterations or until a stopping criterion is met, such as the maximum depth of the tree or a minimum improvement in the objective function. The final prediction is a weighted sum of the predictions of all the trees:

$$\hat{y}(x) = \sum_{t=1}^{T} \eta h_t(x),$$

where T is the total number of trees in the model. The weights of the trees are determined by the learning rate and the regularization term.

3.4.4 Long Short-Term Memory (LSTM)

An LSTM is a subtype of Recurrent Neural Network (RNN), which is applicable for regression tasks involving sequential data. It detects patterns in data sequences, such as time series data. LSTM networks are designed to handle the problem of vanishing gradients, which is a common issue in traditional RNNs. The LSTM architecture includes memory cells, input gates, output gates, and forget gates that allow the network to selectively remember or forget past information, allowing it to learn long-term dependencies in the data [70]. The architecture of the LSTM is shown in Figure 3.6, for more details see [71].

While other Artificial Neural Network (ANN) models like back-propagation neural networks and gated recurrent units can also be suitable for time series analysis, LSTMs have shown particular strength in capturing long-term dependencies, handling variable-length sequences, and providing efficient memory management.

In regression tasks, LSTM predicts a sequential value based on past observations. The LSTM receives a sequence of inputs and uses its memory cells to store information from previous time steps. Then, this stored information is used to make predictions for the next time step. LSTMs can be trained using a variety of optimization algorithms, such as stochastic gradient descent, and can be fine-tuned using techniques such as dropout and regularization to prevent overfitting. The LSTM regressor architecture is good when dealing with time series data, where the order of the inputs is essential. The ability of LSTMs to learn long-term dependencies in the data allows them to capture patterns and trends that would be difficult for other models to identify.

We fitted the training data of wind and solar power (separately) into an LSTM deep learning model. Firstly, we defined a sequential model from "keras.models", and we added an LSTM layer with 100 units as the first layer. We specified the input shape of this layer to be the shape of the training data. Next, we added a dense layer with one output unit, and we compiled the model using the mean absolute error (MAE) loss function and the Adam optimizer. We then trained

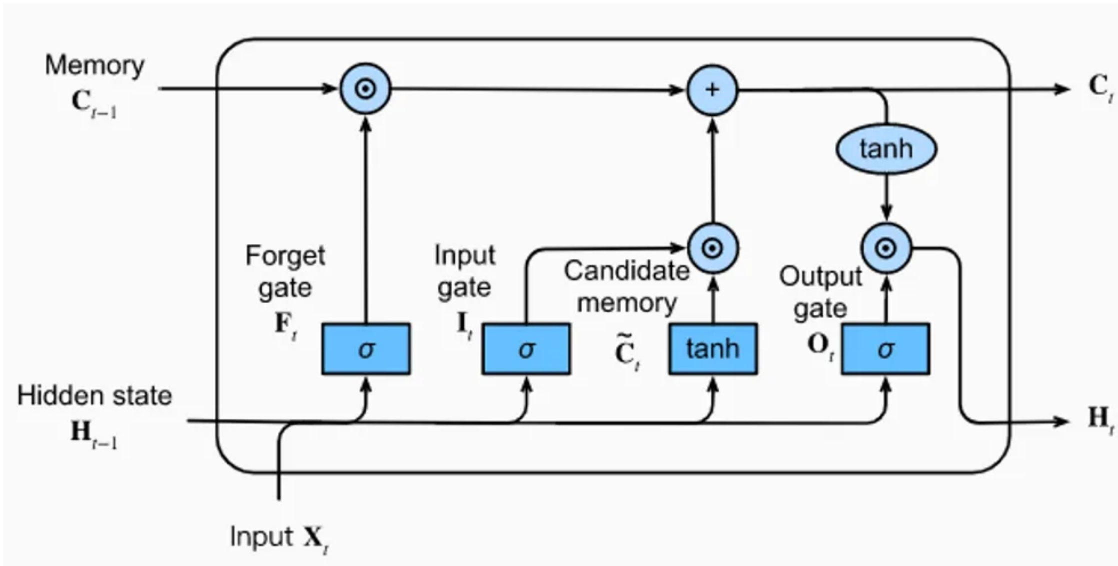


Figure 3.6: LSTM Architecture

the model using the fit function using training data. We also passed the validation data to the fit function for validation during training. We set the shuffle parameter to False to preserve the order of samples during training. Finally, we have used the trained model to predict the output $\hat{y}$ for the test data using the predict function. Also, we have used the metrics in Table 3 to assess the model performance.

3.5 Numerical Experiments

3.5.1 Wind Turbine Power Output Prediction

This section presents the result of the experiment performed using the eleven regression techniques. We have used 10-fold cross-validation for training the models after transforming the training dataset into a standardized form with a mean of zero and a standard deviation of one. We have described the process of cross-validating time series data in Section 3.3. Standardization improves the performance of machine learning models by ensuring that models will be trained more effectively and compared more fairly.

Standardizing a dataset is a common preprocessing step in machine learning, particularly for techniques that rely on distance-based metrics or regularization. The goal of standardization is to

Table 3.2: Performance of Linear Response Models on Wind Turbine Power Prediction

	Evaluation Metric			
Methods	MSE	RMSE	MAE	MAPE
Elastic Net (EN)	26223.12	162.27	120.15	0.78
Bayesian Ridge (BR)	19824.72	140.8	107.44	1.15
LASSO	19823.12	140.8	107.35	1.15
Ridge	19824.01	140.79	107.44	1.15
Linear Regresssion (LR)	19823.88	140.79	107.44	1.15

transform the features of a dataset so that they have the same scale and are centered around zero. Usually, standardizing a dataset before training a machine learning model makes the data scale-independent, ensuring that all features have a similar range of values, which avoids the issues that can arise when different features have different scales. It can help the model converge faster and reduce the risk of overfitting. Also, it can reduce the impact of outliers, which are data points that fall far outside the range of most other data points.

Firstly, we present the evaluation metrics for linear response regression methods on wind turbine power output in Table 3.2 and graphically in Figure 3.7. In Figure 3.7, the size of the errors is shown in the bar chart on the horizontal axis with its corresponding algorithm on the vertical axis. All the linear response methods, except Elastic Net, have similar MSE and RMSE values, while Elastic Net has a bit larger error. Similarly, they have the same MAE and MAPE values. Although they have poor prediction results, we presented them to show that they can not perform well in these prediction problems (wind and solar power predictions). All the methods have almost identical performance with significant errors. The Elastic Net one seems to be the poorest one among them.

Figure 3.8 shows the cross-validation scores in a box-plot format for the linear response models on the wind turbine power prediction. The negative mean squared error is used as a scoring metric. The reason for using this scoring metric is to allow for easier comparison and ranking of different regression techniques. It penalizes models of the regressions more heavily for larger prediction errors, which helps to ensure that the model is not overfitting to the training data and is generalizing

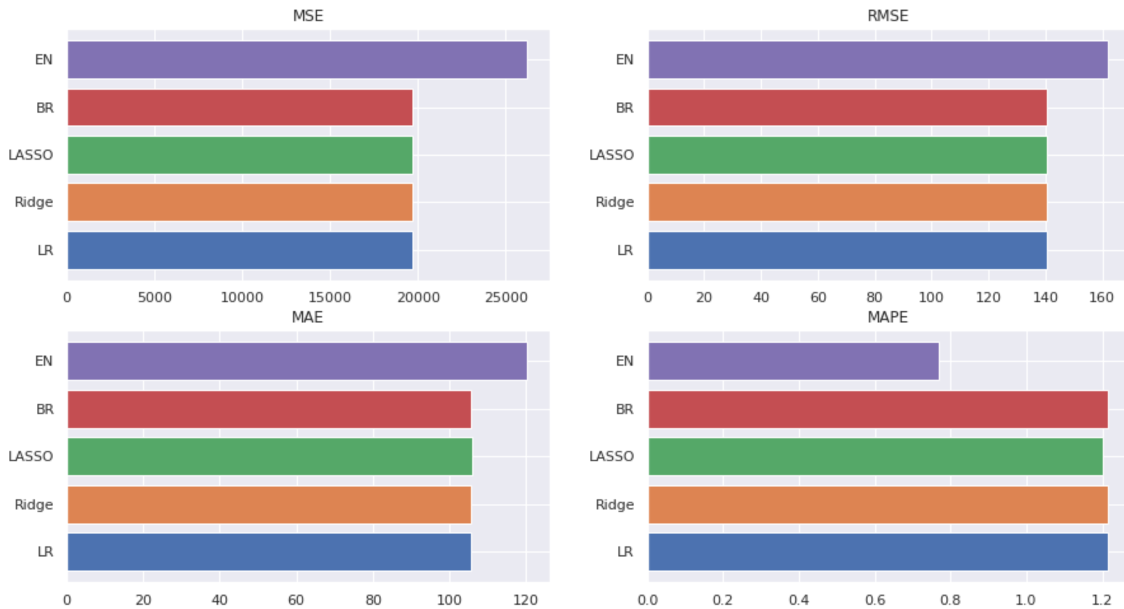


Figure 3.7: Performance of Linear Response Models on Wind Turbine Power Prediction

Table 3.3: Performance of Polynomial Response Models on Wind Turbine Power Prediction

	Evaluation Metric			
Methods	MSE	RMSE	MAE	MAPE
Quadratic Regression	13804.015	117.49	110.52	1.339
Cubic Regression	2262.3373	47.56	36.929	0.706

well to new data. The figure shows that all the models have the same performance except Elastic Net, which has a large standard deviation error. This result matches the results in Table 3.2 and Figure 3.7. We conclude that the linear response models are unsuitable for predicting wind turbine power output.

We have also tested two polynomial-based regressions, Equations (3.5)-(3.6), on wind turbine power prediction. We presented their evaluation metrics in Table 3.3. These results in Table 3.3 are better than those obtained by linear response models in Table 3.2, but still not suitable, and their errors are very high. Increasing the degree of regression yields decreases the prediction error, but still, they have large errors.

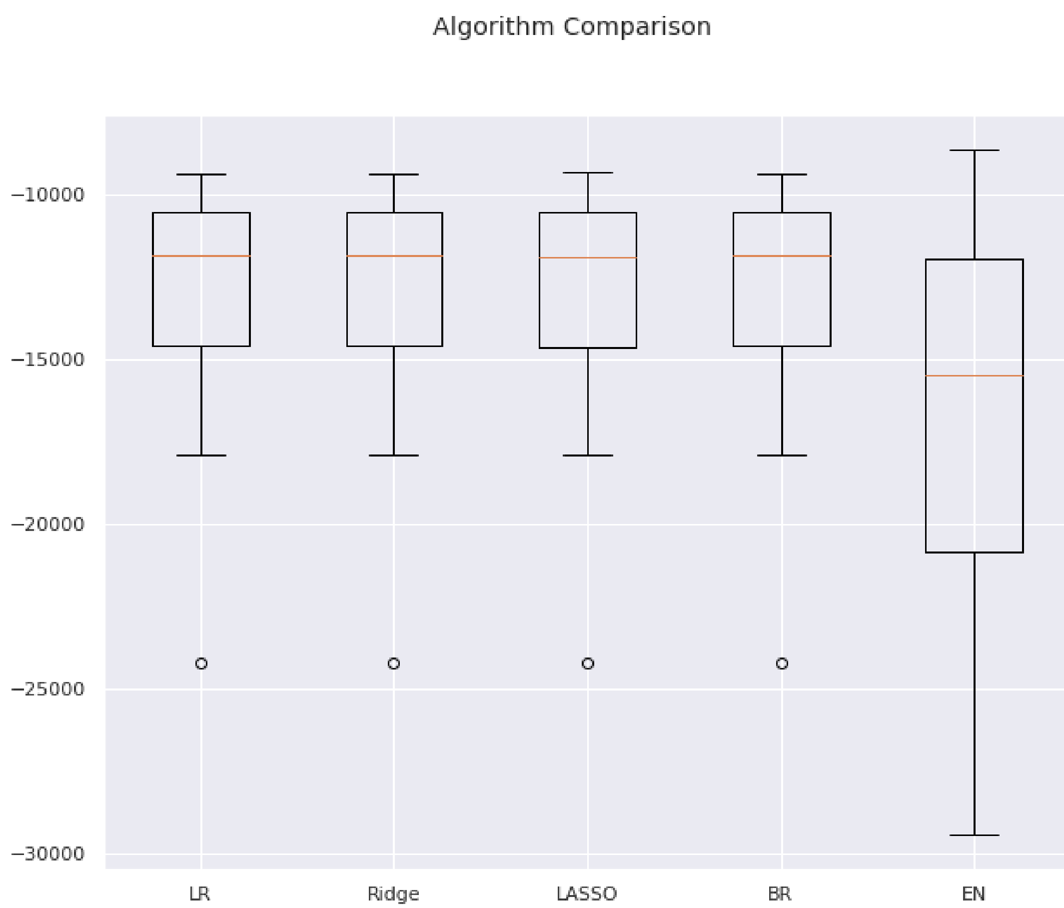


Figure 3.8: Cross-validation Scores of Linear Response Models on Wind Turbine Power Prediction

Figure 3.9 shows the prediction of wind power output using the simple linear regression model. The blue line in the figure represents the actual values of the wind power, while the red line is the predicted values. Although the model can capture the trend, the predicted values have deviated from the actual values, as the evaluation metrics shown in Table 3.2. All the regression methods using the linear response model have almost the same poor performance as the simple linear regression.

On the other hand, Table 3.4 and Figure 3.10 present the evaluation metrics of the decision tree, LSTM, and ensemble methods to the wind turbine power prediction. All these methods show more accurate results than the linear response model-based regressions. The performance of the methods varies from one to another. The LSTM gives a better performance among all the

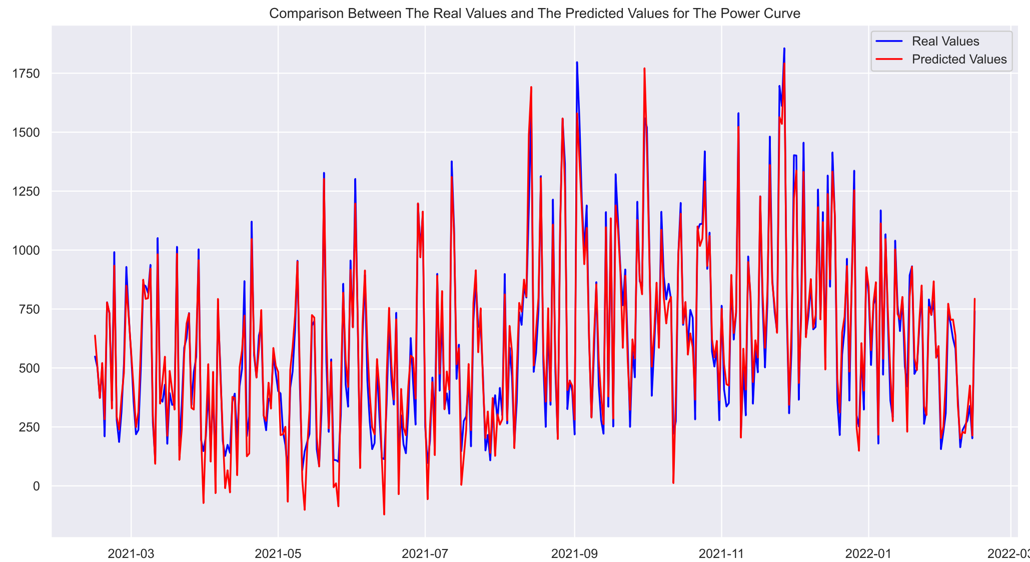


Figure 3.9: Prediction of Wind Turbine Power Output Using Simple Linear Regression

methods. Since RMSE gives a relatively high weight to significant errors, it is useful when large errors are undesirable. Thus, for this prediction problem, it is more suitable to consider RMSE as an evaluation metric, which means that LSTM has superior performance over all the regressors, where its RMSE is 0.915.

Figure 3.11 shows the cross-validation scores in a box-plot format for the decision tree, LSTM, and the ensemble methods. We- have used the negative mean squared error (NMSE) as a score for the cross-validation. By using NMSE instead of MSE, we can maximize the performance metric we are interested in. The figure shows that all the methods have a tight and small standard deviation of the error, while the GBM has the largest one. Among the ensemble models, the random forest achieved better performance than the other models.

Figure 3.12 shows the prediction of power output using the XGB model. The blue line of actual values and the red line of the predicted values almost coincide. Although the data we worked with does not include explicit parameters for cloudy, clear sky, or rain, it does comprise several relevant factors like wind speed, month, hour of the day, wind direction, barometric pressure, relative humidity, and solar irradiance. These variables can still provide valuable insights into the

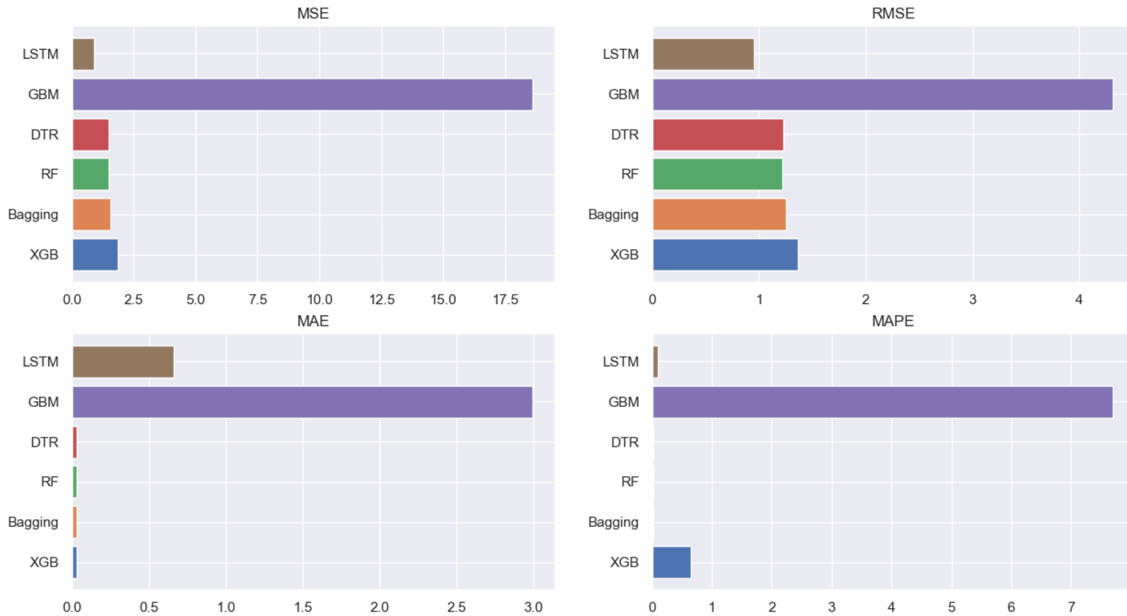


Figure 3.10: Performance of Decision Trees, LSTM, and Ensemble-based Models on Wind Turbine Power Prediction

relationship between weather patterns and the model's performance. The same reasoning applies to solar power predictions.

3.5.2 Solar Power Prediction

Here we present the result of the experiment on solar power prediction using the six regression techniques. We eliminated the linear response model-based and polynomial regressions as they perform poorly. Here, we continue examining the ensemble-based methods, decision trees, and LSTM. We have followed the same procedure by using 10-fold cross-validation for training the models after standardizing the dataset. Moreover, we applied the same performance metrics in Table 3.1.

Table 3.5 and Figure 3.13 present the evaluation metrics results of the decision trees, LSTM, and ensemble methods for solar power prediction. Here too, we found that the LSTM has superior performance; its RMSE is 1.181.

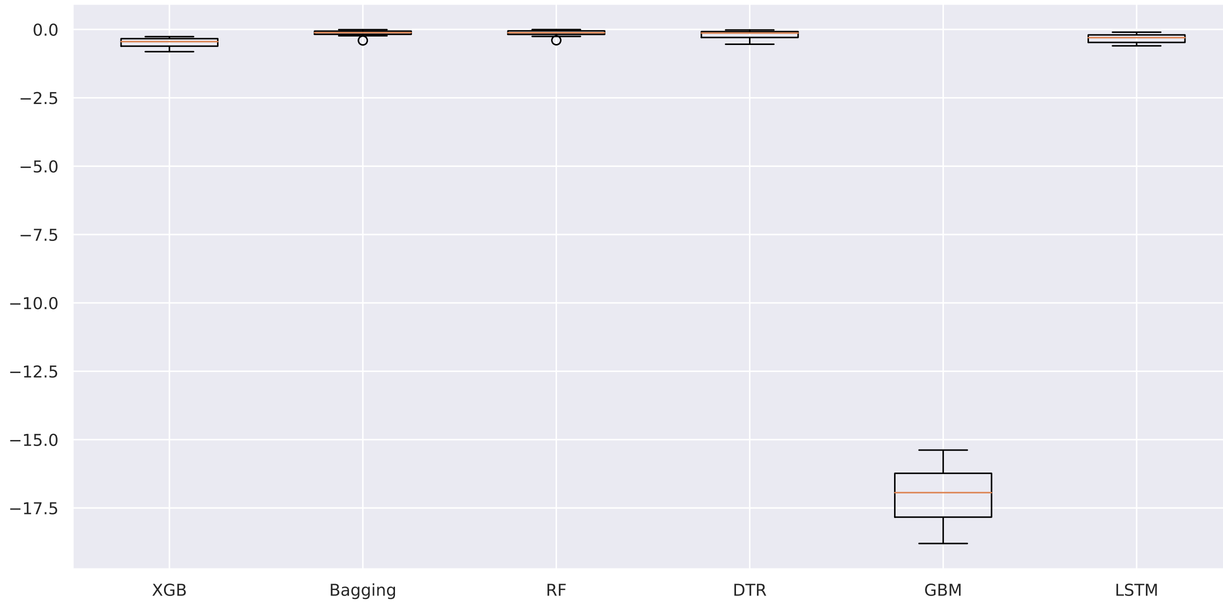


Figure 3.11: Cross-Validation Scores of Decision Trees, LSTM, and Ensemble-based Models on Wind Turbine Power Prediction

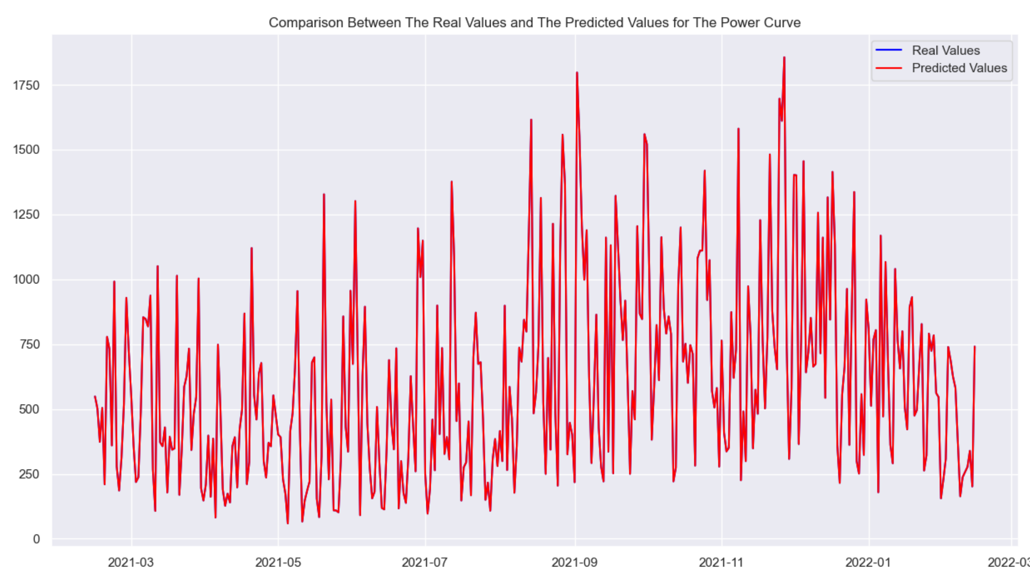


Figure 3.12: Prediction of Wind Turbine Power Output Using XGB

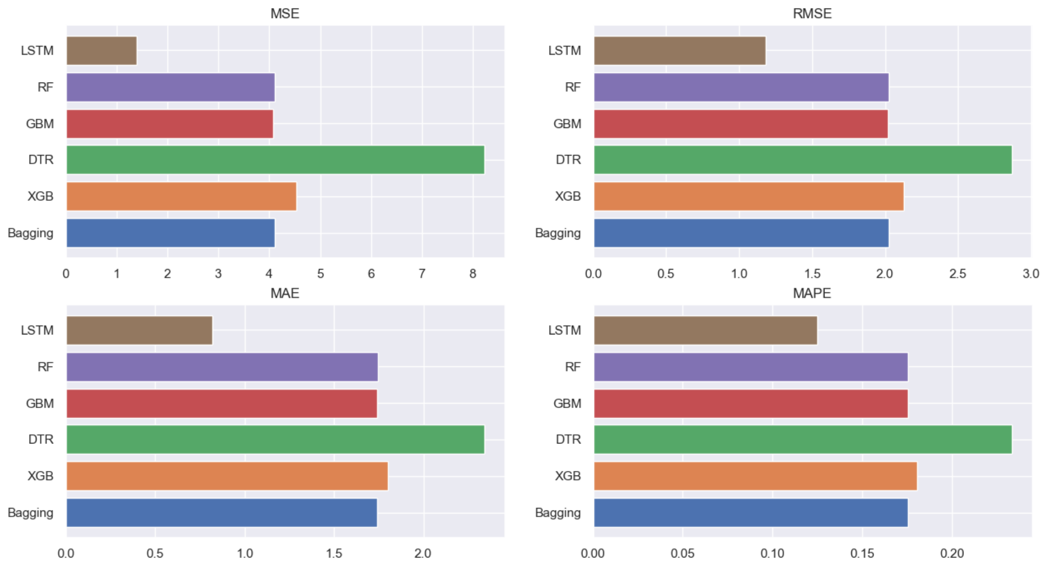


Figure 3.13: Performance of Decision Trees, LSTM, and Ensemble-based Models on PV power Output

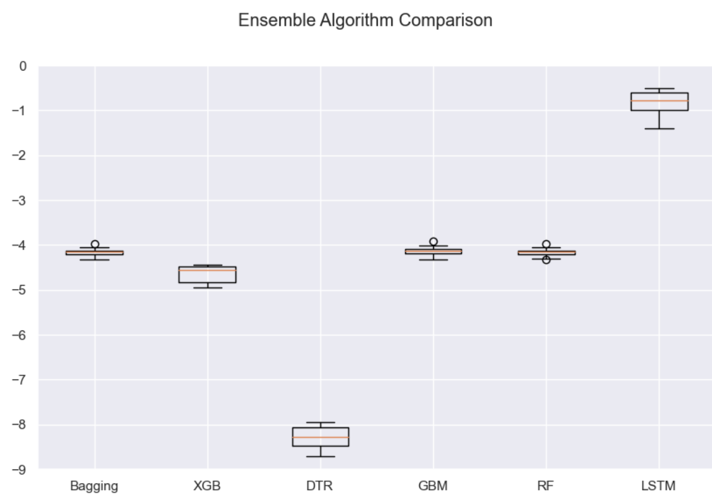


Figure 3.14: Cross-Validation Scores of Decision Tress, LSTM, and Ensemble-based Models on PV Power Prediction

Table 3.4: Performance of Decision Trees, LSTM, and Ensemble-based Models on Wind Turbine Power Prediction

	Evaluation Metric			
Methods	MSE	RMSE	MAE	MAPE
LSTM	0.911	0.951	0.661	0.097
XGB	1.854	1.362	0.442	0.654
Bagging	1.575	1.255	0.031	0.022
RF	1.485	1.210	0.032	0.020
DTR	1.50	1.227	0.031	0.015
GBM	18.59	4.310	2.992	7.695

Table 3.5: Performance of Decision Trees, LSTM, and Ensemble-based Models on Solar Power Prediction

	Evaluation Metric			
Methods	MSE	RMSE	MAE	MAPE
LSTM	1.396	1.181	0.821	0.125
XGB	4.533	2.129	1.802	0.181
Bagging	4.103	2.026	1.743	0.175
RF	4.099	2.024	1.739	0.174
DTR	8.223	2.867	2.340	0.233
GBM	4.079	2.020	1.741	0.175

Figure 3.14 shows the cross-validation scores in a box-plot format for the decision trees, LSTM, and the ensemble models for solar power prediction. We used the same negative mean squared score for the cross-validation. Figure 3.14 shows that LSTM has a small standard deviation of the error, which is the best score, while the decision tree regressor has the largest standard deviation of the error. Among the ensemble models, GBM has better performance than the other models.

Figure 3.15 shows the prediction of PV solar power output using XGB. The figure depicts two lines, one in blue representing the actual power output values and another in red representing the values predicted by XGB. It is immediately evident from the figure that the red line of predicted values aligns closely with the blue line of actual values, indicating that XGB is able to predict the power output accurately. The figure also highlights a period of decreased power output during May, June, and July, which is indicated by a dip in the red line. This drop in power output is due to the fact that these months correspond to the winter season in South Africa, where there is less

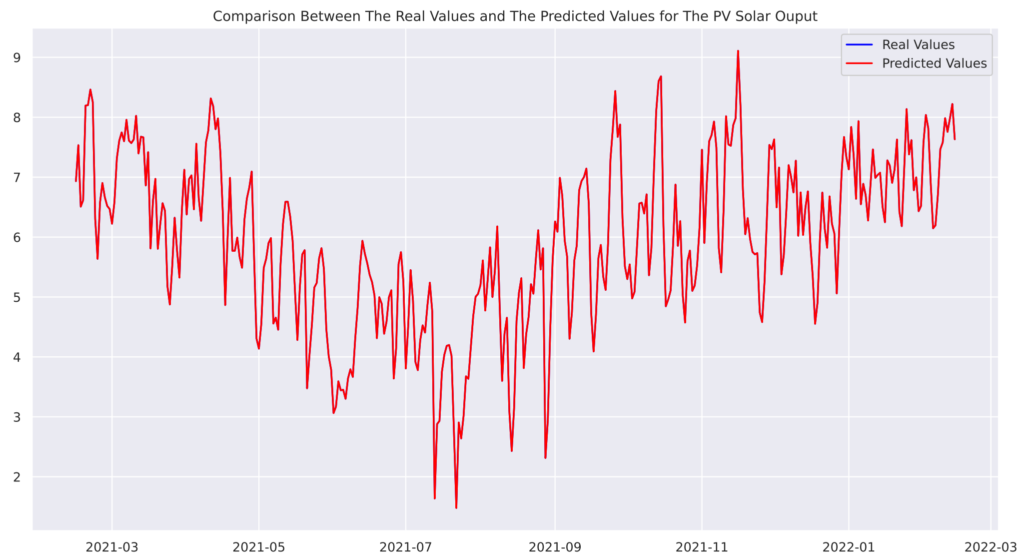


Figure 3.15: Prediction of PV Solar Power Output Using XGB

sunlight available for the PV solar panels to generate power. This reduction in power output is accurately reflected in the red line of the predicted values, further demonstrating the accuracy of XGB. Overall, Figure 3.15 provides valuable insight into the performance of XGB for predicting PV solar power output. The close alignment between the actual and predicted values, even during periods of reduced power output, highlights the potential of the XGB model for accurate and reliable predictions in the field of renewable energy. This information is valuable for decision-makers and engineers working in the renewable energy industry.

3.6 Discussion

This chapter focused on modeling wind and solar power prediction using machine learning techniques. We have used time-series data in the analysis of different models. Various machine learning methods, ranging from linear to ensemble, deep learning (LSTM), and decision trees, were employed to predict the response of the wind turbine and solar power output.

We started our experiment for wind turbine power prediction using linear response-based model regressions. We utilized four evaluation metrics to assess the performance of the models by measuring the distance between the actual observations and the predicted ones. The linear response-based and polynomial models have shown poor results with high errors. On the other hand, the ensembles, LSTM, and decision trees have shown more accurate results. We found that the LSTM stands as the best model for wind turbine power prediction, followed by RF. For solar power prediction, LSTM outperformed the others.

The LSTM is known for its ability to handle sequential data and accurately predict future outcomes based on previous inputs. However, the process of training the LSTM technique can be computationally intensive and requires a longer execution time compared to other techniques. On the other hand, ensemble methods are generally quick to execute and do not require as many computational resources. Despite their speed, ensemble methods can still produce highly accurate predictions. Therefore, if computational resources are limited and execution speed is a concern, an ensemble method may be a better choice. However, if accuracy is the top priority, the LSTM may be the way to go, even though it may take longer to execute. Ultimately, the choice between LSTM and ensemble methods will depend on the specific requirements of the problem at hand.

Our experiments with the machine learning methods show are intelligent in that they predict different types of input data. The machine learning methods used for time-series data for the wind turbine and solar power output are applicable to other data elsewhere in the world. The employed models enjoy desirable properties. They can handle a large dataset with missing data values. Also, it does all the preprocessing to the time-series data to feed into the machine learning techniques.

Chapter 4

Solving Two-stage Stochastic Programming with Three-block ADMM

4.1 Introduction

In recent years, researchers have shown an increased interest in the Alternating Direction Method of Multipliers (ADMM) [72] and its applications. The importance of ADMM arises from its ability to decompose a large and complex optimization problem into separable subproblems, which are solvable in alternating manners. Then, ADMM solves these subproblems parallelly and efficiently by benefiting from the computational advances of the current computers. Because of its simplicity and extensive applicability, countless problems in machine learning form an application field of ADMM, where it is used as a cutting-edge solver.

The emergence of machine learning and big data applications necessitates distributed computation and storage due to large problem sizes. Boyd et al. [73] have applied ADMM on various distributed large-scale convex optimization problems, including the Least Absolute Shrinkage and Selection Operator (LASSO) [74], support vector machine (SVM), and sparse logistic regression. Also, Wang et al. [75] have used ADMM to replace the stochastic gradient descent (SGD) optimizer in their deep learning algorithm and showed it converges to the optimal solution.

The convergence properties and application of the classical ADMM have been reported in the literature [73, 76, 77]. However, the performance of ADMM is strongly dependent on its penalty parameter. Typically, the ADMM's penalty parameter is manually selected by the user making ADMM results inaccurate for non-expert users.

Also, Sun et al. [78] have introduced an intermediate step in the Gauss-Seidel cycle for updating the block coordinate descent for ADMM, which requires more computational time. They have argued that the directly extended ADMM for three-block, a version of the classical ADMM developed by adding a third variable to the objective function and the constraints, is non-convergent. We have used the usual Gauss-Seidel approach where each variable is updated once without adding the intermediate step in Sun et al. [78] approach; hence no extra computation step is needed. Also, we have proved that the directly extended ADMM is convergent under some suitable assumptions in Chapter 5. Then, we applied ADMM to the two-stage SPs with a finite number of scenarios. The formulation is separable and dimension intensive, but the scenario-dependent variables and related dual residuals of ADMM are executable in parallel.

We have studied the convergence analysis of the three-block formulation of ADMM, and implemented it for solving the two-stage SP without any external solver. We have tested our algorithm on a number of problems from the literature [6, 79]. Moreover, we have numerically compared our algorithm with the state-of-art Progressive Hedging (PH) [42, 80] and ADMM algorithm with proximal terms for two-stage stochastic programs (pADMMts) [81] in Chapter 8.

The rest of the chapter is organized as follows. Section 4.2 introduces the three-block minimization problem, and in Section 4.3 we give a brief introduction to the two-stage SP.

4.2 Three-block ADMM Minimization Problem

The classic 2-block ADMM, its numerical implementation and convergence properties have been well documented in the literature [73, 76, 77]. Sun et al. [78] have applied three-block semi-proximal ADMM to a class of convex conic programming with several types of constraints. In this paper, we study the three-block problem for a class of non-convex programming problems using the directly extended ADMM and provide its convergence under some suitable assumptions in Chapter 5.

Consider the three-block minimization problem in the form

$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & f(x_1) + g(x_2) + h(x_3), \quad \text{subject to} \\ & A_1x_1 + A_2x_2 + A_3x_3 = t, \end{aligned} \tag{4.1}$$

where $f : \mathbb{R}^{n_1} \to \mathbb{R}, g : \mathbb{R}^{n_2} \to \mathbb{R}$ are proper lower semi-continuous functions, $h : \mathbb{R}^{n_3} \to \mathbb{R}$ is a linear function; $t \in \mathbb{R}^m$, and A_1, A_2 and A_3 are in appropriate dimensions [82]. The augmented Lagrangian for Problem (4.1) is defined as

$$\begin{aligned} L_\rho(x_1, x_2, x_3, \mu) :&= f(x_1) + g(x_2) + h(x_3) + \mu^T(A_1x_1 + A_2x_2 \\ &\quad + A_3x_3 - t) + \frac{\rho}{2}\left\|A_1x_1 + A_2x_2 + A_3x_3 - t\right\|^2, \end{aligned} \tag{4.2}$$

where $\rho > 0$ is the penalty parameter, and μ is the Lagrange multiplier.

IWe included regularized terms to Problem (4.1). These terms enhance the convergence by reducing the oscillation between progressive iterates [80, 83]. We have used the following iterative procedure with the regularized terms to find the optimal solution to the three-block problem:

$$\begin{aligned} x_1^{k+1} &= \arg\min_{x_1}\left(L_\rho(x_1, x_2^k, x_3^k, \mu^k) + \frac{\lambda}{2}\left\|x_1 - x_1^k\right\|^2\right), \\ x_2^{k+1} &= \arg\min_{x_2}\left(L_\rho(x_1^{k+1}, x_2, x_3^k, \mu^k) + \frac{\lambda}{2}\left\|x_2 - x_2^k\right\|^2\right), \\ x_3^{k+1} &= \arg\min_{x_3}\left(L_\rho(x_1^{k+1}, x_2^{k+1}, x_3, \mu^k) + \frac{\lambda}{2}\left\|x_3 - x_3^k\right\|^2\right), \\ \mu^{k+1} &= \mu^k + \rho(A_1x_1^{k+1} + A_2x_2^{k+1} + A_3x_3^{k+1} - t), \end{aligned} \tag{4.3}$$

where λ is a penalty parameter linked with the regularized terms. The augmented regularized term added to each subproblem in (4.3) eliminates the need for a convex objective function [82].

Boyd et al. [73] state the necessary and sufficient optimality conditions for Problem (4.1) as follows

$$A_1x_1^* + A_2x_2^* + A_3x_3^* - t = 0, \tag{4.4}$$

$$0 \in \partial f(x_1^*) + A_1^T\mu^*, \tag{4.5}$$

$$0 \in \partial g(x_2^*) + A_2^T\mu^*, \tag{4.6}$$

$$0 \in \partial h(x_3^*) + A_3^T\mu^*, \tag{4.7}$$

where Equation (4.4) is the primal feasibility, and Equations (4.5)-(4.7) are dual feasibilities. Here, ∂f denotes the subdifferential set of f, see [82].

Provided that x_3^{k+1} minimizes $L_\rho(x_1^{k+1}, x_2^{k+1}, x_3, \mu^k)$, we have

$$\begin{aligned} 0 &\in \partial h(x_3^{k+1}) + A_3^T \mu^k + \rho A_3^T (A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t) \\ &= \partial h(x_3^{k+1}) + A_3^T \mu^{k+1}, \qquad \text{by (4.3)}, \end{aligned}$$

which means that x_3^{k+1} and μ^{k+1} satisfy the duality condition in Equation (4.7).

In the case of x_2^{k+1}, we have

$$\begin{aligned} 0 &\in \partial g(x_2^{k+1}) + A_2^T \mu^k + \rho A_2^T (A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k} - t) \\ &= \partial g(x_2^{k+1}) + A_2^T \mu^{k+1} + s_1^{k+1}, \end{aligned}$$

where

$$s_1^{k+1} = \rho A_2^T A_3 (x_3^k - x_3^{k+1}), \tag{4.8}$$

is the residual for the dual feasibility.

By repeating the same steps, we find the residual associated with x_1^{k+1} as

$$s_2^{k+1} = \rho A_1^T \left[A_2 (x_2^k - x_2^{k+1}) + A_3 (x_3^k - x_3^{k+1}) \right]. \tag{4.9}$$

At iteration $k+1$, we denote

$$r^{k+1} = A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t = \frac{1}{\rho} \left(\mu^{k+1} - \mu^k \right) \tag{4.10}$$

as *primal residual* and s_1^{k+1}, s_2^{k+1} as *dual residuals*. We will see in Section 5 that all the residuals converge to zero as $k \to \infty$.

4.3 Two-stage SP

The two-stage SP is defined as [6]

$$\begin{aligned} \min_{x} \quad & c^T x + \mathbf{E}_\omega \mathcal{Q}(x,\omega), \quad \text{subject to} \\ & Ax = b, \quad x \geq 0, \quad x \in \mathbb{R}^{n_1}, \end{aligned} \tag{4.11}$$

where $c \in \mathbb{R}^{n_1}$, $A \in \mathbb{R}^{m_1 \times n_1}$, $b \in \mathbb{R}^{m_1}$, are constants and ω is a random variable of a known distribution. The term, $\mathcal{Q}(x,\omega)$, outlines the second-stage optimum value and is defined as [42]

$$\begin{aligned} \mathcal{Q}(x,\omega) = \min_{y} \quad & q(\omega)^T y(\omega), \quad \text{subject to} \\ & T(\omega)x + W(\omega)y(\omega) = h(\omega), \\ & y(\omega) \geq 0, \quad y(\omega) \in \mathbb{R}^{n_2}, \end{aligned} \tag{4.12}$$

where $q(\omega) \in \mathbb{R}^{n_2}$, $T(\omega) \in \mathbb{R}^{m_2 \times n_1}$, $W(\omega) \in \mathbb{R}^{m_2 \times n_2}$, $h(\omega) \in \mathbb{R}^{m_2}$ encode the random variable data.

The value of the first-stage variable, x, needs to be determined before any future revealing of the random variable ω; y is known as the second-stage decision variable, which corresponds to the decisions after the realizations of the random variable revealed; y is also known as recourse variable since it compensates any bad decisions that may occur at the first-stage.

When the random variable ω has finitely many realizations, or scenarios $S = \{\omega_1, \omega_2. \dots, \omega_s\}$, with corresponding probabilities $\{p_1, p_2, \dots, p_s\}$, Problem (4.11) can be reformulated as a large linear programming (LP) problem [84] in the form

$$\begin{aligned} \min_{x,y} \quad & c^T x + \sum_{i=1}^{s} p_i q_i^T y_i, \quad \text{subject to} \\ & Ax = b, \\ & T_i x + W_i y_i = h_i, \quad i = 1, 2, \dots, s, \\ & x, y_i \geq 0, \qquad i = 1, 2, \dots, s, \end{aligned} \tag{4.13}$$

where $\omega = \omega_i$ is i-th the scenario.

Cutting-plane methods, e.g., the L-shaped method, are used to solve (4.13) when the recourse matrix W is not dependent on the random variable ω. When the recourse matrix depends on the random variable, a different method is needed, e.g., Progressive Hedging. We have taken a different approach that can deal with the recourse matrix of either type. Problem (4.13) can be equivalently restated as

$$\min \quad c^T x + I(\hat{x}) + \sum_{i=1}^{s} \left(p_i q_i^T y_i + I(\hat{y}_i) \right), \quad \text{subject to} \tag{4.14}$$

$$Ax = b, \tag{4.15}$$

$$x - \hat{x} = 0, \tag{4.16}$$

$$y_i - \hat{y}_i = 0, \qquad i = 1, 2, \ldots, s \tag{4.17}$$

$$T^i x + W^i y_i = h^i, \quad i = 1, 2, \ldots, s, \tag{4.18}$$

where

$$I(z) = \begin{cases} 0, & z \geq 0 \\ +\infty, & \text{otherwise,} \end{cases}$$

is a convex function since it is defined on a convex set [85].

The two indicator functions in (4.14) for the pairs $(\hat{x}, \hat{y}_i)$ compensate for the non-negativity condition of the variables x and y_i in the constrains (4.16) and (4.17), respectively. The new formulation in Equations (4.14)-(4.18) is considered as a three-block ADMM model. The first-stage decision variable x is presented the first block, while the second stage decision variable y_i in the second block, and the third one for the pair $(\hat{x}, \hat{y}_i), i = 1, 2, \cdots, s$.

We present the theoretical convergence of the three-block ADMM in the following Chapter.

4.4 Three-block ADMM for Two-Stage SP

We consider the deterministic version of the two-stage SP (4.14)-(4.18) and the notations used thereof. We redefine the variables x_1, x_2, and x_3 of Equation (4.1) as $x_1 := (y_1, y_2, \cdots, y_s)^T$, $x_2 := (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_s, \hat{x})^T$, and $x_3 := x$ to solve the two-stage SP as three-blocks ADMM. Correspondingly, the functions will be defined as follows:

$$
\begin{aligned}
f(x_1) &:= \sum_{i=1}^{s} p_i q_i^T y_i, \\
g(x_2) &:= \sum_{i=1}^{s} I(\hat{y}_i) + I(\hat{x}), \quad \text{and} \\
h(x_3) &:= c^T x.
\end{aligned}
$$

Furthermore, in order to make the correspondence between problem (4.14)-(4.18) and (4.1), we define the matrices as follows:

$$
A_1 = \begin{pmatrix} 0_{m_1,sn_2} \\ \mathbf{I}_{sn2,sn_2} \\ 0_{n_1,sn_2} \\ W_{sm_2,sn_2} \end{pmatrix}, A_2 = \begin{pmatrix} 0_{m_1,sn_2} & 0_{m_1,n_1} \\ -\mathbf{I}_{sn_2,sn_2} & 0_{sn_2,n_1} \\ 0_{n_1,sn_2} & -\mathbf{I}_{n_1,n_1} \\ 0_{sm_2,sn_2} & 0_{sm_2,n_1} \end{pmatrix}, A_3 = \begin{pmatrix} A_{m_1,n_1} \\ 0_{sn_2,n_1} \\ \mathbf{I}_{m_1,n_1} \\ T_{sm_2,n_1} \end{pmatrix}, t = \begin{pmatrix} b_{m_1} \\ 0_{sn_2} \\ 0_{n_1} \\ h_{sm_2} \end{pmatrix},
$$

where

$$
T_{sm_2,n_1} = \begin{pmatrix} T^1_{m_2,n_1} \\ T^2_{m_2,n_1} \\ \vdots \\ T^s_{m_2,n_1} \end{pmatrix}, W_{sm_2,sn_2} = \begin{pmatrix} W^1_{m_2,n_2} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & W^s_{m_2,n_2} \end{pmatrix}, h_{sm_2} = \begin{pmatrix} h^1 \\ h^2 \\ \vdots \\ h^s \end{pmatrix},
$$

where each $sn_x = s \times n_x$ and $sm_x = s \times m_x$.

The vector of multipliers is defined as

$$\mu = (\alpha, \beta, \gamma_1, \cdots, \gamma_s, \delta_1, \cdots, \delta_s)^T.$$

In what follows, we derive the detailed steps to find the solution to Problem (4.14)-(4.18), and then we present a step by step algorithm for the solution. Consider the following augmented Lagrangian function

$$\begin{aligned} L_\rho(x, \hat{x}, y, \hat{y}) =& c^T x + \beta^T(x - \hat{x}) + I(\hat{x}) + \sum_{i=1}^{s} \left(p_i q_i^T y_i + \gamma_i^T(y_i - \hat{y}_i) + I(\hat{y}_i)\right) \\ & + \alpha^T(Ax - b) + \sum_{i=1}^{s} \delta_i^T(T_i x + W_i y_i - h_i) + \frac{\rho}{2}(\|Ax - b\|^2 \\ & + \|x - \hat{x}\|^2 + \sum_{i=1}^{s}(\|T_i x + W_i y_i - h_i\|^2 + \|y_i - \hat{y}_i\|^2)), \end{aligned} \tag{4.19}$$

where $\beta \in \mathbb{R}^{n_1}$, $\alpha \in \mathbb{R}^{m_1}$, $\gamma_i \in \mathbb{R}^{n_2}$, and $\delta_i \in \mathbb{R}^{m_2}$, $i = 1, 2, \cdots, s$.

Solving $\nabla_x(L_\rho + \frac{\lambda}{2}\left\|x - x^k\right\|^2) = 0$ for x^{k+1}, yields:

$$\begin{aligned} & \nabla_x\left(L_\rho(x, \hat{x}^k, y^k, \hat{y}^k) + \frac{\lambda}{2}\left\|x - x^k\right\|^2\right) = c + \beta + A^T\alpha + \sum_{i=1}^{s} T_i^T \delta_i + \\ & \rho\bigg(A^T(Ax - b) + (x - \hat{x}^k) + \sum_{i=1}^{s} T_i^T(T_i x + W_i y_i^k - h_i)\bigg) + \lambda(x - x^k) = 0 \\ & - 2\lambda x^k + c + \beta + A^T\alpha + \sum_{i=1}^{s} T_i^T\delta_i + \rho\bigg(- A^T b - \hat{x} + \sum_{i=1}^{s} T_i^T(W_i y_i - h_i)\bigg) + \\ & \rho\bigg(A^T A + \mathbf{I}(1 + \frac{\lambda}{\rho}) + \sum_{i=1}^{s} T_i^T T_i\bigg)x = 0, \end{aligned}$$

where $\mathbf{I} \in \mathbb{R}^{n_1 \times n_1}$ is identity matrix.

Since $A^T A$ and $\sum_{i=1}^{s} T_i^T T_i$ are symmetric matrices, they have n_1 real eigenvalues and a complete set of orthonormal eigenvectors [86]. Since $\mathbf{I}(1 + \frac{\lambda}{\rho})$ is also a symmetric matrix, it has the same set of eigenvalues as $A^T A$ and $\sum_{i=1}^{s} T_i^T T_i$. Therefore, the sum of these three matrices has n_1 distinct non-zero eigenvalues and thus is invertible. Then, solving for x at iteration $k + 1$ give the following equation

$$
\begin{aligned}
x^{k+1} = &\left(A^TA + \mathbf{I}(1+\frac{\lambda}{\rho}) + \sum_{i=1}^{s} T_i^T T_i\right)^{-1} \left(A^Tb + \hat{x}^k + \sum_{i=1}^{s} T_i^T(h_i - W_i y_i^k) + \right. \\
&\left. \frac{-1}{\rho}(-\lambda x^k + c + \beta + A^T\alpha + \sum_{i=1}^{s} T_i^T\delta_i)\right).
\end{aligned} \tag{4.20}
$$

Similarly, $\nabla_{y_i}(L_\rho + \frac{\lambda}{2}\left\|y_i - y_i^k\right\|^2) = 0, \quad i = 1, 2, \cdots, s$, yields:

$$
\nabla_{y_i} L_\rho(x, \hat{x}, y, \hat{y}) = p_i q_i + \gamma_i + W_i^T\delta_i + \; + \rho\left(W_i^T(T_i x^k + W_i y_i^k - h) + (y_i - \hat{y_i}^k)\right) + \lambda(y_i - y_i^k) = 0
$$

$$
-\lambda y_i^k + p_i q_i + \gamma_i + W_i^T\delta_i + \rho\left(W_i^T(T_i x^k - h_i) - \hat{y_i}^k\right) + \rho\left(W_i^T W_i + \mathbf{I}(1+\frac{\lambda}{\rho})\right) y_i = 0.
$$

where $\mathbf{I} \in \mathbb{R}^{n_2 \times n_2}$. Similarly, the matrix $W_i^T W_i + \mathbf{I}(1+\frac{\lambda}{\rho})$ is invertible. Then, solving for y_i at iteration $k+1$ give the following equation

$$
\begin{aligned}
y_i^{k+1} = &\left(W_i^T W_i + \mathbf{I}(1+\frac{\lambda}{\rho})\right)^{-1} \left(W_i^T(h_i - T_i x^k) + \hat{y}_i^k + \frac{-1}{\rho}(-\lambda y_i^k + \right. \\
&\left. p_i q_i + \gamma_i + W_i^T\delta_i)\right), \quad i = 1, 2, \cdots, s.
\end{aligned} \tag{4.21}
$$

And by differentiating $(L_\rho + \frac{\lambda}{2}\left\|\hat{x} - \hat{x}^k\right\|^2)$ and $(L_\rho + \frac{\lambda}{2}\left\|\hat{y}_i - \hat{y}_i^k\right\|^2)$ with respect to $\hat{x}, \hat{y}_i$ respectively yields

$$
\begin{aligned}
\hat{x}^{k+1} &= \max\left\{\frac{\beta + \rho x^k + \lambda\hat{x}^k}{\rho + \lambda}, 0\right\}, \\
\hat{y_i}^{k+1} &= \max\left\{\frac{\gamma_i + \rho y_i^k + \lambda\hat{y}_i^k}{\rho + \lambda}, 0\right\}, \quad i = 1, 2, \cdots, s.
\end{aligned}
$$

The updates for the dual variables will be expressed as

$$
\begin{aligned}
\beta &= \beta + \rho\left(x - \hat{x}\right), \\
\alpha &= \alpha + \rho\left(Ax - b\right), \\
\gamma_i &= \gamma_i + \rho\left(y_i - \hat{y}_i\right), \\
\delta_i &= \delta_i + \rho\left(T_i x + W_i y_i - h_i\right), \quad i = 1, 2, \cdots, s.
\end{aligned} \tag{4.22}
$$

Algorithm 2 Solving two-stage SP using ADMM

Precondition: $\rho_1 > 0$, λ, $k_{max} > 0$, $\epsilon > 0$. All the initialization to the variables $\hat{x}^1, y_i^0, \hat{y}_i^1, i = 1, \cdots s$, are obtained by first phase Linear Programming (LP) step, while $\alpha^1, \beta^1, \delta_i^1, \gamma_i^1, i = 1, \cdots, s$, are random variables following the standard normal distribution.

$x^1 = \arg\min_x L_{\rho_1}(x, \hat{x}^1, y_i^0, \hat{y}_i^1)$
$k \leftarrow 1$
while True **do**
  for $i \leftarrow 1$ to s **do**
    $y_i^k = \arg\min_y \left(L_{\rho_k}(x^k, \hat{x}^k, y, \hat{y}_i^k) + \frac{\lambda}{2} \left\| y_i - y_i^{k-1} \right\|^2 \right)$
  end for
  $\hat{x}^{k+1} = \max\{\frac{\beta + \rho_k x^k + \lambda \hat{x}^k}{\rho_k + \lambda}, 0\}$
  $\hat{y_i}^{k+1} = \max\{\frac{\gamma_i + \rho_k y_i^k + \lambda \hat{y}_i^k}{\rho_k + \lambda}, 0\}, \quad i = 1, 2, \cdots, s.$
  $x^{k+1} = \arg\min_x \left(L_{\rho_k}(x, \hat{x}^{k+1}, y_i^k, \hat{y}^{k+1}) + \frac{\lambda}{2} \left\| x - x^k \right\|^2 \right)$
  $\alpha^{k+1} \leftarrow \alpha^k + \rho_k \left(Ax^{k+1} - b \right)$
  $\beta^{k+1} \leftarrow \beta^k + \rho_k \left(x^{k+1} - \hat{x}^{k+1} \right)$
  $\delta_i^{k+1} \leftarrow \delta_i^k + \rho_k \left(T_i x^{k+1} + W_i y_i^{k+1} - h_i \right), \quad i = 1, \cdots, s.$
  $\gamma_i^{k+1} \leftarrow \gamma_i^k + \rho_k \left(y_i^{k+1} - \hat{y}_i^{k+1} \right), \quad i = 1, \cdots, s.$
  $r_0 \leftarrow Ax^{k+1} - b$
  $r_i \leftarrow T_i x^{k+1} + W_i y_i^{k+1} - h_i, \quad i = 1, \cdots, s$
  $s_i^1 \leftarrow \rho_k (x^{k+1} - x^k)$
  $s_0^2 \leftarrow \rho_k (\hat{x}^k - \hat{x}^{k+1})$
  $s_i^2 \leftarrow \rho_k (\hat{y}_i^k - \hat{y}_i^{k+1})$
  $r = \left(r_0, r_1, \ldots, r_s, \right)^T, s^1 = \left(s_1^1, s_2^1, \ldots, s_s^1, \right)^T, s^2 = \left(s_0^2, s_1^2, \ldots, s_s^2, \right)^T$

$$\rho_{k+1} = \begin{cases} \nu \rho_k, & r_k > \mu \max\{s_k^1, s_k^2\} \\ \rho_k / \nu, & \min\{s_k^1, s_k^2\} > \mu r_k \\ \rho_k, & \text{otherwise}, \end{cases}$$

  If $\|r\| \leq \epsilon$, $\|s^1\| \leq \epsilon$, and $\|s^2\| \leq \epsilon$ stop; otherwise $k \leftarrow k + 1$
  If k equals k_{max}, stop; the algorithm does not converge.
end while

Algorithm 2 presents the steps of the three-block ADMM for solving two-stage SP in Equations (4.14)-(4.18). An important parameter of Algorithm 2 is the penalization parameter ρ. The convergence of the ADMM algorithm is highly sensitive to ρ; poor selection may lead to slow or non-convergence in practical problems [87]. A variant of ADMM, residual balancing, where the penalty parameter ρ_k changes at each iteration as proposed by [88]. The intuition behind the adaptive method is to make the primal and dual residuals norms have similar magnitudes. By doing so, the residuals will have small values at the stage of convergence. This approach makes the performance less dependent on the initial choice of ρ. Superlinear convergence with $\rho_k \to \infty$ has been achieved by [89]. We have implemented iteration dependent adaptive ρ_k version to three-block ADMM similar to the one suggested in [88].

The implemented approach guaranteed convergence to all benchmark problems in optimal CPU time. In Algorithm 2, the preconditioning step initializes penalty parameters ρ_1 and λ, the maximum number of iterations k_{max}, and the convergence tolerance ϵ. The initialization in Line 1 provides an initial first-stage solution for the primary iterations $k \geq 1$. We update the penalty parameter adaptively to maintain the gap difference between the primal and dual residuals; see Equation (4.23).

Algorithm 2 runs on deterministic initialization, in which we applied the first phase procedure of linear programming to bring the starting point to a feasible region. Since Algorithm 2 decomposes the problem by scenarios, Lines $5, 8, 12, 13, 15$, and 18 are implemented in parallel. Line 5 provides a scenario-wise solution to the second-stage variable in which all the components of the second-stage variable are solved concurrently. The second-stage solution needs to be assembled in one variable to find the first-stage solution, which is given in Line 9. Lines 10-13 give the updates for the dual variables using (4.22). The primal and dual residuals of Algorithm 2 are given in Lines 14-19. The adaptive penalty parameter ρ_{k+1} is given in Line 20.

Algorithm 2 terminates in two ways; the convergence case, where all the residuals are less than the threshold ϵ, Line 21, or the non-convergence case, Line 22, where the algorithm hits the maximum number of iterations without meeting the convergence requirements. In the non-convergence case, the algorithm enforced to stop when it hits the maximum iterations.

The adaptive selection to the penalty parameter is given by

$$\rho_{k+1} = \begin{cases} \nu\rho_k, & r_k > \mu \max\{s_k^1, s_k^2\} \\ \rho_k/\nu, & \min\{s_k^1, s_k^2\} > \mu r_k \\ \rho_k, & \text{otherwise}, \end{cases} \tag{4.23}$$

where $\nu > 1$, $\mu > 1$. Usually, we choose $\nu = 2$, and $\mu = 4$. The value $\epsilon = 10^{-3}$ and $k_{max} = 5\times10^4$ were used, respectively, in steps 20 and 21 of Algorithm 2.

Chapter 5

Convergence Analysis

5.1 Introduction

The Alternating Direction Method of Multipliers (ADMM) is a popular optimization algorithm for solving convex optimization problems with multiple constraints. The convergence of ADMM has been studied extensively in the literature, and several convergence guarantees have been established [73, 76, 77].

The ADMM algorithm updates the primal and dual variables of a problem alternatively, using a penalty parameter to enforce consistency between the two sets of variables. The algorithm has been shown to converge under various conditions, including strong convexity, smoothness, and linear constraints.

One of the most famous convergence results for ADMM is the one for convex optimization problems with linear constraints, which was proved by Boyd et al. [73]. The authors showed that, under mild assumptions, ADMM converges linearly to the optimal solution of the problem. Other convergence guarantees have been established for more general classes of problems, such as nonconvex problems and problems with non smooth objectives by Hong et al. [90]. Also, several papers have investigated the convergence of ADMM for nonconvex problems with various structures, such as separable nonconvex objectives or nonconvex constraints. Hong et al. [90] have shown that choosing a sufficiently large fixed penalty parameter makes ADMM converge. They have not imposed any assumptions on the iterates generated by their algorithm, and they have stated that the algorithm is applicable to ADMM variants.

Xu et al. [91] showed there is a solid relationship between Douglas-Rachford splitting [91] and ADMM where the two methods are equivalent in their dual form. They presented an adaptive penalty parameter selection for ADMM based on this connection, which achieves fast convergence irrespective of the initial penalty selection. Arpón et al. [81] have implemented ADMM to the two-stage stochastic program (SP). They have not provided the theoretical convergence of ADMM for two-stage SP but referred to the work of Sun et al. [78] for the convergence proof. We have taken a different approach to establish the theoretical convergence of ADMM. Because our SP problem class under consideration is nonconvex and nonsmooth, our convergence argument is based on semicontinuity and the Kurdyka-Lojasiewicz condition of the objective function[82].

The rest of this chapter is structured as follows. In Section 5.2, we will start by giving an introduction to the Kurdyka-Lojasiewicz property and present its definition. Then, we will introduce the Lyapunov function and its properties. Moving on to Section 5.3, we will delve into the mathematical proof of convergence. We will give the proof through a series of lemmas.

5.2 Preliminaries

In this chapter, we present the mathematical convergence of the three-block ADMM under some conditions. We started by outlining definitions and lemmas that are needed throughout the proof.

The Kurdyka-Lojasiewicz (K-L) property is a property that is related to the geometry of a function and its gradient. It states that a semi-algebraic function with a bounded gradient must also have a sublinear convergence rate, meaning that the distance between the function's values and its minimum value goes to zero at a rate slower than a linear function. It has been shown that under some conditions, the K-L property holds for the objective functions that are minimized by ADMM and can be used to theoretically justify the linear convergence rate of the ADMM algorithm when applied to certain types of problems [82].

Definition 5.2.1. *The function $f : \mathbb{R}^n \to \mathbb{R}$ has the Kurdyka-Lojasiewicz (K-L) [82] property at $\bar{x}$, if there exist positive constants $\eta > 0, \delta > 0$, and a concave function $\varphi : [0, \eta) \to \mathbb{R}^+$, such that $\forall x \in \mathcal{N}(\bar{x}, \delta) \cap \{x : f(\bar{x}) < f(x) < f(\bar{x}) + \eta\}$,*

$$\varphi^{'}(f(x) - f(\bar{x}))\mathtt{dist}(0, \partial f(x)) \geq 1,$$

where the function φ has the following characteristics: (i) φ is continuous on $[0, \eta)$; (ii) φ is smooth concave on $(0, \eta)$; (iii) $\varphi(0) = 0$, $\varphi^{'}(x) > 0$, $\forall x \in (0, \eta)$; $\mathtt{dist}(x, D) ::= \inf\{\|x - d\| : d \in D, x \in \mathbb{R}^n\}$.

Since all the functions we are considering in the two-stage SP are real analytic (linear, and indicator functions), they satisfy K-L inequality [82]. All the lemmas and theorems presented here apply to real analytic functions.

In our context, we define a function $V : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_3} \times \mathbb{R}^{m} \times \mathbb{R}^{n_3} \to \mathbb{R}$ by

$$V(x_1, x_2, x_3, \mu, \bar{x}_3) ::= L_\rho(x_1, x_2, x_3, \mu) + \frac{\tau}{2} \|x_3 - \bar{x}_3\|^2, \tag{5.1}$$

where $\tau = \frac{4\lambda^2}{\rho\sigma}$, λ and σ are constants, and $\bar{x}_3$ is the immediate prior iteration of x_3.

Let V be a lower semi-continuous function with non-empty domain and $V(x) > -\infty$ for every $x \in \text{dom } V$, and a_1 and a_2 are fixed positive constants. We suppose that the iterates of ADMM in (4.3) have a Lyapunov function that verifies the following subgradient decent conditions of [92]:

(C1) $V(x^{k+1}) \leq V(x^k) - a_1 \left\|x^k - x^{k+1}\right\|^2, \forall k \in \mathbb{N}$; this means that $V(\cdot)$ is a Lyapunov function that decreases in each iteration.

(C2) $\mathtt{dist}(0, \partial V(x^{k+1})) \leq a_2 \left\|x^k - x^{k+1}\right\|, \forall k \in \mathbb{N}$.

(C3) There exists a sub-sequence $\{x^{k_j}\}$ converge to the stationary optimal point x^* such that $V(x^{k_j}) \to V(x^*)$ as $j \to \infty$.

In the sequel, we assume the properties (A1)-(A4) hold, where the given matrices, functions, and parameters are subject to Equations (4.2)-(4.3):

(A1) V is a K-L function;

(A2) There is a constant $\sigma > 0$, such that $\sigma \|\mu\|^2 \leq \left\|A_3^T \mu\right\|^2, \forall \mu \in \mathbb{R}^m$;

(A3) The parameters are chosen such that $\rho > \frac{8\lambda}{\sigma}$;

(A4) All the matrices A_1, A_2, and A_3 are bounded under Frobenius norm.

We will demonstrate that the function V meets the conditions (C1)-(C3) once the sequence has been generated with the iterative procedure in Equation (4.3). The proofs of Lemmas 5.3.1 and 5.3.3 are based on our choice of V in (5.1).

5.3 Mathematical Proof of Convergence

The statements of Lemmas 5.3.1 and 5.3.4 are given in [92] and [82], respectively, but we have taken a different approach to apply them for two-stage SP.

Lemma 5.3.1. *Let $\{x^k\}$ be a sequence that meets the conditions (C1)-(C3). If V is a K-L function, then the sequence $\{x^k\}$ converges to $\bar{x}$. Furthermore, the sequence $\{x^k\}$ has a finite length, i.e., $\sum_{k=1}^{\infty} \left\|x^{k+1} - x^k\right\| < \infty$ [Theorem 3.1, [92]]. It follows that $V(\bar{x}) = V(x^*)$, where x^* is the optimal minimizer point. Moreover, the augmented function L_ρ and Lyapunov function V has the same optimal value.*

Proof: Condition (C1) states that the function V is a nonincreasing function, i.e., $V(x^{k+1}) \leq V(x^0)$. Also, condition (C1) and the fact $V(x^*) \leq V(x^{k+1})$ yield

$$\left\|x^{k+1} - x^k\right\| \leq \sqrt{\frac{V(x^k) - V(x^{k+1})}{a_1}} \leq \sqrt{\frac{V(x^k) - V(x^*)}{a_1}}, \quad \forall k \in \mathbb{N}. \tag{5.2}$$

If $V(x^*) = V(x^k)$, then x^k is the optimal point and the iteration will be terminated. Now, if $x^k \in \mathcal{N}(x^*, \delta)$ for $k \geq 1$, we claim that the following inequality holds

$$2\left\|x^{k+1} - x^k\right\| \leq \left\|x^k - x^{k-1}\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^k) - V(x^*)) - \varphi(V(x^{k+1}) - V(x^*))\right), \tag{5.3}$$

where the constants a_1 and a_2 are given in the conditions (C1)-(C2). If $x^{k+1} = x^k$, the inequality is trivial. Let us assume that $x^{k+1} \neq x^k$, $V(x^*) < V(x^k)$. The quantity $V(x^k) - V(x^*) > 0$ and (C2) for x^k and x^{k-1}, $x^k \neq x^{k-1}$, combined with K-L inequality, $\varphi'(V(x^k) -$

$V(x^*))\mathtt{dist}(0, \partial V(x^k) \geq 1$, $\mathtt{dist}(0, \partial V(x^k)) \neq 0$, imply

$$\varphi^{'}(V(x^k) - V(x^*)) \geq \frac{1}{\text{dist}(0, \partial V(x^k)} \geq \frac{1}{a_2 \left\| x^k - x^{k-1} \right\|}. \tag{5.4}$$

Since any concave differentiable function f satisfies the inequality

$$f(x) - f(y) \geq f^{'}(x)(x - y),$$

using this fact about the concave function combined with (C1) we have

$$\begin{aligned} \varphi(V(x^k) - V(x^*)) - \varphi(V(x^{k+1}) - V(x^*)) \geq& \varphi^{'}(V(x^k) - V(x^*)) \left(V(x^k) - V(x^{k+1})\right) \\ \geq& \varphi^{'}(V(x^k) - V(x^*)) \left(a_1 \left\| x^{k+1} - x^k \right\|^2\right). \end{aligned} \tag{5.5}$$

By combining the inequalities in (5.4) and (5.5) we have

$$\frac{a_2}{a_1} \left(\varphi(V(x^k) - V(x^*)) - \varphi(V(x^{k+1}) - V(x^*))\right) \geq \frac{\left\| x^{k+1} - x^k \right\|^2}{\left\| x^k - x^{k-1} \right\|},$$

and by multiplying the both sides with $\left\| x^k - x^{k-1} \right\|$ and taking the square root we have

$$\left\| x^{k+1} - x^k \right\| \leq \sqrt{\left\| x^k - x^{k-1} \right\| \frac{a_2}{a_1} \left(\varphi(V(x^k) - V(x^*)) - \varphi(V(x^{k+1}) - V(x^*))\right)}. \tag{5.6}$$

Then, by applying Young's inequality $2\sqrt{\alpha\beta} \leq \alpha + \beta$ to Equation (5.6) and multiplying the both sides by 2 we get

$$\begin{aligned} 2 \left\| x^{k+1} - x^k \right\| \leq& 2\sqrt{\left\| x^k - x^{k-1} \right\| \frac{a_2}{a_1} \left(\varphi(V(x^k) - V(x^*)) - \varphi(V(x^{k+1}) - V(x^*))\right)} \\ \leq& \left\| x^k - x^{k-1} \right\| + \frac{a_2}{a_1} \left(\varphi(V(x^k) - V(x^*)) - \varphi(V(x^{k+1}) - V(x^*))\right), \end{aligned}$$

from which inequality (5.3) follows.

Let $\epsilon, \delta > 0$ such that $\delta \in (0, \epsilon)$. We assume that

$$\left\|x^* - x^0\right\| + 2\sqrt{\frac{V(x^0) - V(x^*)}{a_1}} + \frac{a_2}{a_1}\varphi'(V(x^0) - V(x^*)) < \delta, \tag{5.7}$$

holds, where $x^0 \in \mathcal{N}(x^*, \delta)$ and we claim that

$$\forall k \in \mathbb{N}, x^k \in \mathcal{N}(x^*, \delta) \Longrightarrow x^{k+1} \in \mathcal{N}(x^*, \epsilon) \quad \text{where } V(x^*) \leq V(x^{k+1}). \tag{5.8}$$

The intuition behind the assumption (5.7) is that if our initial starting solution is close to the optimal solution of $V(x)$, the generated sequence will also remain bounded. The properties of the concave function, φ, also play a role in it.

We claim that $x^j \in \mathcal{N}(x^*, \epsilon)$ hold for $j = 1, 2, \ldots$. We prove this claim by induction. From (5.2), when $k = 0$, we have

$$\left\|x^1 - x^0\right\| \leq \sqrt{\frac{V(x^0) - V(x^*)}{a_1}}. \tag{5.9}$$

By combining inequality (5.9) with the assumption (5.7), and using triangle inequality we have

$$\left\|x^* - x^1\right\| \leq \left\|x^* - x^0\right\| + \left\|x^0 - x^1\right\| \leq \left\|x^* - x^0\right\| + \sqrt{\frac{V(x^0) - V(x^*)}{a_1}} < \delta,$$

which concludes that $x^1 \in \mathcal{N}(x^*, \delta)$. Suppose that the claim $x^j \in \mathcal{N}(x^*, \delta)$ holds for $j \geq 1$. We will now show $x^{j+1} \in \mathcal{N}(x^*, \delta)$. By taking the sum of (5.3) from $k = 1$ to j we get the following

$$\begin{aligned}
\sum_{k=1}^{j} 2\left\|x^{k+1} - x^k\right\| - \sum_{k=1}^{j}\left\|x^k - x^{k-1}\right\| &= \sum_{k=1}^{j}\left\|x^{k+1} - x^k\right\| + \sum_{k=1}^{j}\left(\left\|x^{k+1} - x^k\right\| - \left\|x^k - x^{k-1}\right\|\right) \\
&= \sum_{k=1}^{j}\left\|x^{k+1} - x^k\right\| + \left\|x^{j+1} - x^j\right\| - \left\|x^1 - x^0\right\| \\
&\leq \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right),
\end{aligned}$$

which can be written visibly as

$$\sum_{k=1}^{j} \left\|x^{k+1} - x^k\right\| + \left\|x^{j+1} - x^j\right\| \le \left\|x^1 - x^0\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right). \tag{5.10}$$

Then by adding and subtracting each $x^0, x^1, \cdots, x^j$ with the term $x^* - x^{j+1}$ and using triangle inequalities, and by (5.10) we have

$$\begin{aligned}
&\left\|x^* - x^{j+1}\right\| \\
&\le \left\|x^* - x^0\right\| + \left\|x^0 - x^1\right\| + \sum_{k=1}^{j} \left\|x^k - x^{k+1}\right\| && \text{(5.11a)}\\
&\le \left\|x^* - x^0\right\| + \left\|x^0 - x^1\right\| + \sum_{k=1}^{j} \left\|x^k - x^{k+1}\right\| + \left\|x^j - x^{j+1}\right\|, && \text{(5.11b)}\\
&\le \left\|x^* - x^0\right\| + 2\left\|x^0 - x^1\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right), \text{by (5.10)}, && \text{(5.11c)}\\
&\le \left\|x^* - x^0\right\| + 2\sqrt{\frac{V(x^0) - V(x^*)}{a_1}} + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right), \text{by (5.9)} && \text{(5.11d)}\\
&\le \left\|x^* - x^0\right\| + 2\sqrt{\frac{V(x^0) - V(x^*)}{a_1}} + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*))\right) && \text{(5.11e)}\\
&\le \left\|x^* - x^0\right\| + 2\sqrt{\frac{V(x^0) - V(x^*)}{a_1}} + \frac{a_2}{a_1}\left(\varphi(V(x^0) - V(x^*))\right) < \delta, && \text{(5.11f)}
\end{aligned}$$

where we have added the positive term, $\|x^j - x^{j+1}\|$, to the right hand side of inequality (5.11a). Since the term $\varphi(V(x^{j+1}) - V(x^*))$ in inequality (5.11d) is positive, so removing it will results in (5.11e). Inequality (5.11f) follows from $V(x^1) \le V(x^0)$, of Lyapunov function V. Thus, we conclude that $x^{j+1} \in \mathcal{N}(x^*, \delta)$. Therefore, inequality (5.3) holds for $k = j + 1$, namely

$$2\left\|x^{(j+1)+1} - x^{j+1}\right\| \le \left\|x^{j+1} - x^j\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^{j+1}) - V(x^*)) - \varphi(V(x^{(j+1)+1}) - V(x^*))\right). \tag{5.12}$$

Inequality (5.10) implies

$$\sum_{k=1}^{j} \left\|x^{k+1} - x^k\right\| \leq \left\|x^1 - x^0\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*))\right) -$$
$$\left[\frac{a_2}{a_1}\varphi(V(x^{j+1}) - V(x^*)) + \left\|x^{j+1} - x^j\right\|\right]$$
$$\leq \left\|x^1 - x^0\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*))\right),$$

since the expression in bracket is positive.

Therefore, it follows that

$$\sum_{k=1}^{\infty} \left\|x^{k+1} - x^k\right\| < \infty,$$

which infers that the sequence $\{x^k\}$ converge to some $\bar{x}$. Then, from condition (C2) and the proven claim $x^j \in \mathcal{N}(x^*, \delta), j = 1, 2, \cdots$, we conclude that $\mathtt{dist}(0, \partial V(\bar{x})) \to 0$ as $k \to \infty$ and

$$\liminf_{k\to\infty} V(x^k) = V(\bar{x}) \geq V(x^*), \quad \text{by lower semi-continuity.} \tag{5.13}$$

If $V(\bar{x}) > V(x^*)$, then using K-L inequality we have

$$\varphi^{'}(V(\bar{x}) - V(x^*))\mathtt{dist}(0, \partial V(\bar{x})) \geq 1,$$

which contradicts the fact that $\mathtt{dist}(0, \partial V(\bar{x})) \to 0$ as $k \to \infty$. Therefore

$$V(\bar{x}) \leq V(x^*). \tag{5.14}$$

Then, the inequalities (5.13) and (5.14) conclude that $V(\bar{x}) = V(x^*)$. Due to the convergence of x^k to $\bar{x}$, the regularized term $\|x_3 - \bar{x}_3\| \to 0$ from where $V(x^*) = L_\rho(x^*)$ follows.

Lemma 5.3.2. *For each $k \in \mathbb{N}$, there exists a positive constant $a_1 > 0$ such that $V(\hat{w}^{k+1}) \leq V(\hat{w}^k) - a_1 \left\|\hat{w}^{k+1} - \hat{w}^k\right\|^2$, where $\hat{w}^k = (x_1^k, x_2^k, x_3^k, \mu^k, x_3^{k-1})$.*

Proof:

Taking partial derivative with respect to x_3-subproblem in (4.3) at iteration $k+1$, yields

$$\begin{aligned}
&\nabla_{x_3} h(x_3^{k+1}) + A_3^T \mu^k + \rho A_3^T (A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t) + \\
&\lambda(x_3^{k+1} - x_3^k) = \nabla_{x_3} h(x_3^{k+1}) + A_3^T \mu^{k+1} + \lambda(x_3^{k+1} - x_3^k) = 0.
\end{aligned} \tag{5.15}$$

Using the expressions for $\nabla_{x_3} h(x^{k+1})$ and $\nabla_{x_3} h(x^k)$ from (5.15) and using the Cauchy-Schwartz and Young inequalities after re-arranging of terms we get

$$\begin{aligned}
\left\|A_3^T(\mu^{k+1} - \mu^k)\right\|^2 &= \left\|(\nabla h(x_3^{k+1}) - \nabla h(x_3^k)) + \lambda(x_3^{k+1} - x_3^k) - \lambda(x_3^k - x_3^{k-1})\right\|^2 \\
&\le \left\|(\nabla h(x_3^{k+1}) - \nabla h(x_3^k))\right\|^2 + \lambda^2 \left\|(x_3^{k+1} - x_3^k) - (x_3^k - x_3^{k-1})\right\|^2 \\
&+ 2\lambda \left\|\nabla h(x_3^{k+1}) - \nabla h(x_3^k)\right\| \left\|(x_3^{k+1} - x_3^k) - \lambda(x_3^k - x_3^{k-1})\right\| \\
&= \lambda^2 \left(\left\|(x_3^{k+1} - x_3^k) - (x_3^k - x_3^{k-1})\right\|^2\right) \\
&\le \lambda^2 \left(\left\|x_3^{k+1} - x_3^k\right\|^2 + \left\|x_3^k - x_3^{k-1}\right\|^2 + 2\left(\left\|x_3^{k+1} - x_3^k\right\| \left\|(x_3^k - x_3^{k-1}\|\right)\right) \\
&\le 2\lambda^2 \left(\left\|x_3^{k+1} - x_3^k\right\|^2 + \left\|x_3^k - x_3^{k-1}\right\|^2\right), \quad \text{by Young inequality}
\end{aligned}$$

where $\nabla h(x^{k+1}) - \nabla h(x^k) = 0$ because h is a linear function. Then, it follows from property $(A2)$ that

$$\left\|\mu^{k+1} - \mu^k\right\|^2 \le \frac{2\lambda^2}{\sigma} \left(\left\|x_3^{k+1} - x_3^k\right\|^2 + \left\|x_3^k - x_3^{k-1}\right\|^2\right). \tag{5.16}$$

Also, we have

$$L_\rho(x_1^{k+1}, x_2^k, x_3^k, \mu^k) + \frac{\lambda}{2}\left\|x_1^{k+1} - x_1^k\right\|^2 \le L_\rho(x_1^k, x_2^k, x_3^k, \mu^k), \tag{5.17a}$$

$$L_\rho(x_1^{k+1}, x_2^{k+1}, x_3^k, \mu^k) + \frac{\lambda}{2}\left\|x_2^{k+1} - x_2^k\right\|^2 \le L_\rho(x_1^{k+1}, x_2^k, x_3^k, \mu^k), \tag{5.17b}$$

$$L_\rho(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \mu^k) + \frac{\lambda}{2}\left\|x_3^{k+1} - x_3^k\right\|^2 \le L_\rho(x_1^{k+1}, x_2^{k+1}, x_3^k, \mu^k), \tag{5.17c}$$

$$L_\rho(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \mu^{k+1}) = L_\rho(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \mu^k) + \frac{1}{\rho}\left\|\mu^{k+1} - \mu^k\right\|^2 \tag{5.17d}$$

where the inequalities (5.17a)-(5.17c) follow from (4.3), and the equality (5.17d) follows from (4.2) and (4.3). By adding up (5.17a)-(5.17d), we obtain

$$L_\rho(w^{k+1}) \leq L_\rho(w^k) - \frac{\lambda}{2}\left\|u^{k+1} - u^k\right\|^2 + \frac{1}{\rho}\left\|\mu^{k+1} - \mu^k\right\|^2, \tag{5.18}$$

where $u^k = (x_1^k, x_2^k, x_3^k)$, and $w^k = (x_1^k, x_2^k, x_3^k, \mu^k)$.

Dividing both sides of inequality (5.16) by ρ and adding up with inequality (5.18) it follows that

$$L_\rho(w^{k+1}) \leq L_\rho(w^k) - \frac{\lambda}{2}\left\|u^{k+1} - u^k\right\|^2 + \frac{2\lambda^2}{\rho\sigma}\left(\left\|x_3^{k+1} - x_3^k\right\|^2 + \left\|x_3^k - x_3^{k-1}\right\|^2\right) \tag{5.19}$$

Therefore, by rearranging the terms of Equation (5.19) using $\tau = 4\frac{\lambda^2}{\rho\sigma}$, we get

$$\begin{aligned}
V(\hat{w}^{k+1}) &= L_\rho(w^{k+1}) + \frac{\tau}{2}\left\|x_3^{k+1} - x_3^k\right\|^2 \\
&\leq L_\rho(w^k) + \frac{\tau}{2}\left\|x_3^k - x_3^{k-1}\right\|^2 - \frac{\lambda}{2}\left\|u^{k+1} - u^k\right\|^2 + \tau\left\|x_3^{k+1} - x_3^k\right\|^2 \\
&= V(\hat{w}^k) - \frac{\lambda}{2}\left\|u^{k+1} - u^k\right\|^2 + \tau\left\|x_3^{k+1} - x_3^k\right\|^2 \\
&\leq V(\hat{w}^k) - \frac{\lambda}{2}\left\|u^{k+1} - u^k\right\|^2 + \tau\left\|u_3^{k+1} - u_3^k\right\|^2 \\
&\leq V(\hat{w}^k) - a_1\left\|\hat{w}^{k+1} - \hat{w}^k\right\|^2,
\end{aligned} \tag{5.20}$$

where $a_1 \in (0, \frac{\lambda}{2} - \tau]$, which is positive constant, sine $\frac{\lambda}{2} > \tau$, and this is due to Assumption (A3), which states that $\rho > \frac{8\lambda}{\sigma}$.

Lemma 5.3.3 (Lemma 3, [82])**.** *Let u^k and w^k respectively denote (x_1^k, x_2^k, x_3^k) and $(x_1^k, x_2^k, x_3^k, \mu^k)$. If the sequence $\{u^k\}$ is bounded, then $\sum_{k=1}^{\infty}\left\|\hat{w}^k - \hat{w}^{k+1}\right\|^2 < \infty$. In fact, w^k is asymptotically regular, indeed, $\left\|w^k - w^{k+1}\right\| \to 0$ as $k \to \infty$. Moreover, any accumulation point of $\{w^k\}$ is a stationary point of the augmented Lagrangian function L_ρ.*

Proof:

From Assumption (A2), and Equation (5.15) we have for $c = \left\|\nabla h(x_3^k)\right\|$

$$\sqrt{\sigma}\left\|\mu^k\right\| \leq \left\|A_3^T\mu^k\right\| \leq \left\|\nabla h(x_3^k)\right\| + \lambda\left\|x_3^k - x_3^{k-1}\right\| \leq c + \lambda\left\|x_3^k - x_3^{k-1}\right\|.$$

It follows that $\{\mu^k\}$ is bounded since $\{u^k\}$ is bounded. Hence $\{\hat{w}^k\}$ is bounded since $\hat{w}^k = (x_1^k, x_2^k, x_3^k, \mu^k, x_3^{k-1})$ is a combination of u^k and μ^k. Thus, for being the iterates of lower semi-continuous function V, there exist a subsequence $\{\hat{w}^{k_j}\}$ which converges to $\hat{w}^*$; it follows that $\liminf_{j\to\infty} V(\hat{w}^{k_j}) \geq V(\hat{w}^*)$. Lemma 5.3.1 showed that the function V decreases in each iteration, and thus $\{\hat{w}^{k_j}\}$ is convergent. Taking the summation for Equation (5.20) it follows that

$$a_1 \sum_{i=1}^{k} \left\|\hat{w}^{i+1} - \hat{w}^i\right\|^2 \leq V(\hat{w}^1) - V(\hat{w}^{k+1}) \leq V(\hat{w}^1) - V(\hat{w}^*),$$

which implies that $\sum_{k=1}^{\infty}\left\|\hat{w}^{k+1} - \hat{w}^k\right\|^2 < \infty$; then it follows that $\left\|\hat{w}^{k+1} - \hat{w}^k\right\|^2 \to 0$ as $k \to \infty$. This implies that primal and dual residuals (4.8)-(4.10) converge to zero as $k \to \infty$.

The following lemma states that $\mathtt{dist}(0, \partial V(\hat{w}^{k+1}))$ goes to zero.

Lemma 5.3.4. *There is a positive constant* a_2 *such that* $\mathtt{dist}(0, \partial V(\hat{w}^{k+1})) \leq a_2\left\|\hat{w}^{k+1} - \hat{w}^k\right\|$.

Proof: By taking the partial derivative of V in (5.1) with respect x_1 and equating it with zero and substituting $\mu^k = \mu^{k+1} - \rho\left(A_1x_1^{k+1} + A_2x_2^{k+1} + A_3x_3^{k+1} - t\right)$ we have

$$\begin{aligned} 0 &= \partial f(x_1^{k+1}) + A_1^T\mu^k + \rho A_1^T\left(A_1x_1^{k+1} + A_2x_2^k + A_3x_3^k - t\right) \\ &= \partial f(x_1^{k+1}) + A_1^T\mu^{k+1} + \rho A_1^TA_2(x_2^k - x_2^{k+1}) + \rho A_1^TA_3(x_3^k - x_3^{k+1}), \end{aligned} \tag{5.21}$$

by adding and subtracting the term $A_1^T\left(\mu^{k+1} - \mu^k\right)$ to the right hand-side of Equation (5.21) yields

$$\begin{aligned} 0 = &\underbrace{\partial f(x_1^{k+1}) + A_1^T\mu^{k+1} + A_1^T\left(\mu^{k+1} - \mu^k\right)}_{\nabla_{x_1}V(\hat{w}^{k+1})} + \rho A_1^TA_2(x_2^k - x_2^{k+1}) + \\ &\rho A_1^TA_3(x_3^k - x_3^{k+1}) - A_1^T\left(\mu^{k+1} - \mu^k\right), \end{aligned}$$

where in $\nabla_{x_1}V(x_1, x_2^{k+1}, x_3^{k+1}, \mu^{k+1}, x_3^k)\mid_{x_1 = x_1^{k+1}}$ we have used

$$\mu^k = \mu^{k+1} - \rho\left(A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t\right)$$

.

Replicating this step for the rest of variables respectively yields

$$\begin{aligned}
&0 = \partial g(x_2^{k+1}) + A_2^T \mu^{k+1} + A_2^T\left(\mu^{k+1} - \mu^k\right) + \rho A_2^T A_3 (x_3^k - x_3^{k+1}) - A_2^T\left(\mu^{k+1} - \mu^k\right), \\
&0 = \partial h(x_3^{k+1}) + A_3^T\left(\mu^{k+1} - \mu^k\right) + A_3^T \mu^{k+1} - A_3^T\left(\mu^{k+1} - \mu^k\right), \\
&0 = \nabla_\mu V(\hat{w}^{k+1}) = \frac{1}{\rho}(\mu^{k+1} - \mu^k) \quad \text{and} \\
&0 = \nabla_{\bar{x}_3} V(\hat{w}^{k+1}) = \lambda(x_3^k - x_3^{k+1}).
\end{aligned} \tag{5.22}$$

Adding up all the previous equations in (5.21)-(5.22) associated with each variable and applying the Frobenius norm to the both sides results in

$$\texttt{dist}(0, \partial V(\hat{w}^{k+1}) \le b_1 \left\|x_2^k - x_2^{k+1}\right\| + b_2 \left\|x_3^k - x_3^{k+1}\right\| + b_3 \left\|\mu^{k+1} - \mu^k\right\|,$$

where $b_1 = \left\|\rho A_1^T A_2\right\|_F, b_2 = \rho\left(\left\|A_1^T A_3\right\|_F + \left\|A_2^T A_3\right\|_F + \frac{\lambda}{\rho}\right)$ and , $b_3 = \|A_1\|_F + \|A_2\|_F + \|A_3\|_F + \frac{1}{\rho}$. Since the matrices A_1, A_2, A_3 are all bounded in Frobenius norm, implies that $\exists a_2 > 0$ such that $a_2 = \max\{b_1, b_2, b_3\}$. Then, adding the appropriate norm for x_1^{k+1} and x_3^{k-1} the inequality holds.

Lemma 5.3.5. *Suppose the sequence $\{w^{k_j}\}$ generated by the iterative procedure* (4.3) *converge to $\hat{w}^*$and V is lower semicontinuous, i.e.* $\liminf_{j\to\infty} V(\hat{w}^{k_j}) \ge V(\hat{w}^*)$, *then* $\lim_{j\to\infty} V(\hat{w}^{k_j}) = V(\hat{w}^*)$.

Proof: The proof of convergence follows directly from Lemma 5.3.1, in which the function V satisfies the conditions (C1)-(C3). We have shown in Lemmas 5.3.2-5.3.4 that V meets all conditions (C1)-(C3).

Chapter 6

Optimization of Hybrid Renewable Energy

6.1 Introduction

Over the last two decades, there has been an increased tendency toward determining the optimal sizing of hybrid renewable energy systems. An optimal sizing approach for a hybrid power system is designed by Dong et al. [41]. They proposed an entirely renewable system that uses PV, wind power, battery, and hydrogen subsystem as backup suppliers. Their goal is to design an optimal system with lower annual costs and higher reliability. They defined the system reliability as a function of Loss of Power Supply Probability (LPSP) [93] and concluded that a reciprocal relationship exists between the system reliability and cost. They also stated that a system with several backup suppliers is more reliable and efficient than a sole source; relying on one backup source increases the possibility of getting damaged. However, they used Ant Colony Optimization (ACO), where the number of ants was selected using a trial and error approach. Most importantly, the solution obtained cannot be guaranteed optimal in the mathematical sense. They agreed that the ACO algorithm did not ensure complete solution accuracy.

Another study by Zhou et al. [94] suggested a two-stage stochastic programming approach for determining the design of a distributed energy system in a hotel. They employed the Genetic Algorithm (GA) to find the first-stage variables and the Monte Carlo approach to address the uncertainty of second-stage variables. A similar study by Sakalauska and Žilinskas [5] suggested a method for solving the two-stage stochastic programming problem using a sequence of Monte Carlo sampling estimators. However, the Monte Carlo technique requires large sample sizes to generate an accurate solution, needing significant CPU times.

There are a variety of techniques for reliability assessment in the literature. The concept of reliability relies on the probability of the loss of the power supply by the renewable system when the load demand is unfulfilled. One of the methods used often in reliability assessment is called loss of load probability (LLP) [11], which is the failure time of the hybrid system over a horizon of time. LLP happens when demand surpasses available producing capacity. The LPSP [11] is the most often used metric of reliability evaluation in the literature. LPSP is the average loss of the renewable system's power supply divided by the total load demand of the system throughout the specified time. A system with zero LPSP ensures that the load is always fulfilled. On the other hand, a score of one indicates that the system never satisfied the load demand.

There are several methods for calculating the cost of a hybrid energy system. The system cost considers the capital, operating, and maintenance expenses. One of the most common techniques is the annualized cost of the system, which adds up the yearly costs of each part of the hybrid system. The second technique examines the overall cost over the stipulated length of the system and is typically used to set up project feasibility.

The hybrid renewable system solely depends on natural resources that seek to reduce the dependency on fuel-based systems and thus downsizes the environmental impact. Designing an optimal hybrid renewable energy system with a minimum cost and high reliability is required in real energy applications. A system with high reliability is costlier even though it will satisfy the load demand required by the operator. A system with lower reliability, and the low-cost system, may result in a severe power shortage. The proposed optimal system should consider the trade-off between those two crucial factors. We have achieved this by formulating the problem mathematically and solving it deterministically and thereby ensuring optimality and accuracy.

We plan to formulate an optimization problem for a hybrid renewable system by constructing an objective function coupled with the constraints throughout the rest of this section. The set of conditions describes the practical restrictions of the real problem.

6.1.1 Reliability of the System

The uncertainty of the weather conditions results in fluctuating energy production, affecting the system's reliability. The LPSP is used as an index to estimate reliability. In the constraints, we satisfy a certain level of reliability to meet the system demand by a specified percentage of the probability of LPSP. The system reliability for the time interval T, as given in [95], is described in terms of LPSP and expressed as

$$LPSP = \frac{\sum_{t=1}^{T} LPS(t)}{\sum_{t=1}^{T} P_l(t)} \leq \alpha, \quad 0 \leq \alpha \leq 1, \tag{6.1}$$

where $LPS(t)$ is the loss of power supply at time t and given by

$$LPS(t) = (P_l(t) - P_g(t)),$$

$P_l(t)$ and $P_g(t)$ are the required load and the generated power, respectively, at time t. The system will be more reliable when it has a small $LSPS$, $\alpha \to 0$, and less reliable when it is large, $\alpha \to 1$. We can write Equation (6.1) as

$$\sum_{t=1}^{T} P_g(t) \geq (1-\alpha) \sum_{t=1}^{T} P_l(t).$$

6.1.2 The Mathematical Model

We assume that the projected demand for electricity follows a known probability distribution. We consider m blocks of demand for various quantities of power throughout the year. The formulation of the objective function for n different renewable energy plants for m demand blocks of generation in t years is given by

$$\min \sum_{k=1}^{n} c_k x_k + \mathbf{E}_\omega \left[\sum_{k=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{t} q_{k\omega} \tau_j y_{kjl\omega} \right], \quad \text{subject to} \tag{6.2}$$

$$\sum_{k=1}^{n} y_{kjl\omega} \geq (1-\alpha) d_{jl\omega}, \qquad j = 1, \cdots, m, \quad l = 1, \cdots, t, \forall \omega \tag{6.3}$$

$$y_{kjl\omega} \leq x_k, \qquad k = 1, \cdots, n, \forall j, l, \omega \tag{6.4}$$

$$\sum_{k=1}^{n} c_k x_k \leq b \tag{6.5}$$

$$x_k, y_{kjl\omega} \geq 0, \forall k, j, l, \omega \tag{6.6}$$

where c_k is the capital cost per kilowatt (kW) of the power plant k, x_k the capacity in kW to build for the power plant k, b is the total allocated budget, $y_{kjl\omega}$ the amount of electricity capacity fully used to produce electricity by power plant k for demand block j in year l under scenario ω measured in kW, $d_{jl\omega}$ load demand in year l at block j under scenario ω in kW, α is the factor of LPSP, $q_{k\omega}$ the operating cost of unit k under scenario ω in $/kWh, and τ_j is the duration of the demand at block j in hours.

Equation (6.2) presents the objective function that determines the cost for each plant in kW capacity together with the operational cost. The constraint in Equation (6.3) represents the reliability condition as the power generated by the hybrid system should meet a minimum level of reliability given by α. Equation (6.4) ensures that the electricity generated by unit k does not exceed the total capacity of that unit. Equation (6.5) guarantees that the capital cost does not surpass the available budget.

We have used five modes to measure the percentage of the power generated by plants and the amount of power shortage by the system.

The rest of this chapter is organized as follows. Section 6.2 presents a case study that applies the mathematical model to a rural area. It covers the problem formulation and a numerical experiment of the model. Then, Section 6.3 compares the numerical results obtained from our model with those obtained using the three-block ADMM with Monte-Carlo Method.

6.2 Case Study: Ga-Nkoane

In this study, we analyze building a hybrid renewable energy system in the village of Ga-Nkoana to meet the current and future load demands of electrical power. Ga-Nkoana is a suburb located in Limpopo province, South Africa, at a latitude of $-24.416673°$ and a longitude of $29.783335°$. According to the 2011 SA census, there are 443 households at Ga-Nkoana. Based on the assumption that the household growth rate is 3% per year, the logistic growth equation with a carrying capacity of $K = 10,000$ [96] can be used to estimate the number of households; there will be approximately 623 households by the year 2023.

The renewable system is to be constructed once and used for the whole duration of the project, 20 years. We chose four renewable resources for this study, solar power, wind power, hydrogen power, and backup batteries. We assume an initial budget of 120 thousand dollars for this construction. The purpose of this case study is to find the ideal mix of these renewable plants for the Ga-Nkoane rural area that minimizes the total capital cost and the expected operating cost over 20 years while maintaining a certain level of reliability using LPSP. The operating cost is assumed to be stochastic due to the uncertainty in the weather condition and the fluctuating load demand.

The hourly load day for Ga-Nkoana is obtained from the Domestic Electrical Load Metering Data (DELM) between 1994 and 2014 available in DataFirst's secure center [97]. The climatic data for the village is obtained from the Prediction of Worldwide Energy Resources (POWER) [98] project from 2013 to 2020, run by NASA. The data consist of temperature at 2 meters, wind speeds at 10 and 60-meter height, and solar irradiance. We have used the climatic data to build machine learning models to predict the solar and wind power output in Chapter 3. We have summed up the hourly load data for 632 residential households in Ga-Nkoana to express the monthly load profile for the whole village.

Table 6.1 shows the average monthly load profile, solar irradiance, wind speed, and temperature. It demonstrates that the average load per year is 6.837 kW, which is minimal compared to any city. We have noticed that the average load for all the months was almost the same, except for January, when it is slightly higher. We considered the yearly growth rate of the demand load to be the same as the population growth of 3%, so we added this percentage every year to the demand load to accommodate any uncertainty that might happen by increasing the population. Also, it presents the average temperature, which is a crucial factor that affects the power generated by the PV cells. Also, we notice the highest temperature degrees and solar irradiance is during the summer season in December and January, while the lowest values are in June and July, which is the winter season. During the months with peak temperature and solar irradiance, we expect the solar power output to be the most productive. On the other hand, the highest wind speed is in October, and the lowest is in May. The wind turbine is most generative during the months with high wind speed.

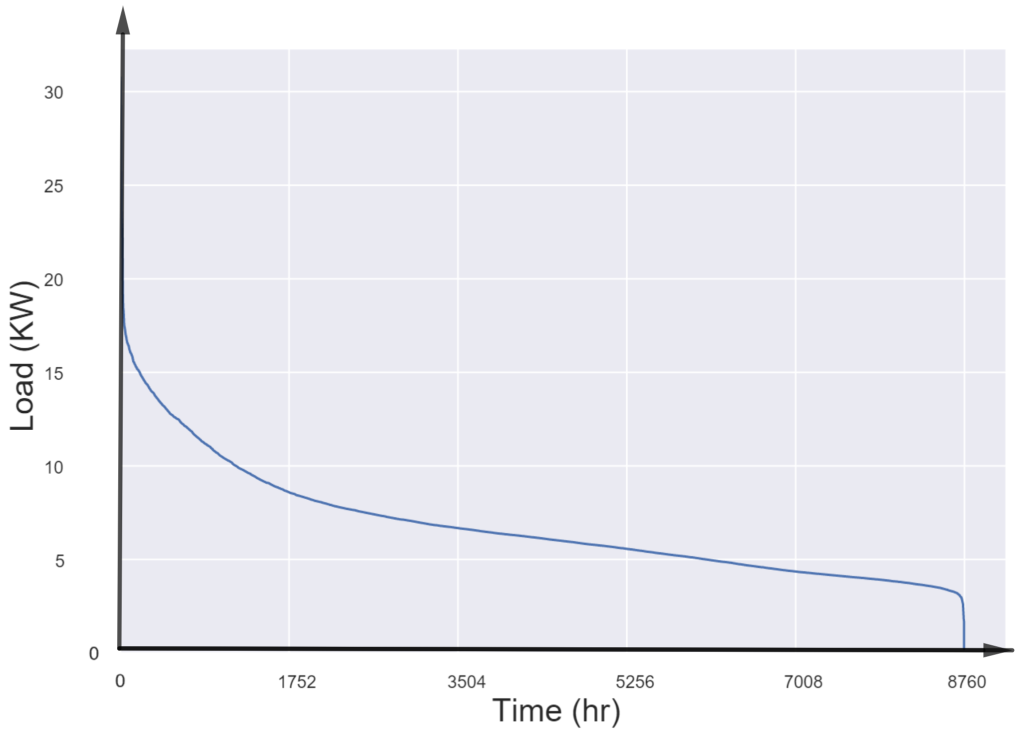


Figure 6.1: Load Duration Curve of the Demand

Table 6.1: Monthly Average Load and Climatic Data for Ga-Nkoane

	Load (kW)	Solar Irradiance (W/m^2)	Wind Speed (m/s)	Temperature (C°)
January	8.60	377.36	3.79	23.03
February	6.42	376.77	3.57	22.67
March	6.55	365.21	3.18	21.55
April	6.91	343.62	3.11	19.20
May	6.45	303.55	2.95	16.51
June	6.68	282.99	3.39	13.67
July	7.01	284.19	3.60	13.61
August	6.87	300.19	3.91	16.66
September	6.80	327.52	4.39	19.99
October	6.67	351.60	4.70	21.40
November	6.15	367.99	4.34	22.67
December	6.94	380.23	3.81	23.50

The prices of power plants are determined according to their electric capacity, measured in kilowatts (kW). Table 6.2 shows the capital cost per kW of capacity for each power plant.

Demands for electric power are described using the load duration curve, which illustrates the relationship between generation capacity requirements and capacity utilization [99]. The load duration curve is ordered in descending order by magnitude. The electric load for Ga-Nkoana using the continuous load duration curve is shown in Figure 6.1, which displays the load duration curve for 2013 for a total of 8760 hours. For example, in Figure 6.1, the load was approximately above 5 kW for 5256 h in 2013 and above 9 kW for 1752 h.

We obtained the quantized demand set in Table 6.3 from the continuous curve, presented in Figure 6.1, using the Mid-riser quantizer [100]. The load data range between 0.27 and 26.05 kW, and we want to quantize it using a Mid-riser quantizer with five demand blocks. The Mid-riser quantizer divides the input range into equally sized intervals, with the thresholds between the intervals will be at the midpoints of each interval. For our case, the intervals and thresholds are:

Interval 1: 20.22 to 26.05 kW

Threshold 1: 23.14 kW

Interval 2: 15.86 to 20.22 kW

Threshold 2: 18.04 kW

Interval 3: 10.44 to 15.86 kW

Threshold 3: 13.15 kW

Interval 4: 5.25 to 10.44 kW

Threshold 4: 7.85 kW

Interval 5: 0.27 to 5.26 kW

Threshold 4: 2.76 kW.

Then, we measured the duration of the load demand in each interval to find the corresponding number of hours for each block, which is given in the third column of Table 6.3.

6.2.1 Problem Formulation

We have created a hybrid renewable energy system that utilizes a two-stage SP optimization technique to reduce the total capital and operational costs over a certain period. The two-stage model has two types of variables: the first, known as the first-stage decision, is computed before any information about the potential realization of the random variable in the future is revealed. The first-stage variables define the amount of capacity in kW that needs to be built for each plant, and it is defined by the vector $x \in \mathbb{R}^4$. This order will be used in this section: 1. Solar; 2. Wind; 3. Hydrogen; 4. Batteries; and the subscript notation x_1, x_2, x_3, x_4 points to each type, respectively. The capital cost is represented by the vector $c \in \mathbb{R}^4$ in Table 6.2.

The second-stage variable y corresponds to the decision after the realization of the random event, which represents the capacity needed to produce electricity at each power plant for each demand block in each year under each scenario. The expected power demands over scenarios with the duration of five blocks during the first year are shown in Table 6.3.

Table 6.2: Capital Cost per kW Capacity [1], [2], [3], [4]

Plant	Cost per kW in \$ (c_k)
Solar	3000
Wind turbine	1300
Fuel cell & Electrolyzer	1700
Batteries	1500

Table 6.3: Expected Power Demand Data for Ga-Nkoane

Demand Block	Demand (kW) (d_{jl})	Duration (hours) (τ_j)
1	23.14	29
2	18.04	121
3	13.15	935
4	7.85	3134
5	2.76	4541

The second-stage decisions depend on stochastic problem data. Each possible value of the stochastic data is called a scenario indexed by ω. For this problem, there are five possibilities of operating costs for PV (see Table 6.4) and five possible operating costs for wind turbines (see Table 6.6), for a total of $\omega = 5 \times 5 = 25$ scenarios. The reason is that the probabilities of the operating costs are independent.

The operating costs, $q_{k\omega}$, $k = 1, \cdots, 4$, and $\omega = 1, \cdots, 25$, for different electricity sources are given in $\$/kWh$; see Tables 6.5, 6.4, and 6.6. The operating costs for the Fuel Cell & electrolyzer and batteries will remain the same for all the scenarios. We have used the growth rate of 3% to find the operating costs for the rest of the years. The duration in hours of each demand block, τ_j, $j = 1, \cdots, 5$, are given in the third column in Table 6.3.

We can write the full optimization formulation as:

Table 6.4: Probability Distribution of PV Operating Costs

Scenario	Cost $/kWh	Probability
1	0.0583	10%
2	0.0592	20%
3	0.0603	40%
4	0.0614	20%
5	0.0615	10%

Table 6.5: Operation Cost of Power Generation. * Solar Plant and Wind Turbine Operating Costs are Expected Values

Plant	Cost $/kWh ($q_k$)
Solar *	0.060
Wind turbine *	0.015
Fuel cell & Electrolyzer	0.025
Backup batteries	0.150

$$
\begin{aligned}
&\min_{x,y} \sum_{k=1}^{4} c_k x_k + \sum_{\omega=1}^{25} p_\omega \sum_{k=1}^{4} \sum_{j=1}^{5} \sum_{l=1}^{20} q_{k\omega} \tau_j y_{kjl\omega}, && \text{subject to} \\
&\textstyle\sum_{k=1}^{4} c_k x_k \leq b, && \text{budget constraint,} \\
&y_{kjl\omega} \leq x_k, \quad k = 1, \cdots, 4, \forall j, l, \omega && \text{capacity constraint,} \\
&\textstyle\sum_{k=1}^{4} y_{kjl\omega} \geq (1-\alpha) d_{jl\omega}, \qquad \forall j, l, \omega, && \text{demand constraint,}
\end{aligned}
\tag{6.7}
$$

where $d_{jl\omega}$ the demand load in block j and year l under scenario ω. To simplify Problem (6.7), we can convert it into the format of Problem (4.13) by introducing appropriate slack variables. This conversion allows us to express the budget constraints as $Ax = b$ and the capacity and demand constraints as $Tx + Wy = h$. In other words, by incorporating slack variables in Problem (6.7), we can more easily transform the inequality into equality, as expressed in Problem (4.13).

The size of these problems is enormous, and the standard linear programming (LP) solvers might face difficulties in solving them. There are four first-stage variables and $4 \times 5 \times 20 \times 25$ second-stage variables in this problem, for a total of $10,004$ variables. There is also one budget constraint, $4 \times 5 \times 20 \times 25$ capacity constraints, and $5 \times 20 \times 25$ demand constraints, totaling $12,501$ constraints. Such a problem size presents some difficulties to many modern LP solvers.

Table 6.6: Probability Distribution of Wind Turbine Operating Costs

Scenario	Cost $\$/kWh$	Probability
1	0.0141	10%
2	0.0148	20%
3	0.0149	40%
4	0.0153	20%
5	0.0154	10%

Increasing the number of scenarios will lead to an explosion in the problem size, making the use of the LP solver unrealistic.

Modes

Based on the available electricity generated by the hybrid system, the suggested system shifts between five modes at each block. In the case study of the rural area considered in this chapter, it is expected that wind turbines and PV solar panels will generate the bulk of power. As a result, this will influence our choice of power generation mode. We employ these modes as a virtual controller to monitor the percentage of the power supplied by wind turbines, PV-wind, PV-wind-batteries, and PV-wind-batteries-fuel cells with the load demand for the entire duration of the project. We used the final optimal solution to find the contribution of each power plant as a percentage of the power supply.

Firstly, we examine the power generated by the wind turbine and compare it with the load; if it can meet the load demand, it joins the first mode. If the wind power is deficient in meeting the load demand, then the combined power of wind and PV is checked; if the combined power meets the load, the system switches to the second mode. If the combination of wind and PV power is insufficient to meet the load demand, then we check the battery's state of charge (SoC) to see if it is at a minimum level. If the battery has enough power, then the battery power is added to the blended power, and the system switches to mode 3.

If the load is not satisfied by adding the battery power, or if the battery's SoC falls below the minimum level, we check the available hydrogen in the tank. If there is enough hydrogen to operate the FC, we add the FC power to the existing one, and the system switches to mode 4. If there is insufficient hydrogen or the SoC of the battery is at the minimum level, we encounter a power outage, and the system enters mode 5. The system moves to mode 5 even if all the available resources generate power, but they can still not satisfy the demand. The transition process is shown in Figure 6.2.

6.2.2 Numerical Experiments

This section discusses the numerical experiment of Algorithm 2 implemented to solve the two-stage SP of the hybrid renewable energy system presented above. The computer specification that runs the experiments has CPU Intel ® Core ™i7-7700 CPU @ 3.60GHz ×8 and 15.5 GB of RAM using MATLAB 2022b. The CPU times for all problems are given. The graphs of the residuals are provided in the log/log scale.

We run Algorithm 2 (the algorithm for three-block ADMM) using the load data for Ga-Nkoane with LPSP = 0.01 (defined as α in Equation (6.7)). We found that the optimal configuration of the hybrid renewable energy system is: $x_1 = 20.3829$kW using PVs, $x_2 = 9.1330$ kW using wind turbines, $x_3 = 12.1059$kW, and $x_4 = 10.4619$ kW of batteries. This configuration of the hybrid renewable system generates a maximum of 41.6218 kW ($x_1 + x_2 + x_3$) at optimal conditions. If we assume that the battery (x_4) can be charged or discharged at a constant rate of 10.4619 kW for 1 hour, then the battery has an energy capacity of 10.4619 kW hour $= 10.4619$ kWh. The cost of these configurations is $\$109,950$ as presented in Table 6.7.

Figure 6.3 shows all the residuals of Algorithm 2, which are given by Equations (4.8)-(4.10), are converged before or by 272 iterations on the hybrid renewable energy problem with a CPU time of 0.1985 seconds. Figure 6.4 shows the convergence behavior of the objective function to the optimal value.

Figure 6.5 shows the relationship between LPSP and the system cost (construction and operating costs). As depicted in the figure, a system with small LPSP values and higher system reliability is costlier because it requires building more units of renewable sources to avoid any power outage.

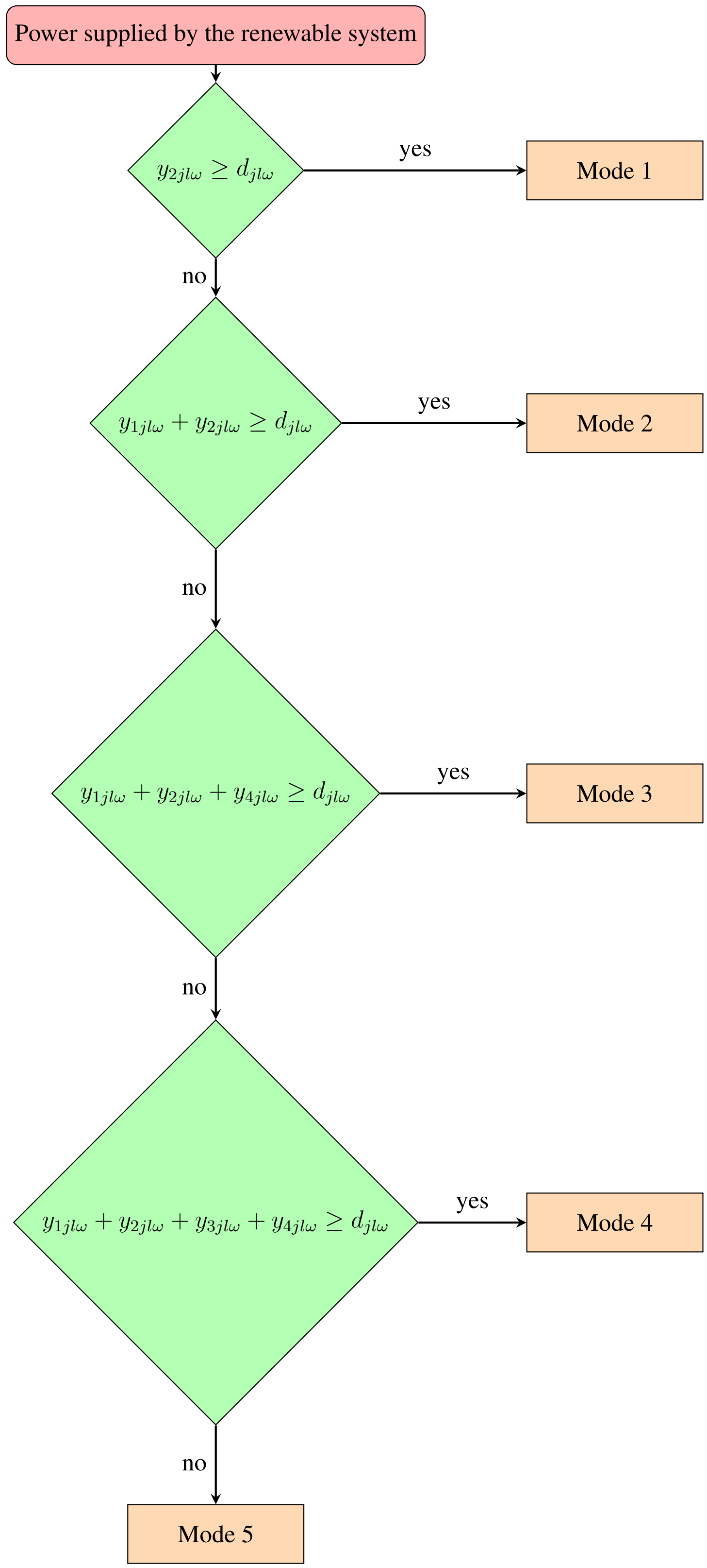


Figure 6.2: Flowchart of Transitioning between Different Modes

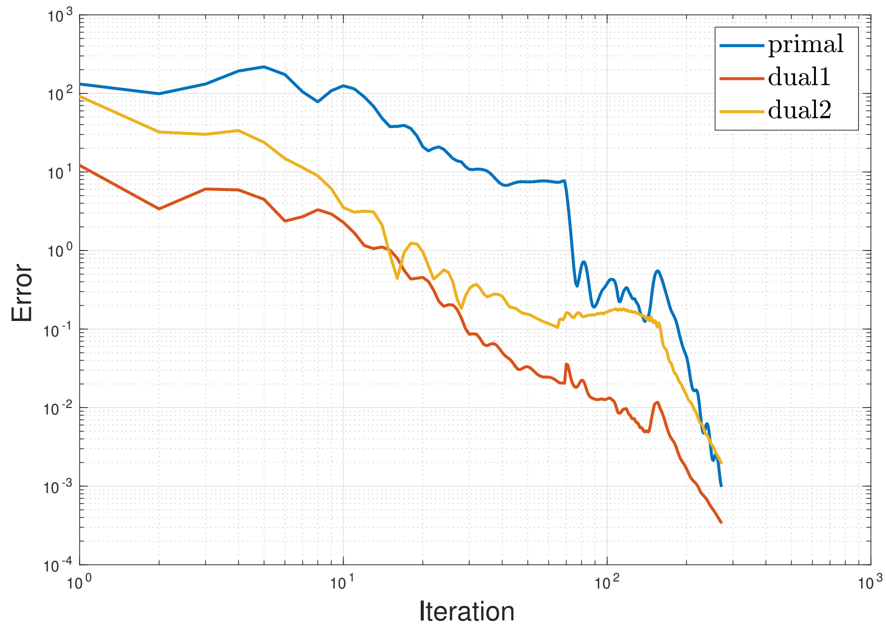


Figure 6.3: Convergence Behavior of the Hybrid Optimal Power System

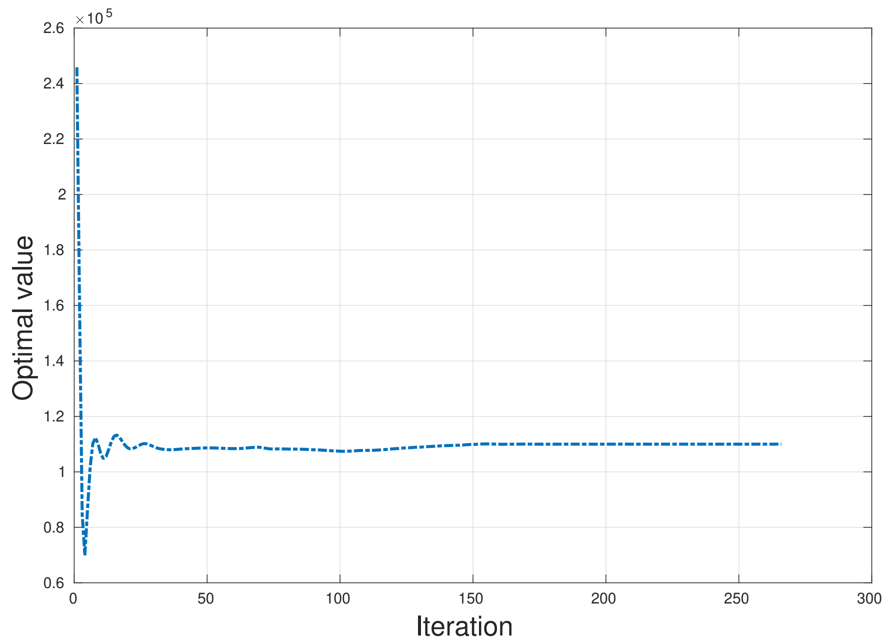


Figure 6.4: Convergence Behavior of the objective function

Table 6.7: Optimal Power Plant that Supply the Load of Ga-Nkoane

Plant	Optimal decision
Solar PV	20.3829 kW
Wind Turbine	9.1330 kW
Fuel cell	12.1059 kW
Batteries	10.4619 kWh
Total Cost $	109,950

Decreasing the tightness of reliability results in a high reduction in the cost of the system, which is good from an economic perspective, but it might have some power outages from time to time.

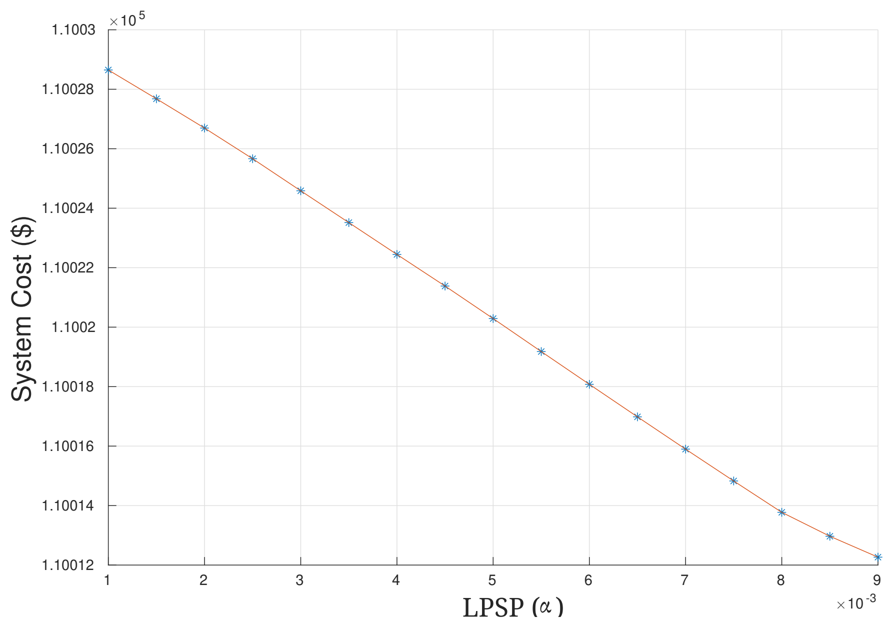


Figure 6.5: System Cost vs. LPSP

Figure 6.6 shows the barplot of the five modes of the hybrid system using LPSP = 0.01. It depicts the changes in the system between the modes for the whole duration. From the figure, it is clear that wind power alone does not satisfy the load demand as it appears in mode 1. Wind power independently contributed 13.6% to the system. Hybridizing the wind and solar power, as in mode 2, makes the system more robust and meets the demand 62.5% of the time. The batteries are used with wind and solar power only 10.7% of the time to supply the shortage that might happen due to peak load demand or weather uncertainties. The utilization of hydrogen power in mode 4 contributed to the system by a noteworthy 12.9%. Additionally, we observed that the system experienced a power outage of merely 0.4% in mode 5, as we had configured the LPSP of the reliability to be 1% to ensure that the system maintains an impressive 99% reliability.

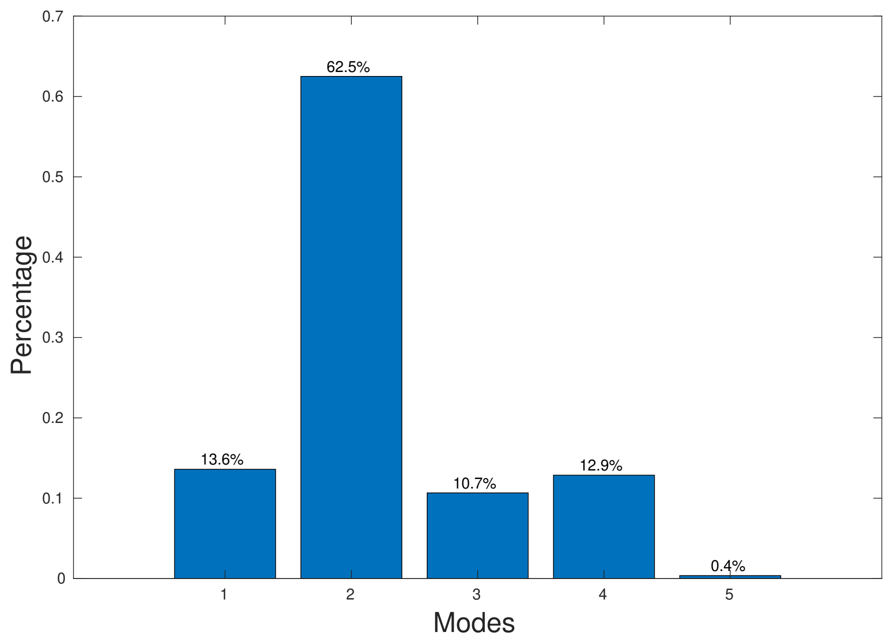


Figure 6.6: Modes of the System with LPSP = 0.01

The results of this case study may be applied to the installation of a hybrid renewable energy system in a location with comparable load data.

6.3 Comparing ADMM with Monte Carlo Method

Sakalauskas and Zilinskas [5] delved into power plant investment by employing two-stage stochastic programming with Monte Carlo simulation. Their proposed project was for 15 years, with a cost estimate of $10 billion to construct four power plants: gas, coal, nuclear, and hydropower. The researchers utilized five independent and normally distributed demand blocks to represent varying power consumption levels throughout the year. They reported the average and

Table 6.9: Capital Cost per GW Capacity [5]

Plant	Cost per GW in billion $\$$
Gas Turbine	1.1
Coal	1.8
Nuclear power	4.5
Hydroelectric	9.5

standard deviation of the expected power demands in gigawatts (GWs) depending on the duration of the blocks in Table 6.8.

Table 6.8: Expected Power Demand During the Year [5]

Demand Block	Demand (GW) $\hat{\mu}_j$	Standard deviation of power demands (GW) $\hat{\sigma}_j$	Duration (τ_j)
1	26.0	1.3	490
2	21.5	1.1	730
3	17.3	0.9	2190
4	13.9	0.7	3260
5	11.1	0.6	2090

The price of each power plant is determined based on its electricity production capacity in gigawatts. They displayed the investment costs for each plant, see Table 6.9. [5] stated that the hydropower capacity does not exceed 5.0 GW due to geographical constraints in the area.

Table 6.10 shows the operating costs and the cost of purchasing electricity from a third party for each kind of power plant in dollars per kilowatt-hour (kWh).

Table 6.10: Operating Cost of Power Generation

Power plant type	Cost \$/kWh
Gas Turbine	0.0392
Coal	0.0244
Nuclear	0.0140
Hydroelectric	0.0040
External Source	0.1500

They proposed the two-stage SP formulation for the problem as follows:

$$
\begin{aligned}
&\min_{x \geq 0} \sum_{k=1}^{4} c_k x_k + \mathbf{E}_\omega \left(\sum_{k=1}^{4} \sum_{j=1}^{5} \sum_{l=1}^{15} q_{k\omega} \tau_j y_{kjl\omega} \right), \quad \text{subject to} \\
&\sum_{k=1}^{4} c_k x_k \leq 10 \;\text{ billion}, \\
&x_4 \leq 5.0 \\
&y_{kjl\omega} \leq x_k, \quad k = 1, \cdots, 4, \forall j, l, \omega \\
&\sum_{k=1}^{5} y_{kjl\omega} \geq d_{jl\omega}, \qquad \forall j, k, \omega \\
&x \geq 0, y \geq 0.
\end{aligned}
$$

Sakalauskas and Zilinskas [5] applied the Monte Carlo method to expected and stochastic data. They found that the cost of the system based on expected data is $\$17.137 \pm 0.053$ billion and the one based on stochastic data is $\$16.508 \pm 0.028$ billion. Through this experiment, they argued the importance of using uncertain data in the investment planning of power systems, which costs less than the deterministic approach.

Table 6.11: Optimal Power Plant Sizing Based on the Stochastic Data

Power plant	Size in GW using Monte-Carlo	Size in (GW) using ADMM
Gas Turbine	4.4500	4.4497
Coal	4.3600	4.3593
Nuclear	4.6000	4.6003
Hydroelectric	5.0000	4.9995
Cost (billion \$)	16.508 ± 0.028	16.5037 ± 0.023

We solved the same problem using the same data provided by Sakalauskas and Zilinskas with three-block ADMM. ADMM approach terminated after 94 iterations in 0.0460 seconds of the CPU time, while the Monte Carlo approach ended after 123 iterations. Although the Monte Carlo method terminated in a moderate number of iterations, the size of the samples used by the objective function was very large. At each iteration, the sampling objective estimator requires a 15887 sample size. The authors did not provide the CPU time for this problem, but due to the large sample size used in the Monte Carlo method, the CPU time will be much larger than ours.

Figures 6.7 and 6.8 show the behavior of the optimal objective function and the convergence of ADMM, respectively. Table 6.11 compares the solutions of Monte Carlo and ADMM using stochastic data. The cost of the system using ADMM is $\$16.5037 \pm 0.023$ billion, which is almost identical to their solution. Also, the decisions of the power plants were the same.

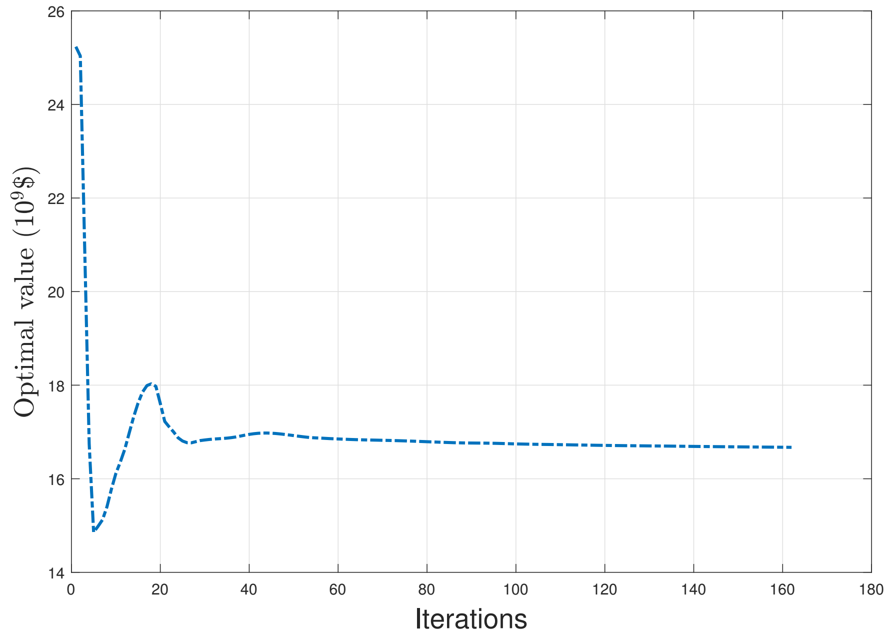


Figure 6.7: Optimal Objective Function using ADMM on the Stochastic Data

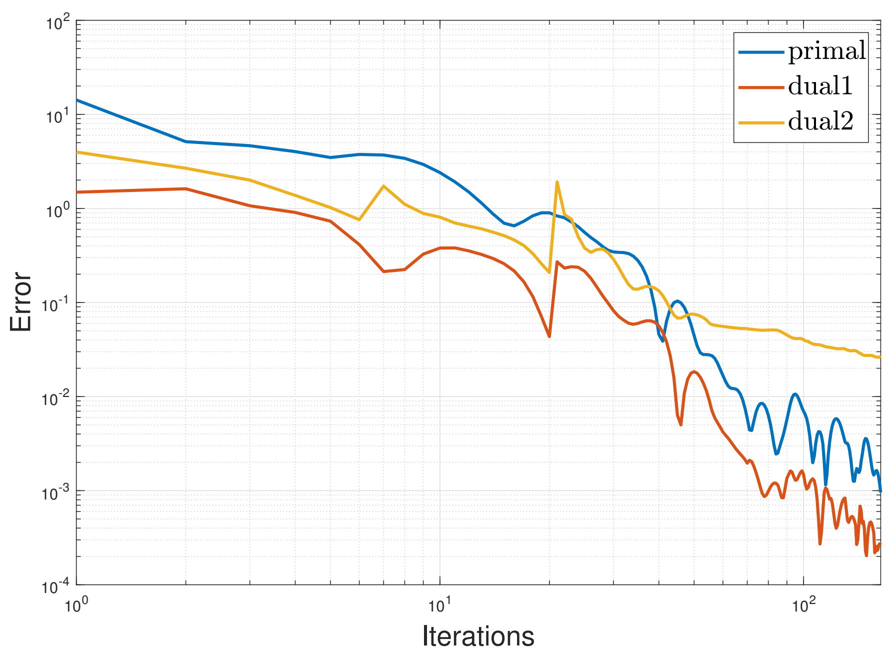


Figure 6.8: Convergence Behavior using ADMM on the Stochastic Data

We have also applied the ADMM to the expected data presented by the authors in the second column in Table 6.8, and we found that the ADMM converged at 199 iterations with 0.063319 seconds of CPU time. Figure 6.9 shows the convergence behavior of ADMM based on the expected

Table 6.12: Optimal Power Plant Sizing Based on the Expected Data

Power plant	Size in GW using Monte-Carlo	Size in GW using ADMM
Gas Turbine	1.78	1.7791
Coal	3.23	3.2286
Nuclear	4.07	4.0669
Hydroelectric	5.00	4.9937
Cost (billion $)	17.137 ± 0.053	17.2125 ± 0.0299

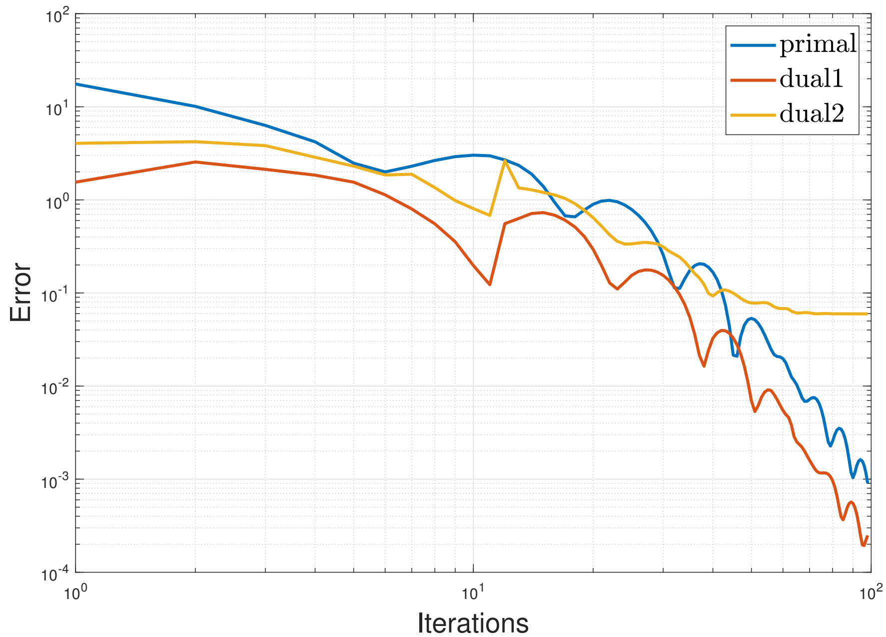


Figure 6.9: Convergence Behavior using ADMM on the Expected Data

data. We found that the cost using this data is $\$17.2125 \pm 0.0299$ billion. Figure 6.10 presents the convergence of the optimal value using ADMM. We provided the results of the two methods using the expected data in Table 6.12.

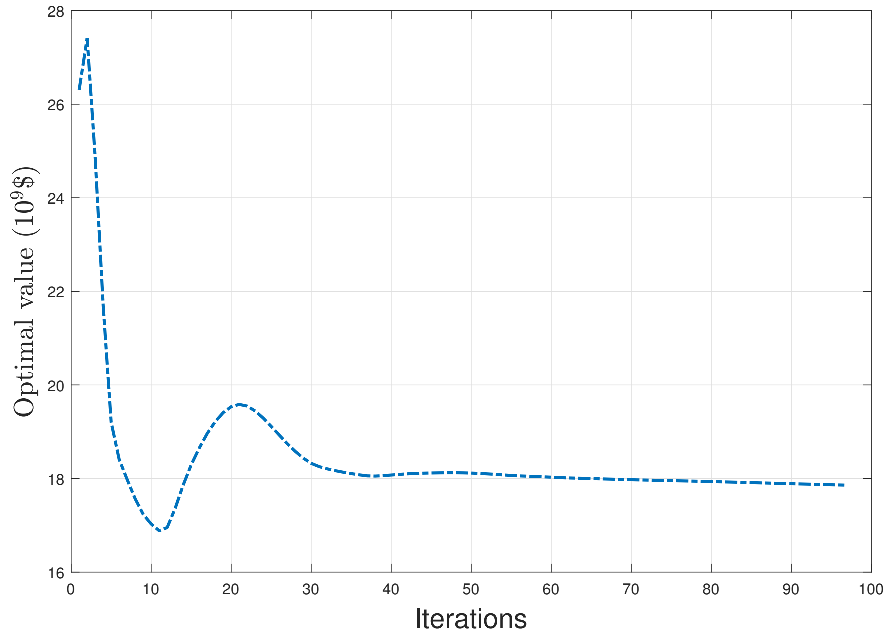


Figure 6.10: Optimal Objective Function using ADMM on the Expected Data

Chapter 7

Differential Equations Based Sizing of Renewable Energy Systems

7.1 Introduction

This chapter introduces a new approach for sizing hybrid renewable energy systems, which utilizes a differential equations-based model that integrates the dynamic cost. The proposed model is designed to minimize the cost of the system using pattern search optimization.

Pattern search is a derivative-free optimization algorithm used to find the optimal values of a bound constraints objective function with an initial starting point. The algorithm centered on the idea of searching for a pattern of points that converge to the optimal solution. The algorithm is beneficial for derivatives-free optimization.

The basic idea of pattern search is to start with an initial guess of the optimal solution and then iteratively explore the search space. The algorithm uses a set of pattern points, which are chosen based on the current search direction, and then evaluates the function at these points to determine the next search direction. This process is repeated until the algorithm stops or one finds a satisfactory solution [45].

One of the advantages of pattern search is that it does not require the computation of derivatives, which makes it well-suited for functions that are difficult or expensive to differentiate, especially for the case of our problem. Additionally, it can escape from poor solutions easily. There are variations of the pattern search algorithm. The well-known one is the Mesh Adaptive Direct Search (MADS) method [101]. Some of the most common uses of pattern search include optimization of engineering design problems, parameter estimation in mathematical models, and function optimization in machine learning.

In the context of renewable energy systems, pattern search optimization can determine the optimal configuration of the system components, such as the number of solar panels, wind turbines, or energy storage systems. The objective function, in this case, is usually to minimize the cost of the system while satisfying constraints, such as budget limit, energy demand, and system reliability.

AlHajri et al. [102] presented a method for optimizing the extraction of solar cell parameters using the pattern search algorithm. Their research aims to improve the accuracy of the extracted solar cell parameters, which are crucial for optimizing the performance of photovoltaic systems. They used a pattern search algorithm to find the optimal parameters for solar cells based on the data collected from measurement devices. Their study showed that the pattern search algorithm effectively extracts solar cell parameters. They highlighted the potential for using the pattern search algorithm in optimizing photovoltaic systems to increase their efficiency and performance.

Also, Kumar et al. [103] proposed an algorithm for automatic online parameter estimation of an Automatic Generation Control (AGC) system with the impact of wind power. Their algorithm uses a pattern search method, which determines the optimal parameters that improve the performance of the AGC system. They tested their algorithm and evaluated its accuracy and effectiveness. They showed that their algorithm could effectively handle the effects of wind power on the AGC system and improve its performance using pattern search optimization.

As renewable energy technologies continue to evolve and become more complex, pattern search will probably be a beneficial tool for optimizing their performance and efficiency. Using pattern search optimization, engineers and energy analysts can find the most cost-effective solution for a renewable energy system, allowing them to make informed decisions about the size and configuration of the system components. Doing so can help to optimize the performance and efficiency of renewable energy systems and improve their overall sustainability and profitability.

In this chapter, we propose a different modeling approach to finding the optimal configurations of the hybrid renewable energy system. We have designed the objective function in the previous chapter to find the optimal cost and sizes of the renewable energy system for a 20-year project, considering population growth and the corresponding increase in demand. However, this approach to system sizing is based on the assumption that the population and energy demand in the area will continue to grow over time. To accommodate this projected increase in demand, this approach will build more power plants than initially required to run the system. This is especially true for the first few years of the project. However, this can lead to the underutilization of resources in the short term because some of the system's components may remain unused. This is because the initial load demand can be met without utilizing all the power plants that were built, resulting in higher operational costs and a waste of resources.

To address this issue, we formulate a new objective functional where the second-stage variable y is a function of time, and the first-stage variable x is the control variable. The performance index is an integral (cost) for over 20 years. A differential equation-based model is then constructed using the constrained system. However, since the variable x is scalar-valued, the conventional optimal control method, such as Pontryagin's Maximum Principle, is not applicable. Therefore, we decided to use the pattern search approach. The results obtained by our formulation make sense in that it allows incremental decisions on x. This approach involves constructing only the power plant components that are required for the first year, followed by building other components in the second and third years. This incremental process allows for efficient budget allocation over multiple years rather than having the entire budget available upfront for the project's entirety.

This dynamic approach offers several advantages. First, it allows for a more flexible and adaptable system, as we can adjust the components each year to meet changing demands. Second, it results in a more cost-effective solution, as we do not waste resources on constructing plants that are not immediately needed. Also, the total budget for the whole planned duration is not required to be available at the beginning of the project, as only the yearly budget is needed to expand the existing renewable system based on the growing demand. Also, it enables a more sustainable energy system, as it allows for the gradual expansion of renewable energy sources over time rather than a sizable and potentially unsustainable increase all at upfront once, which requires maintenance costs.

Despite the growing interest in the renewable energy field, to the best of our knowledge, no research has been conducted on sizing renewable energy systems using differential equation-based modeling. We have addressed this gap in the literature. It has the potential to provide new and innovative solutions for optimizing the performance of renewable energy systems. In this context, it is significant to note that the pattern search algorithm is used to find the best solutions for complex optimization problems. By applying the dynamical approach, it can be possible to create more accurate models for sizing renewable energy systems, as we have shown below.

The rest of this chapter is organized as follows. Section 7.2 presents a detailed description of the optimization problem and outlines the mathematical formulation based on differential equations. Following this, Section 7.3 delves into the numerical results and provides an in-depth discussion of the presented methodology. Additionally, it includes a comparison of the costs obtained by the differential equation-based model and the one found in Chapter 6 using ADMM.

7.2 Mathematical Formulation

The objective functional (performance index) in this context is used to optimize the cost and sizes of a renewable energy system. We formulate it as an integral equation that considers the cost of the system over time. The purpose of this objective functional is to minimize the overall cost of the system by considering both the construction and operating costs of the renewable energy components from the beginning of the project until year t.

The optimization problem is solved using two distinct techniques: pattern search and the system of differential equations. Pattern search helps to find the minimum value of the renewable sizes by searching for patterns in the function's behavior. On the other hand, differential equations are mathematical expressions that describe how the state of variables changes over time. We use a system of differential equations to represent the dynamic of second-stage variables. By using this method, it is possible to find the optimal solution for the renewable energy system that satisfies both the budget and demand constraints.

The performance index, written as an integral expression, allows for a comprehensive analysis of the cost of the system over time. This is important because it enables the decision-makers to make informed decisions about the size and cost of the renewable energy system by considering not just the initial cost of construction, but also the ongoing operating costs. The budget and demand constraints ensure that the solution found is feasible and practical, taking into account the available resources and the energy demand of the system. The model of the problem is as follows.

We consider the second-stage variable as a function of time: $y(t)$. The operating cost over 20 years can now be written as

$$\int_0^{20} \sum_{\omega=1}^{s} p_\omega q_\omega^T(t) y_\omega(t) dt \tag{7.1}$$

which we minimized subject to

$$T_\omega x^t + W_\omega y_\omega(t) = h_\omega(t), \tag{7.2}$$

where $y_\omega(t)$ and $h_\omega(t)$ are vectors of appropriate dimensions under scenario ω. Assuming that we have a finite number of scenarios, where ω takes the values from 1 to s. Here, $x^t \in \mathbb{R}^{n_1}$ is the capacity in kW to be built for the power plants at time t; p_ω the probability associated with scenario ω; $q_\omega(t) \in \mathbb{R}^{n_2}$ is the operating cost of the power plants in $\$/kWh$ at time t under scenario ω; $y_\omega(t) \in \mathbb{R}^{n_2}$ the amount of electricity capacity used to produce electricity by the power plants at time t measured in kW under scenario ω; $h_\omega(t) = (\mathbf{0}, d_\omega(t))^T \in \mathbb{R}^{m_2}$ where $\mathbf{0} \in \mathbb{R}^{n_1}$ and $d_\omega(t)\mathbb{R}^{m_2-n_1}$ is the load demand at time t in kW under scenario ω; and $T_\omega \in \mathbb{R}^{m_2 \times n_1}$ and $W_\omega \in \mathbb{R}^{m_2 \times n_2}$ are matrices ensuring the capacity and demand constraints under scenario ω. Here, the time t represents a year.

The reason for $q_\omega(t)$ is also a function of time is that it increases by 3% annually. The same arguments apply for the demand $d_\omega(t)$, and hence $h_\omega(t)$ is treated as a function of time. The value of the control variable x is taken so that $T_\omega x^t + W_\omega y_\omega(t) = h_\omega(t)$ hold $\forall t$.

We now construct the following functional

$$F(y_\omega(t)) = \frac{1}{2}\left\|T_\omega x^t + W_\omega y_\omega(t) - h_\omega(t)\right\|^2. \tag{7.3}$$

We would like to achieve the minimum value zero of $F(y_\omega(t))$ by the following trajectory of $y_\omega(t)$ such that

$$\begin{aligned}\dot{y}_\omega(t) &= -\nabla_{y_\omega(t)} F(y_\omega(t)) \\ &= -W_\omega^T W_\omega y_\omega(t) + W_\omega^T(h_\omega(t) - T_\omega x^t).\end{aligned} \tag{7.4}$$

Clearly, $y_\omega(t)$ is influenced by the x^t used; therefore, $y_\omega(t)$ can now be treated as our state variable. Hence, we define the following optimal controlled problem

$$\begin{aligned}&\min_x \int_0^{20} \sum_{\omega=1}^{s} p_\omega q_\omega^T(t) y_\omega(t)dt, \qquad \text{such that} \\ &\dot{y}_\omega(t) = -W_\omega^T W_\omega y_\omega(t) + W_\omega^T(h_\omega(t) - T_\omega x^t).\end{aligned} \tag{7.5}$$

The value $t = 0$ is the beginning of the first year, and hence $y_\omega(0) = \mathbf{0} \in \mathbb{R}^{n_2}, \forall\omega$. In order to optimize Problem (7.5), we introduced the new variable $z(t)$ such that

$$z(t) = \int_0^{t} \sum_{\omega=1}^{s} p_\omega q_\omega^T(\tau) y_\omega(\tau)d\tau$$

$$0 \leq t \leq 20.$$

Using the Fundamental Theorem of Calculus, differentiating the objective function $z(t)$ with respect to time, we get the following differential equation

$$\dot{z}(t) = \sum_{\omega=1}^{s} p_\omega q_\omega^T(t) y_\omega(t), \quad \text{with} \quad z(0) = 0. \tag{7.6}$$

Now, the objective function can be written as $\min_x z(20)$. The optimization problem becomes

$$\begin{aligned}
\min_{x} \quad & z(20) \qquad \text{subject to} \\
\dot{y}_\omega(t) &= -W_\omega^T W_\omega y_\omega(t) + W_\omega^T (h_\omega(t) - T_\omega x^t), \\
\dot{z}(t) &= \sum_{\omega=1}^{s} p_\omega q_\omega^T(t) y_\omega(t). \\
z(0) &= 0, \quad y_\omega(0) = \mathbf{0}, \forall \omega.
\end{aligned} \tag{7.7}$$

For a given x^t, the objective function $z(20)$ can be evaluated by solving Problem (7.7). We have solved (7.7) by the Euler method using time discretization by taking the step length $L = 0.2$. The reason that we selected the step length is 0.2 is that we divided the year into five block demands. Hence, $t = 0$ is the beginning of the first year; $t = 1$ is the beginning of the second year, and so on.

We now explain the component of x^t with an example. Assume that due to the budget restriction, the entire hybrid system will have to be built over three years. This means that at $t = 0$ (i.e., the beginning of the first year), there are four decision variables $x_1^1, x_2^1, x_3^1, x_4^1$; at $t = 1$ (i.e., the beginning of the second year) there are another four decision variables, say $x_1^2, x_2^2, x_3^2, x_4^2$; and similarly the variables $x_1^3, x_2^3, x_3^3, x_4^3$ at $t = 2$ (i.e., the beginning of the third year). Hence, the value of x^t used at each step can now be defined as follows:

At $t = 0$ until $t = 0.8$, $x^0 = \hat{x}^0 = (x_1^1, x_2^1, x_3^1, x_4^1)^T \in \mathbb{R}^4$.

At $t = 1$ until $t = 1.8$, $\hat{x}^1 = \hat{x}^0 + x^1 \in \mathbb{R}^4$, where $x^1 = (x_1^2, x_2^2, x_3^2, x_4^2)^T \in \mathbb{R}^4$.

At $t = 2$ until $t = 20$, $\hat{x}^2 = \hat{x}^1 + x^2 = x^0 + x^1 + x^2 \in \mathbb{R}^4$, where $x^2 = (x_1^3, x_2^3, x_3^3, x_4^3)^T \in \mathbb{R}^4$.

Hence, the first component $x_1^1 + x_1^2 + x_1^3$ of $\hat{x}^2$ means that x_1^1 is the decision at $t = 0$; x_1^2 is the decision at $t = 1$; x_1^3 is the decision at $t = 2$. The same explanation applies to other components.

Implementing yearly budget constraints is critical for the successful construction of power plants. The budget serves as a limit for constructing new power plants each year, ensuring that the financial resources are allocated effectively and sustainably. If the construction costs for the power plants exceed the budget, we penalize the objective function with a penalty parameter. This penalty is to prevent violating the budget constraints.

The purpose of the budget constraints is to minimize the objective functional while satisfying the yearly budget limitations. We model it by selecting the maximum value between the costs associated with the power plants and zero. In other words, if the cost is within budget and the difference between the cost and the allocated budget for the specific year is a negative value, which is then compared to zero, the higher between the two values is selected. This ensures that the budget constraint is met while minimizing the objective function.

Let b^t denote the available budget at year t. The objective function can now be augmented to stay within the budget:

$$\min_x z(20) + \sum_{t=0}^{19} \mu_t \max\{c^T x^t - b^t, 0\},$$

where $c \in \mathbb{R}^4$ is the vector of the capital cost for all renewable components.

If b^t is given each year for three years, then the objective function becomes

$$\min_x z(20) + \mu_0 \max\{c^T x^0 - b^0, 0\} + \mu_1 \max\{c^T x^1 - b^1, 0\} + \mu_2 \max\{c^T x^2 - b^2, 0\}, \tag{7.8}$$

where μ_0, μ_1, μ_2 are suitable penalty parameters, and x^0, x^1, and x^2 are decision made at $t = 0, t = 1$, and $t = 2$, respectively. The allocated budget for the first year is $b^0 = \$80,000$, for the second year $b^1 = \$25,000$, and for the thrid year is $b^2 = \$15,000$, for a total of $\$120,000$.

Now, we can further augment the cost function to include the minimization with respect to the capacities need to be installed, x, in addition to the operating cost $z(20)$:

$$\min_x z(20) + c^T(x^0 + x^1 + x^2) + \sum_{t=0}^{2} \mu_t \max\{c^T x^t - b^t, 0\} \tag{7.9}$$

For this experiment, we have used the expected operating costs in Table 6.5 for $q(\cdot)$ as the starting point in the first year. Also, we used the demands $d(\cdot)$ found in Table 6.3. Here, we are assuming that the yearly growth rate of $r = 3\%$; then, the operating costs and demands over time can be expressed as a function of time through the following equations:

$$
\begin{aligned}
q_\omega(t) =& q_\omega(t-1)\,(1+r)^t \\
d_\omega(t) =& d_\omega(t-1)\,(1+r)^t\,.
\end{aligned}
$$

The equations show how the operating costs and demands change from one year to the next, considering the yearly growth rate. The value of $q_\omega(t-1)$ and $d_\omega(t-1)$ represents the operating costs and demands, respectively, from the previous year, and $(1+r)^t$ calculates the growth factor.

Algorithm 3 presents the pseudo-code for the proposed approach.

Algorithm 3 Differential Equations-based algorithm using Pattern Search Optimization

Initialize the dynamical differential equations-based model with system parameters
Set the initial feasible solution x^0 and the positive spanning set D
Set the initial step size Δ_0 and the stopping tolerance Δ_{tol}
Set the function at (7.9) to be optimized as the total cost for 20 years
Evaluate the function at the starting point x^0
Set counter $k \leftarrow 0$ and $i \leftarrow 1$
Solve the differential Equation (7.4) using the starting point
Use the solution of Equation (7.4) to solve Equation (7.6) with system parameters
while $\Delta_k \geq \Delta_{tol}$ **do**
    Evaluate the objective function f in (7.9) at the trial point $p_i = (x^k + \Delta_k d_i) \in P_k, \quad d_i \in D$
    Set the solution of Equation (7.4) and Equation (7.6) with the new trial point p_i
    if $f(p_i) < f(x^k)$ **then**
        Update the current best point $x^{k+1} \leftarrow p_i$
        Increase $k \leftarrow k+1$
        Set $i \leftarrow 1$
        Expand the mesh by increasing $\Delta_{k+1} \leftarrow 2\Delta_k$
    else
        Increase $i \leftarrow i+1$
        if $i > 2n$ **then**
            Set $x^{k+1} \leftarrow x^k$
            Increase $k \leftarrow k+1$
            Set $i \leftarrow 1$
            Reduce the mesh by decreasing $\Delta_{k+1} \leftarrow \frac{1}{2}\Delta_k$
        end if
    end if
end while
Return the optimal solution $x^* = x^k$

7.3 Numerical Results and Discussion

The pattern search algorithm and the solution to the differential equations are implemented using MATLAB 2022b. We run the numerical experiments on a computer with a Core i7 processor and 16 GB of RAM. With this combination of hardware and software, optimization experiments can run efficiently and effectively, producing accurate results promptly.

Table 7.1 summarizes the optimal size and cost of four types of renewable energy plants over a period of three years: solar PV, wind turbine, batteries, and fuel cell. The size of each plant is measured in kW, and the total size at the end of the third year is given in the last column. Starting with the solar PV plant, the optimal size in the first year was 14 kW, while the new decision decreased to 3.75 kW in the second year and further to 1.87 kW in the third year, resulting in a total of 19.63 kW by the end of the third year. The same pattern can be observed for the wind turbine and battery plants, where the optimal size decreases year over year, totaling 9.0 kW and 8.0 kW, respectively, at the end of the third year. Also, the fuel cell plant has an optimal size of 8 kW in the first year, 2.5 kW in the second year, and 1.25 kW in the third year, totaling 11.75 kW.

Table 7.1 also shows the capital cost of the plants, which decreased from \$ 70,900 in the first year to \$ 21,100 in the second year and further to \$ 10,550 in the third year, resulting in a total of \$ 102,550. This decrease in cost is because of the expansion of new power plants based on the existing system instead of building the entire renewable energy system at once. On the other hand, the operating cost in the first year was \$ 5,431, while the accumulated operating cost by the second year was \$ 5,752, and \$ 5,966 by the third year, as shown in Figure 7.1. The decision to expand the system gradually is because of the yearly increase in demand, which we assumed in our case to be 3%. Table 7.1 provides valuable insights into the size and cost of various renewable energy plants over a period of three years. It highlights the trend of decreasing the size and cost of these plants, reflecting the decision to expand new power plants based on the existing system.

The significant jump in operating costs from 0 to 0.5 years in Figure 7.1, followed by a subsequent order of magnitude decrease from 0.5 to 3 years, can be attributed to the characteristics and requirements of the renewable energy system being analyzed. Initially, during the period from

Table 7.1: Optimal Power Plants Decision in kW Using Pattern Search and Differential Equations

Plant	1st year Decision	2nd year Decision	3rd year Decision	Total Size
Solar PV	14	3.75	1.87	19.63
Wind Turbine	6	2.0	1.0	9.0
Batteries	5	2.0	1.0	8.0
Fuel cell	8	2.5	1.25	11.75
Capital Cost ($)	70,900	21,100	10,550	102,550
Accumulated Operating Cost ($)	5,431	5,752	5,966	5,966
Total Cost ($)	76,331	26,852	16,516	108,516

0 to 0.5 years, there are substantial start-up and commissioning costs involved in establishing the renewable energy system. As a result, operating costs experience a noticeable increase during this initial phase.

However, once the system is fully operational and the start-up phase is completed, ongoing operating costs tend to decrease significantly. This reduction can be attributed to the inherent advantages of renewable energy systems, which generally have lower operational and maintenance costs compared to conventional energy sources. Renewable systems eliminate the need for significant fuel expenses or frequent maintenance, as they rely on freely available primary energy sources such as sunlight and wind. Consequently, the operating costs observed from 0.5 to 3 years are considerably lower, often reflecting the efficiency and cost-effectiveness of renewable energy technologies.

Figure 7.2 is a graphical representation of the operating cost $z(t)$ associated with a hybrid renewable energy system. The horizontal axis represents the years, and the vertical axis represents $z(t)$. The graph shows that the operating cost started at zero in year 0, where no power plants were built at the beginning. As time goes, the operating expenses steadily increased.

As shown in Figure 7.1, the operating cost is relatively low throughout the entire period. Renewable energy systems typically have lower operating costs than traditional fossil fuel-based systems. Also, it is due that renewable systems require less maintenance compared to conventional ones.

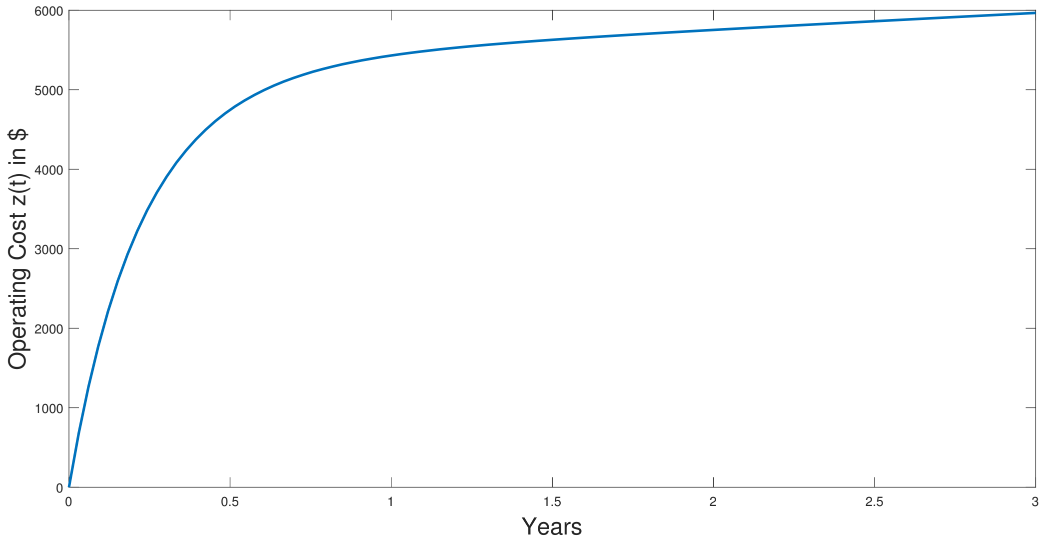


Figure 7.1: Operating Cost Over Three Years

The finding suggests that hybrid renewable energy systems require a substantial investment at the outset. However, over time, the ongoing expenses associated with their operation are minimal and ultimately balance out the initial cost. This is a common characteristic of renewable energy systems and highlights the importance of considering the life-cycle costs of energy systems when evaluating their feasibility. Also, it is crucial to note that the costs associated with renewable energy systems can vary greatly depending on the technology used, the location of the system, and other factors. Therefore, it is crucial to carefully consider the costs associated with any energy system to ensure that it is economically viable in the long run.

We want to compare the capital cost present value obtained by the differential equations-based model and the three-block ADMM from Chapter 6. We need to determine which model provides the lower values. This comparison will help us to make informed decisions about which model to use in future projects.

The three-block ADMM method calculates the capital cost at the beginning of the project and represents the present value. The total capital cost obtained by ADMM was $109,950$, see Table 6.7. Here, we will calculate the present value of the capital cost for the differential equations-based model as follows:

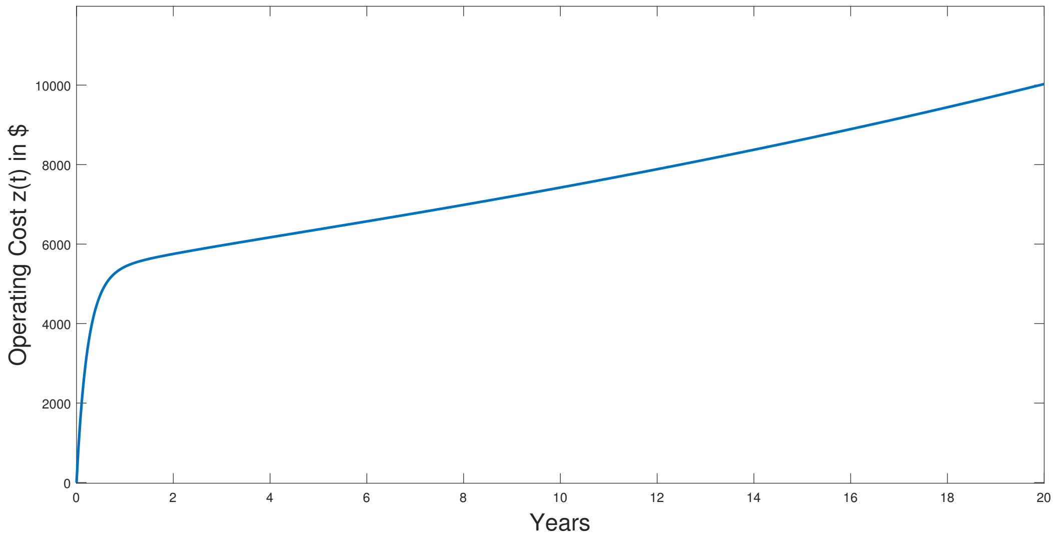


Figure 7.2: Operating Cost Over 20 Years

$$
\begin{aligned}
PV =& \sum_{t=0}^{2} \frac{C_t}{(1+r)^t} \\
=& \$70,900 + \frac{\$21,100}{(1+0.03)^1} + \frac{\$10,550}{(1+0.03)^2} \\
=& \$101,329,
\end{aligned} \tag{7.10}
$$

where the data $70,900$, $21,900$, and $10,550$ denote the capital cost the begining of year 1, year 2, and year 3, respectively, and are given in Table 7.1.

The formula (7.10) discounts the future cash flows at an interest rate of 3%. A comparison between the two costs $109,950$ and $101,329$ shows that the differential equations-based model decreases the total capital cost by 7.8%. This is a significant cost saving and could be a deciding factor in choosing which method to use for the project. The clear and coherent presentation of the results allows for an easy understanding of the superiority of the differential equations-based model over the three-block ADMM method in terms of capital cost.

Figure 7.3 depicts the process by which the objective function reaches its optimal value using the pattern search algorithm. The optimal value of the objective function (Equation (7.8)) is 108,516, which is the total cost after three years. The starting point of the algorithm was a rel-

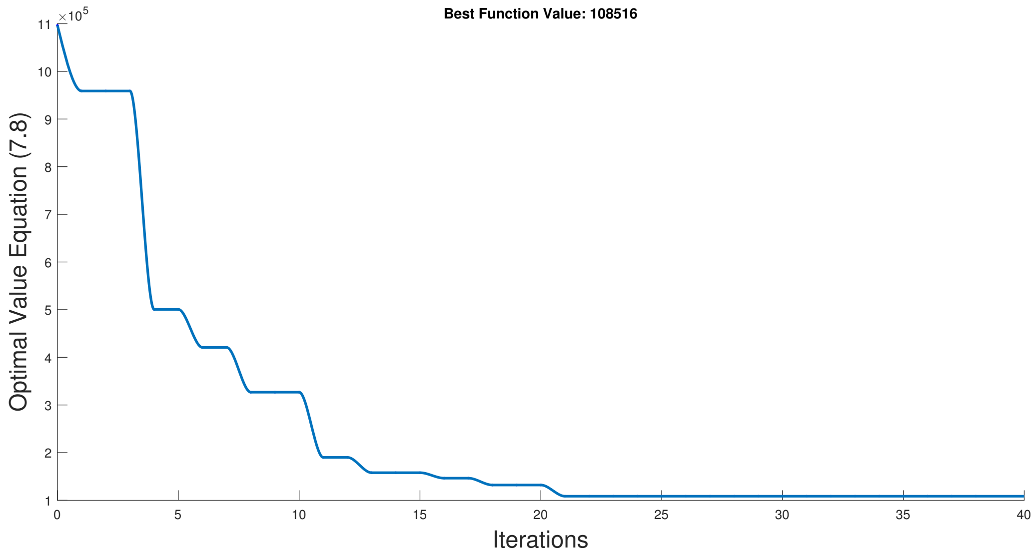


Figure 7.3: Convergence to the Optimal Value Using Pattern Search and Differential Equations

atively high value, which suggests that the initial guess for the solution was far from the actual optimal value. However, as the pattern search algorithm progressed, it began to search for solutions, gradually reducing the value of the objective function until it finally converged to the optimal value of 108,516.15. This convergence of the optimization algorithm indicates that the algorithm has found the best possible solution to the problem. It is important to note that the choice of the starting point can also impact the rate of convergence and the final solution obtained.

Chapter 8

Applications of three-block ADMM to Benchmark Problems

8.1 Introduction

This chapter discusses the numerical experiments of the three-block ADMM Algorithm 2 on seven benchmarks two-stage SP problems and compares its performance with the Progressive Hedging (PH) algorithm. Also, we discuss the convergence and CPU time of each algorithm.

The computer specification that runs the experiments has CPU Intel ® Core ™i7-7700 CPU @ 3.60GHz $\times 8$ and 15.5 GB of RAM. All the experiments run in MATLAB 2022b. The values $\epsilon = 10^{-3}$ and $k_{max} = 5 \times 10^4$ were used, respectively, in steps 21 and 22 of Algorithm 2. The primal and dual residuals that appear in all the relevant figures in this chapter are given in Line 19 of Algorithm 2, where dual1 and dual2 denote s^1 and s^2, respectively.

Each problem is solved with an initial starting feasible solution for the first-stage variable obtained using the phase 1 procedure of linear programming. Therefore, Algorithm 2 executes each problem with a feasible solution. We recorded the CPU times for all the solutions to measure the performance of each method. The residual graphs are all provided in a log-log scale.

Table 8.1 summarizes the dimensions of the tested problems in this chapter. They are all linear SP in their first and second stages. Using Sample Average Approximation, Linderoth et al. [6] provided lower and upper bounds for optimal solutions to the problems *LandS*, *Gbd*, *Ssn*, and *20term*. On the other hand, the problems *Assets* and *Phone* are given in [79]. All the input data for the problems in Table 8.1 are provided in SMPS (Stochastic Mathematical Programming System) format for stochastic linear programs. We converted the data from SMPS to MATLAB format ".mat" using the Python package pysmps [104], which parses MPS and SMPS file formats.

Table 8.1: Test Problem Data [6]

Name	Application	Scenarios	First stage size	Second stage size
LandS	Electricity planning	10^6	$(2, 4)$	$(7, 12)$
Assets	Asset management	3.75×10^4	$(5, 8)$	$(5, 8)$
Phone	Network planning	2^{15}	$(1, 9)$	(23,93)
Gbd	Aircraft allocation	6.5×10^5	$(4, 17)$	$(5, 10)$
Ssn	Telecom network design	10^{70}	$(1, 89)$	$(175, 706)$
20term	Vehicle assignment	1.1×10^{12}	$(3, 64)$	$(124, 764)$

In Table 8.1, the data in the last two columns are given in pair (m, n), where m is the number of rows, and n is the number of columns. The first stage size refers to the size of matrix A, and the second stage size refers to the size of matrix W, see Equation (4.13).

Table 8.2 reports the statistical bounds for the optimal value for problems *LandS*, *Gbd*, *Ssn*, and *20term* found by Linderoth et al. [6].

In this chapter, we present a set of benchmark problems that have been solved using the three-block ADMM algorithm, with numerical results for each problem provided. The rest of this chapter is organized as follows: Firstly, in Section 8.2, we discuss the electrical investment stochastic problem to fulfill the future demand with its two variations: small-size and large-size problems. Then, in Section 8.3, we introduce the asset management stochastic problem, which involves optimizing the allocation of assets in the presence of uncertainty. We present a network model for the problem, which considers the dependencies between the different assets. In Section 8.4, we move on to the telecommunication network planning problem, which involves optimizing the design of a telecommunication network to meet future growth. We provide a detailed description of the problem, including the mathematical formulation and the uncertainty model. Next, in Section 8.5, we present the aircraft allocation problem, which involves optimizing the allocation of aircraft to different routes by maximizing revenue under stochastic demand. In Section 8.6, we introduce the telecom network design problem by adding capacities to routes to minimize the expected number of unmet random demand requests. Section 8.7 presents the vehicle assignment problem, which involves directing vehicles to different locations to meet the point-to-point ship-

Table 8.2: Optimal Values for *LandS*, *Gbd*, *Ssn*, and *20term* for Various Scenarios

Problem	Scenarios	95% confidence intervals
LandS	50	$[225.71 \pm 0.12]$
	100	$[225.55 \pm 0.12]$
	500	$[225.16 \pm 0.12]$
	1000	$[225.70 \pm 0.13]$
Gbd	50	$[1655.86 \pm 1.34]$
	100	$[1656.35 \pm 1.19]$
	500	$[1654.90 \pm 1.46]$
	1000	$[1655.70 \pm 1.49]$
Ssn	50	$[12.68 \pm 0.05]$
	100	$[11.20 \pm 0.05]$
	500	$[10.28 \pm 0.04]$
	1000	$[10.09 \pm 0.03]$
20term	50	$[254317.96 \pm 19.89]$
	100	$[254304.54 \pm 21.20]$
	500	$[254320.39 \pm 27.36]$
	1000	$[254341.36 \pm 20.32]$

ments. Finally, in Section 8.8, we provide a comparison of the numerical results obtained using the three-block ADMM algorithm and the state-of-the-art Progressive Hedging algorithm. We discuss the strengths and weaknesses of each algorithm.

8.2 Electrical Investment Planning (LandS Problem)

8.2.1 Small Size LandS

Louveaux and Smeers [105] provided a power plant expansion model for future electricity generation based on determining the best investment in various power plants to fulfill future demand. It consists of four plants and three different demand blocks. The variable x_i describes the new capacity of power plant i, and y_{ij} is the production rate from plant i for block j. The investment cost, generation cost and load duration are $c = (10, 7, 16, 6)^T$, $q = (4, 4.5, 3.2, 5.5)^T$, and $\tau = \{10, 6, 1\}$, respectively [42]. All the variables are deterministic except the first demand $\mathbf{d}_1 = \omega$ which takes the values $\{3, 5, 7\}$ with the corresponding probabilities $\{0.3, 0.4, 0.3\}$. The other demands are deterministic and take the values $d_2 = 3$ and $d_3 = 2$. The minimum total capacity and budget constraints are given in the first-stage conditions, while the capacity restriction for each of the four plants and the demands are in the second-stage constraints. So, the resulting problem takes the following form:

$$\begin{aligned}
\min \quad & c^T x + \mathbb{E}_\omega \sum_{j=1}^{3} \tau_j q^T y \qquad \text{subject to} \\
& 10x_1 + 7x_2 + 16x_3 + 6x_4 \leq 120, \\
& \sum_{i=1}^{4} x_i = 12, \\
& \sum_{i=1}^{4} y_{i1} = \omega, \\
& \sum_{i=1}^{4} y_{ij} = d_j(\omega), \qquad j = 2, 3 \\
& x \geq 0, \quad y \geq 0.
\end{aligned} \tag{8.1}$$

Louveaux and Smeers [105] report the optimal first-stage solution to Problem (8.1) to be

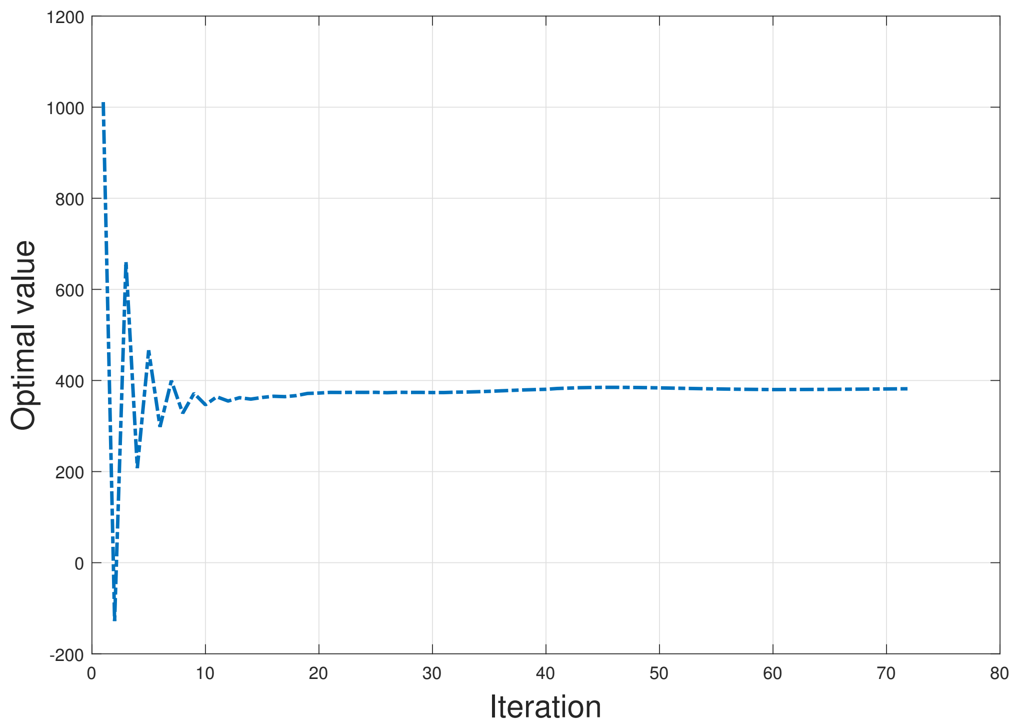
$$x^* = (2.666666, 4.0, 3.3333, 2.0)^T,$$

with the objective value of 381.853.

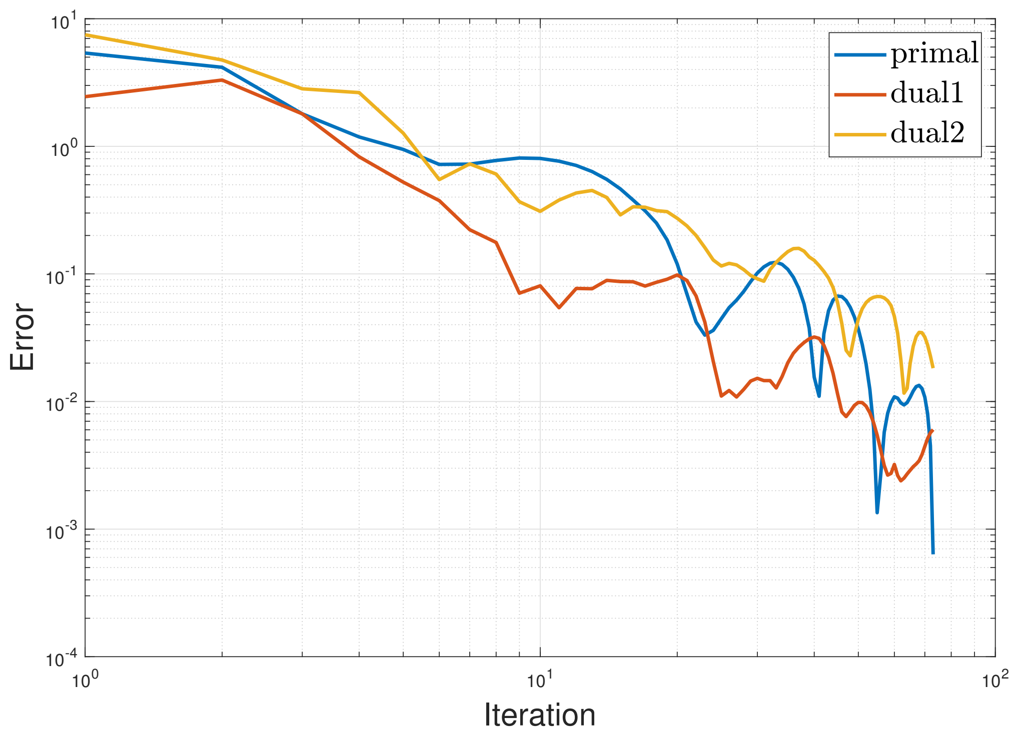
The objective values and residuals for the small-size LandS problem are shown in Figure 8.1 using Algorithm 2. The optimal first-stage solution was found to be

$$x^* = (2.6664, 4.0001, 3.3335, 1.9999)^T,$$

with the optimum value of 381.67 as shown in Figure 8.1a. Figure 8.1b presents that the convergence of the residuals took 73 iterations in 0.006 seconds.



(a) Convergence to the Optimal Value



(b) Residuals Convergence

Figure 8.1: Convergence Behavior of LandS Problem (Small Size).

8.2.2 Large Size LandS

This problem is a modified and extended version of the one discussed in 8.2.1 presented by [105]. The demand constraint, in this case, is given by

$$\sum_{i=1}^{4} y_{ij} \geq d_j, \quad j = 1, 2, 3,$$

where d_j's are given by

$$d_j = 0.04(k-1), \quad k = 1, 2, \cdots, 100, \quad j = 1, 2, 3,$$

each demand with a probability of 0.01. Demands are assumed equally independent; therefore there are $(100)^3 = 10^6$ scenarios with equal probability of 10^{-6} that make the problem with large dimension.

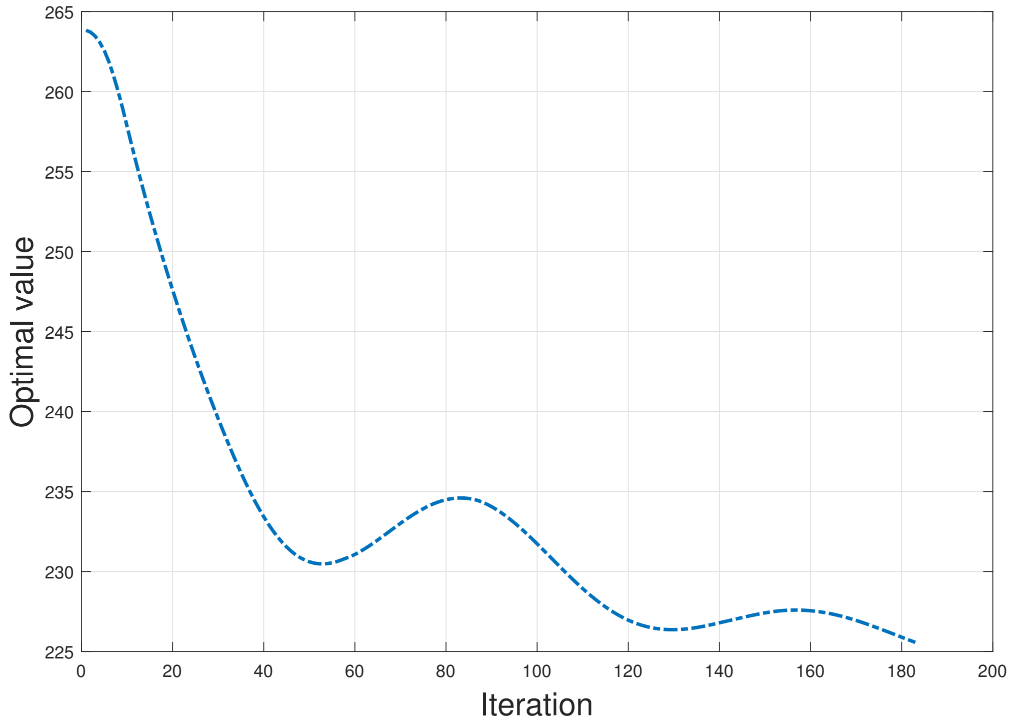
Figure 8.2 shows the numerical results for the large LandS problem with 500 scenarios. As shown in Figure 8.1a, the convergent optimal value for this problem is 225.85; see Table 8.2. Figure 8.1b shows that the algorithm 2 needs 183 iterations to converge, which takes 0.0254 seconds.

8.3 Network Model for Asset Management (Asset Problem)

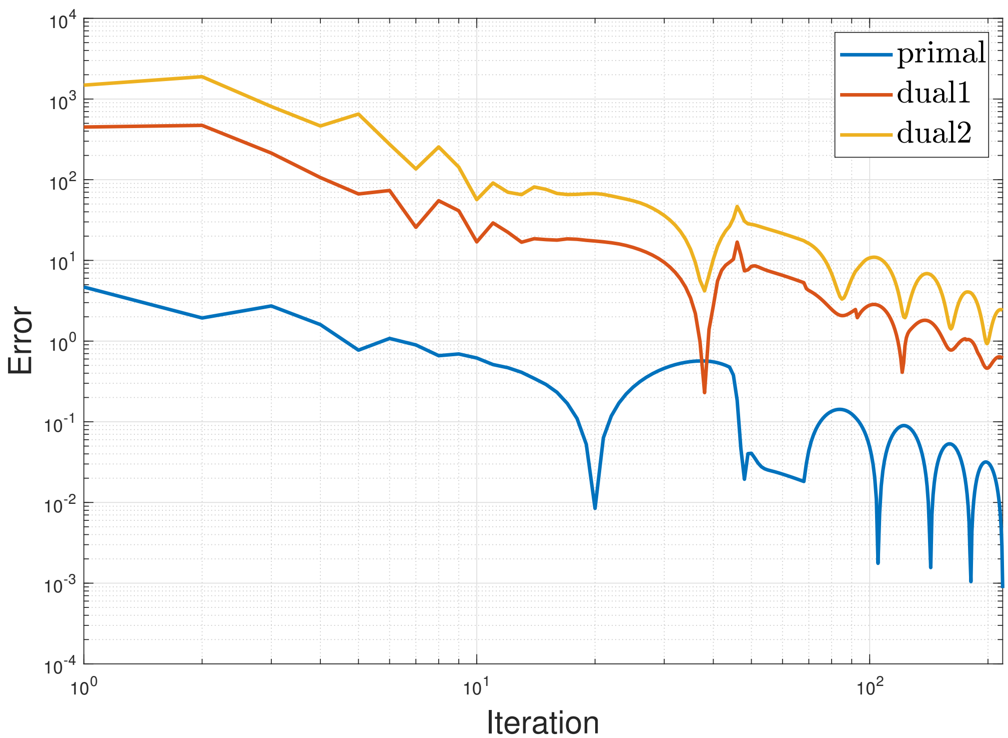
The management of assets or liabilities can be expressed as a network problem. In this problem, the assets are represented by nodes, while arcs represent the transactions. Five nodes are checking, saving, certificate of deposit, cash, and loads [79], where costs are fixed at every stage. This problem has a two-stage stochastic setting because the purchase or sale has deterministic costs, while the return on investment is random.

The set of nodes is denoted by $\mathcal{N}$, the set of arcs by $\mathcal{A}$, and the set terminal arcs by $\mathcal{T} \subset \mathcal{A}$.

The objective function is defined as $f_s(v, y(s)) \in C^1$, which is a convex function under scenario $s \in S$, where $v := \{v_{ij} : (i,j) \in \mathcal{A}_1\} \in \mathbf{R}^{n_1}$ and $y(s) := \{y_{ij}(s) : (i,j) \in \mathcal{A}_2\} \in \mathbf{R}^{n_2}$, where $n_1 = \mathsf{A\_1}$ and $n_2 = \mathsf{A\_2}$, respectively, $n = n_1 + n_2$. Additionally, the deterministic



(a) Convergence to the Optimal Value



(b) Residuals Convergence

Figure 8.2: Convergence Behavior of LandS Problem (Large Size).

and supply nodes are encoded in the vectors $b \in \mathbb{R}^{m_1}$ and $h \in \mathbb{R}^{m_2 \times |S|}$, where $m_1 = |\mathcal{N}_1|$ and $m_2 = |\mathcal{N}_2|$, respectively. The two-stage SP formulation to asset problem is given in the following form [79]

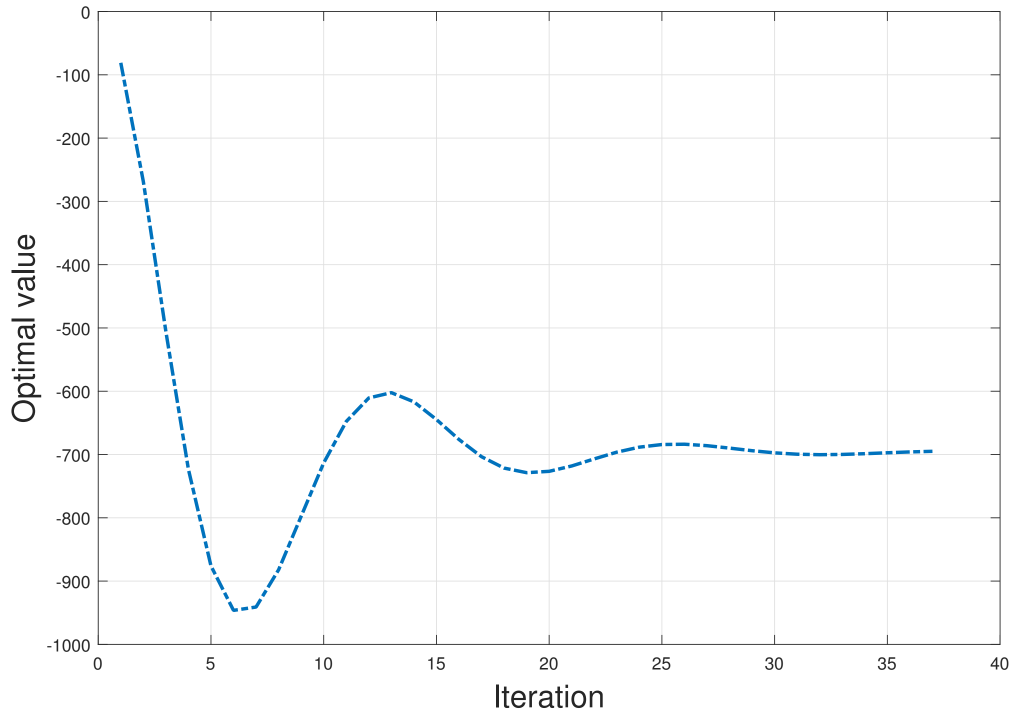
$$\begin{aligned}
\min \quad & \sum_{s \in S} p_s f_s(v, y(s)) \qquad \text{subject to} \\
& Av = b, \\
& T(s)v + W(s)y(s) = h(s), \quad \forall s \in S, \\
& l^1 \leq v \leq u^1, \quad l^2 \leq y(s) \leq u^2, \quad \forall s \in S.
\end{aligned}$$

For the complete mathematical description, we refer to [79].

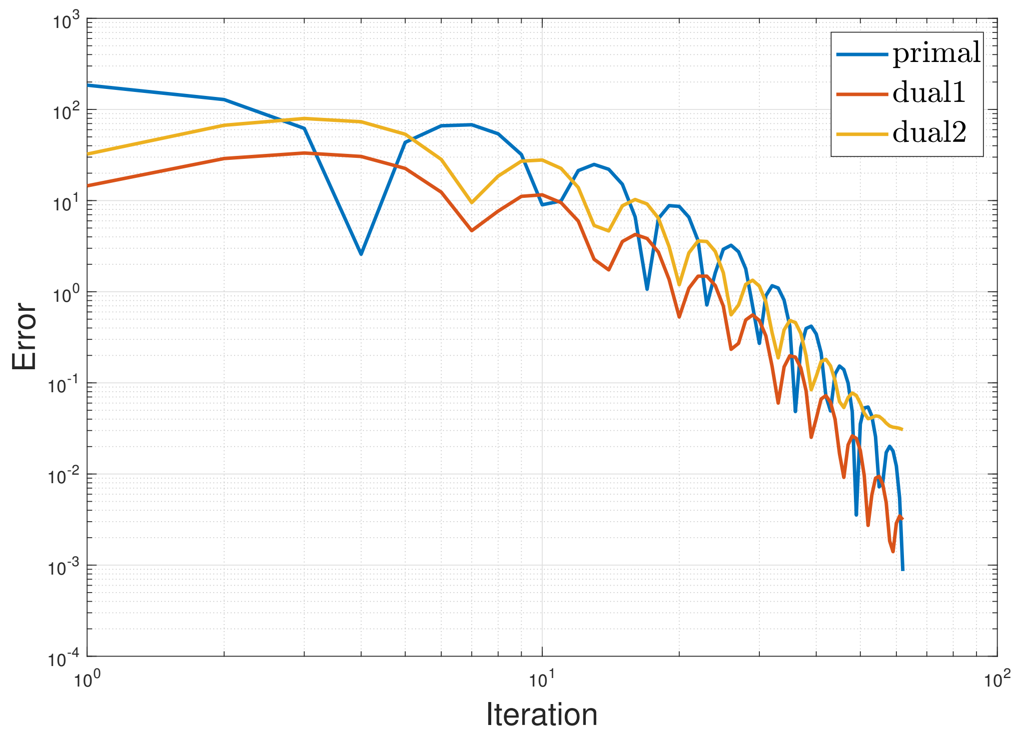
The problem has two sizes; the first has a 100 scenario, and the second has a $37,500$ scenario. The numerical results shown in Figure 8.3 correspond to the $37,500$ scenario; the 100 scenario problem shows similar results. As shown in Figure 8.3a, the three-block ADMM algorithm converges to the optimal value of -696.74, which is consistent with that reported in [79] and [106] matches. Figure 8.3b shows residual convergence, showing that the algorithm needs 37 iterations to converge in 0.0053 seconds.

8.4 Telecommunication Network Planning (Phone Problem)

From local telephone systems to the internet, telecommunication networks have become essential to modern life. Telecommunications managers need to plan for their networks' growth and expansion. A good understanding of the market's future needs and the network's existing capabilities are crucial in making these decisions. Before choosing where and how much to expand a telecommunications network, managers must first define their goals. For example, building access to more expensive long-distance telephone lines might offer higher returns on investment with a reduced impact on profitability. Next, telecommunications managers must assess their existing infrastructure by measuring the network's current capabilities and capital expenditures. They should also estimate future capital requirements for equipment and maintenance budgets. After evaluat-



(a) Convergence to the Optimal Value



(b) Convergence of the Residuals

Figure 8.3: Convergence Behavior of Asset Problem

ing current capabilities, network managers estimate expansion options based on cost, time, and performance expectations.

This problem considers networks that are extremely interconnected in which several routes can provide services to the demand of point-to-point requests. Besides, each route has one or more direct links. The objective, therefore, is to minimize unserved requests by adding new links while staying within the budget constraint.

The problem is expressed mathematically as

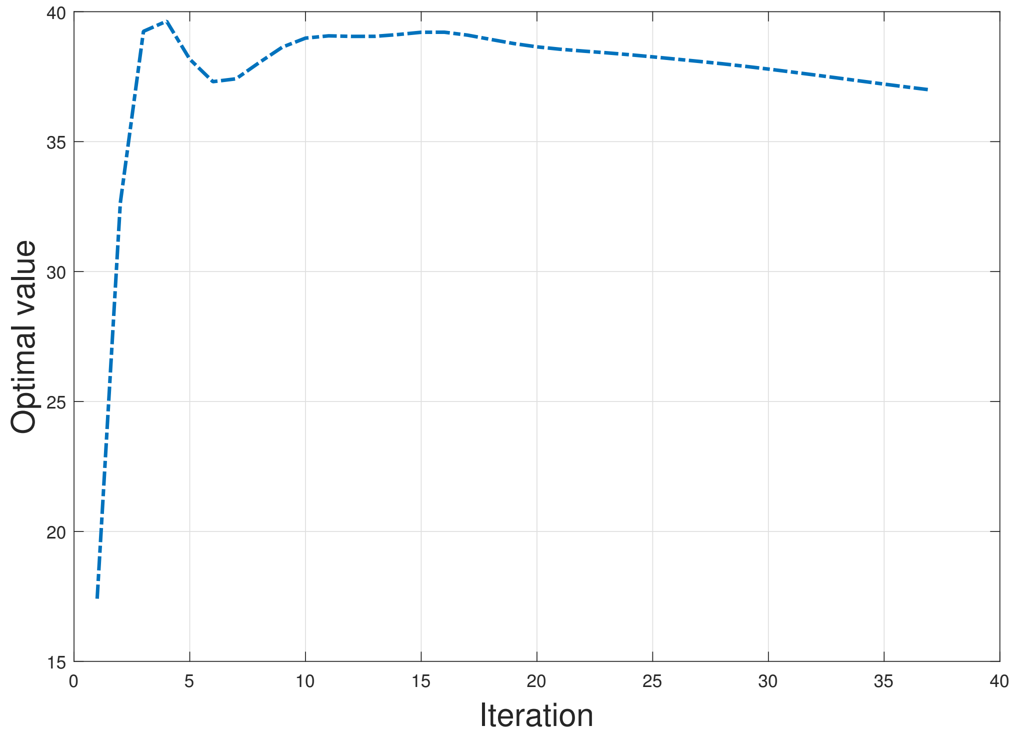
$$\min_{x} \quad \mathbb{E}_{\omega}[Q(x,\omega)] \qquad \text{subject to}$$
$$\sum_{j=1}^{n} x_j \leq b, \quad x \geq 0,$$

where $Q(x, \omega)$ is expressing the number of unserved requests. See[79] for a full mathematical description.

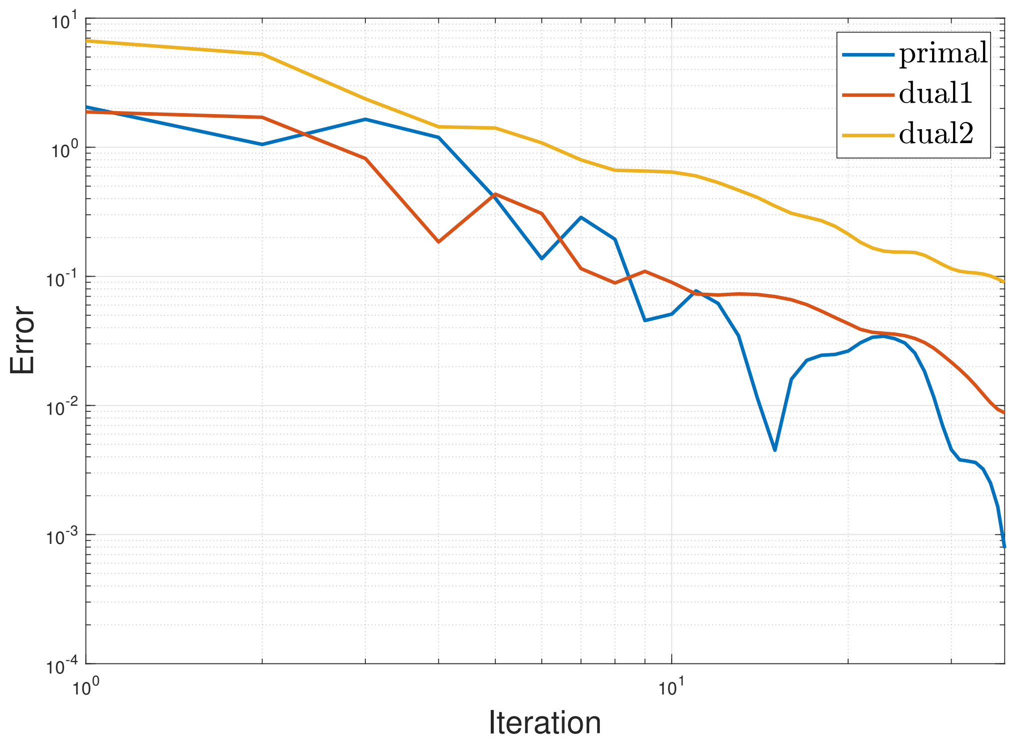
The scenario quantity in this example is 1000. In our numerical experiments we found that the scenarios of 50, 100, 500, 1000, and 5000 achieved the same optimal value at all sizes. Figure 8.4 shows the numerical results, where the algorithm converged in 29 iterations with 0.04 seconds of CPU time and an optimal value of 36.99, as shown in Figure 8.4a. The convergence of the residuals is shown in Figure 8.4b.

8.5 Aircraft Allocation (Gbd Problem)

An aircraft is the most popular mode of long-distance transportation worldwide. Airlines calculate the optimal route for their planes based on variables such as fuel consumption, time, and distance traveled. In this problem, four aircraft must be assigned to five routes to maximize the revenue under the stochastic demand. The first-stage variable encodes how many aircraft are assigned to each route, while the first-stage constraints limit the possible number of aircraft of each type. On the other hand, the second-stage variable denotes the number of transported passengers on each of the five routes, and the random demands represent the second-stage constraints. This



(a) Convergence to the Optimal Value



(b) Convergence of the Residuals

Figure 8.4: Convergence Behavior of Phone Problem

model has $646,425$ possible scenarios. The full detailed mathematical model is represented by [107].

We apply the algorithm to the gbd problem of different sizes. We choose to show the results obtained for the mid-scale 500 scenarios since other sized show similar results. Figure 8.5 shows the numerical experiments on the objective value and residual for the gbd problem. Figure 8.5a displays that the algorithm converges in 0.197 seconds in 184 iterations. The optimal value for this problem is 1649.9, which agrees with the results in the Table 8.2. Figure 8.5b demonstrates the convergence of the residuals.

8.6 Telecom Network Design (Ssn Problem)

This problem arises in network planning with random demand presented by Sen et al. [108]. The objective is to determine by the network manager where to add the capacity to the routes to minimize the expected number of unmet random demand requests. The request comes from a pair of nodes in the network to allocate a prespecified bandwidth in a time frame.

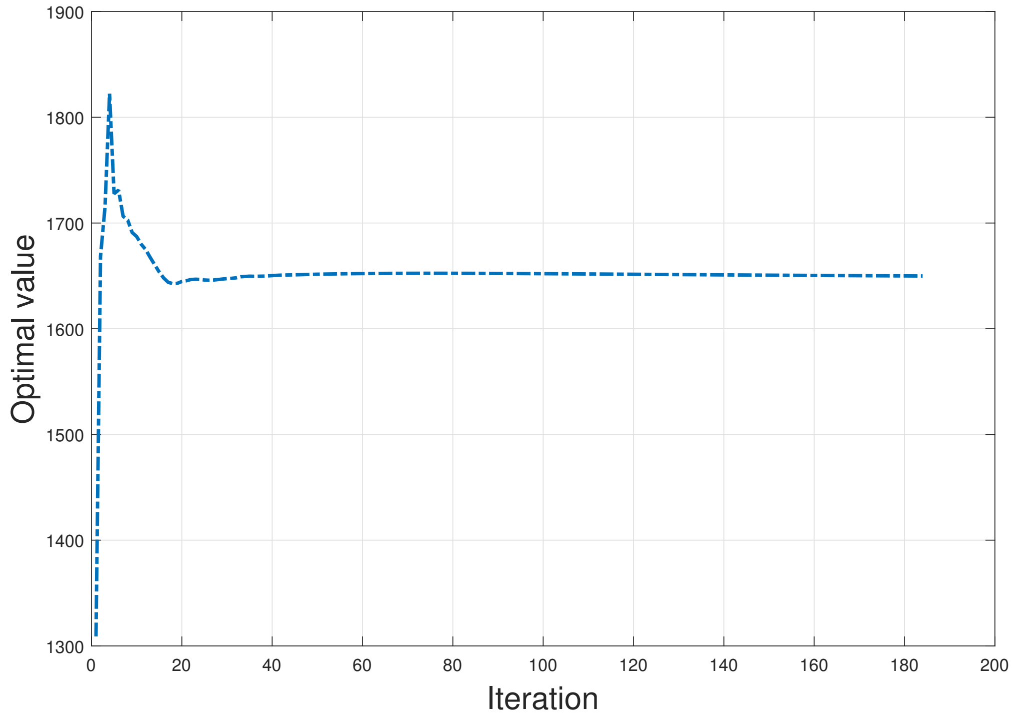
Sen et al. [108] formulated the problems as follows:

$$\begin{aligned} \min_x \quad & \mathbb{E}_\omega[Q(x,\omega)] \qquad \text{subject to} \\ & \sum_{j=1}^{n} x_j \leq b, \quad x \geq 0, \end{aligned}$$

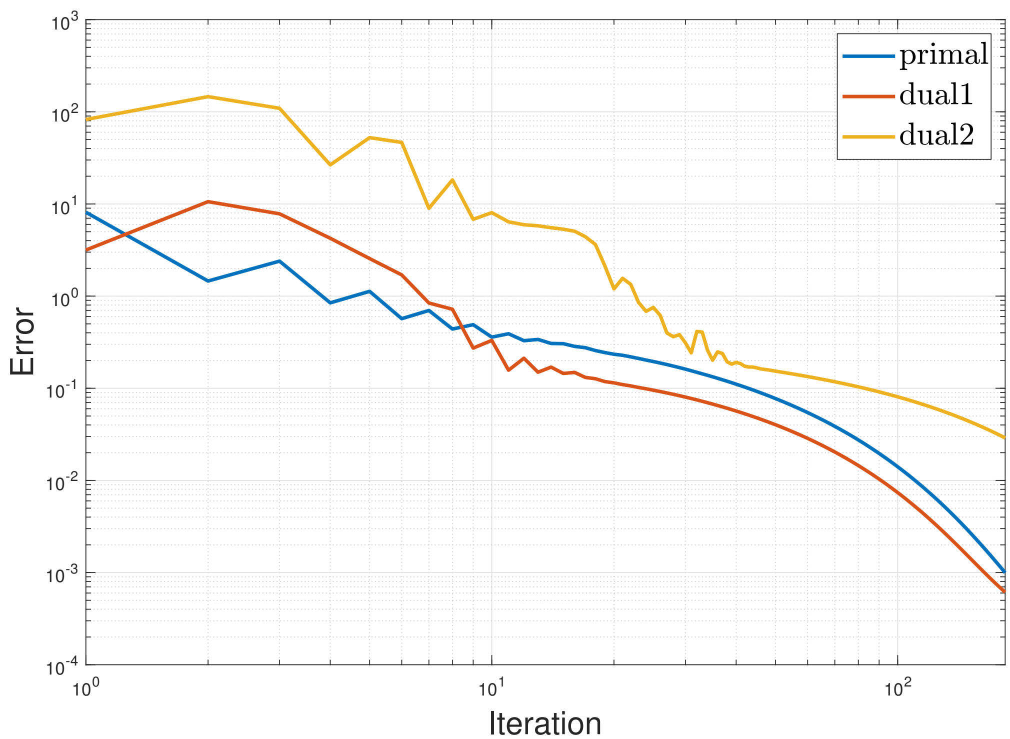
where x is the capacities to be added to the arc of the network, b is the total number of routes to be allocated, and $Q(x,\omega)$ define the total number of unserved requests and it given by

$$\begin{aligned} Q(x,\omega) := \min_{f,s}\{\sum_i s_i : \quad & Af \leq x + c, \\ & Ef + s = \omega, \quad f \geq 0, s \geq \}, \end{aligned}$$

where s is the vector of unsatisfied demands; the columns of A are vectors that represent the network topology; the components of f are the number of connections between each note pair for



(a) Convergence to the Optimal Value



(b) Convergence of the Residuals

Figure 8.5: Convergence Behavior of Gbd Problem

each particular route; c is the vector of existing capacities; the rows of E represent the total flow of a given route; and d is the vector of random demands. As shown in Table 8.1, this model has 10^{70} scenario sets. For more details, see [108].

We have run Algorithm 2 on the ssn problem using different sample sizes. Figure 8.6 shows the objective values and residuals for the ssn problem with 500 scenarios. Figure 8.6a indicates the algorithm converges to the optimal value in 60 iterations, which take 1.445 seconds. The optimal value we obtained for this problem is 10.54, which agrees with the results in Table 8.2 and [6]. Also, Figure 8.6b presents the convergence of the residuals.

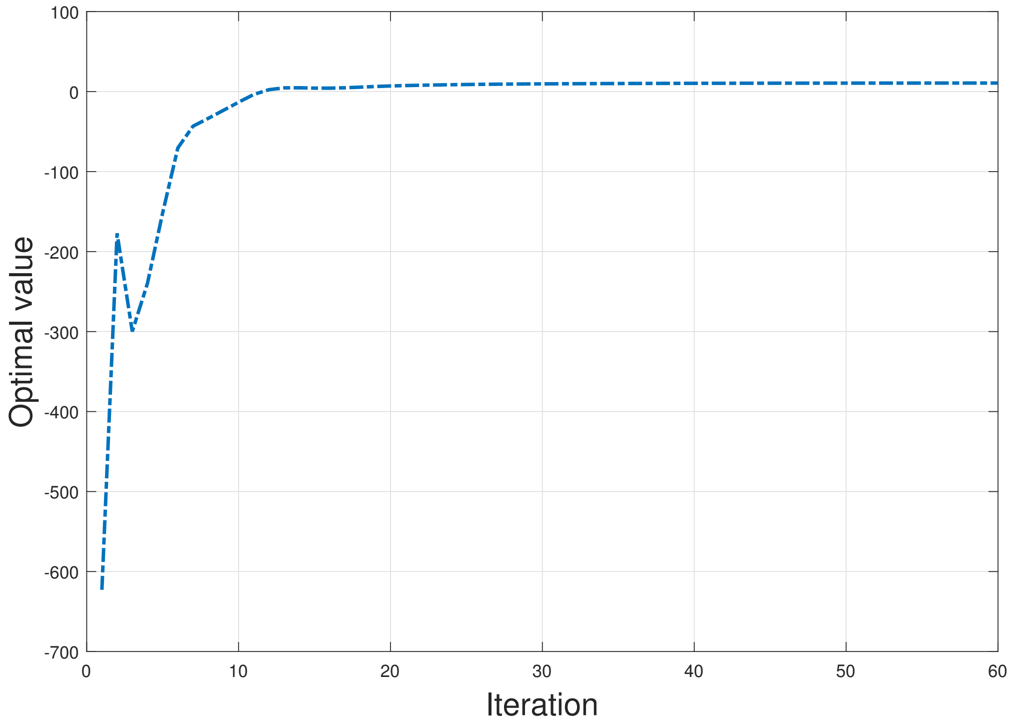
8.7 Vehicle Assignment (20term Problem)

Mak et al. [109] addressed this problem as a model of a motor freight carrier's operations problem. The first-stage variables are the early-morning locations of a fleet of automobiles. The second-stage variables direct the fleet via a network to satisfy point-to-point shipments; if these needs are not satisfied, some penalties will be imposed, and the day will end with the same fleet configuration as it began.

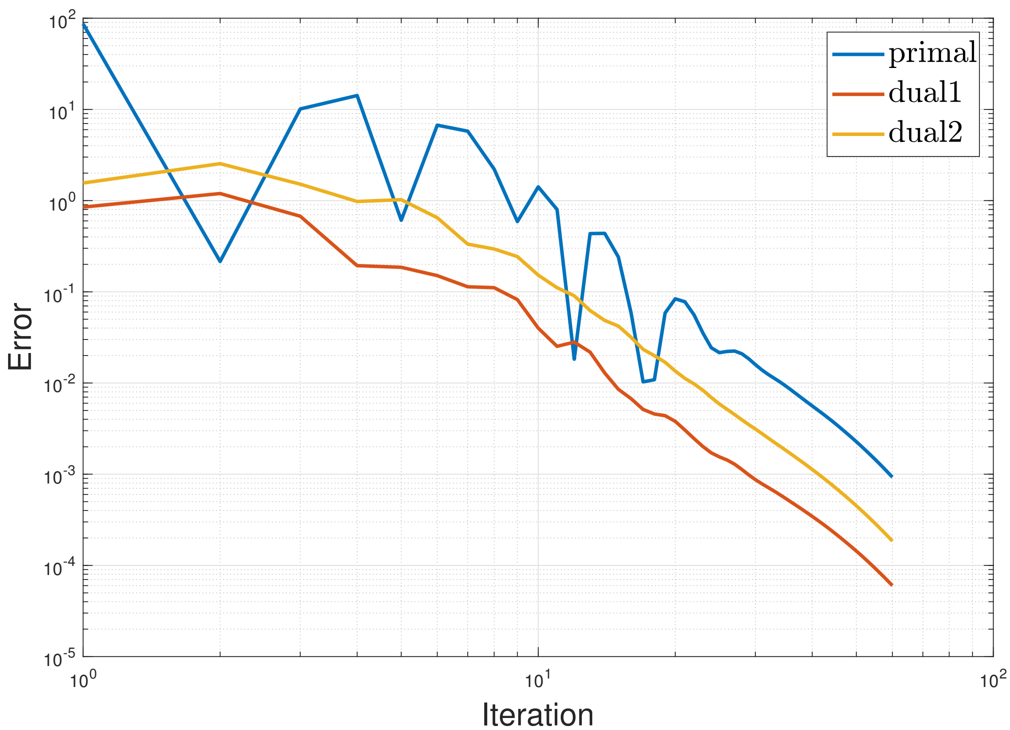
We have run Algorithm 2 on the problem using different sample sizes ranging from 50 to 1000 scenario. Figure 8.7 presents the numerical experiment for the objective values and the residuals. Figure 8.7a shows that the algorithm took 778 iterations to converge for a 100 scenario problem. The optimal value for this problem is 254350.01, which agrees with the result in Table 8.2 and [6]. Figure 8.7b shows the convergence of the residuals.

8.8 Numerical Comparisons and Discussion

This section discusses the numerical comparison of the three-block ADMM Algorithm 2 with the Progressive Hedging (PH) algorithm with respect to CPU time needed for convergence, the number of iterations, the percentage gap, and the convergence status for each problem. The numerical results of this comparison are presented in Table 8.3 and Figure 8.8. Also, we have provided a comparison of the Algorithm 2 with the pADMMts algorithm suggested by [81] with respect to

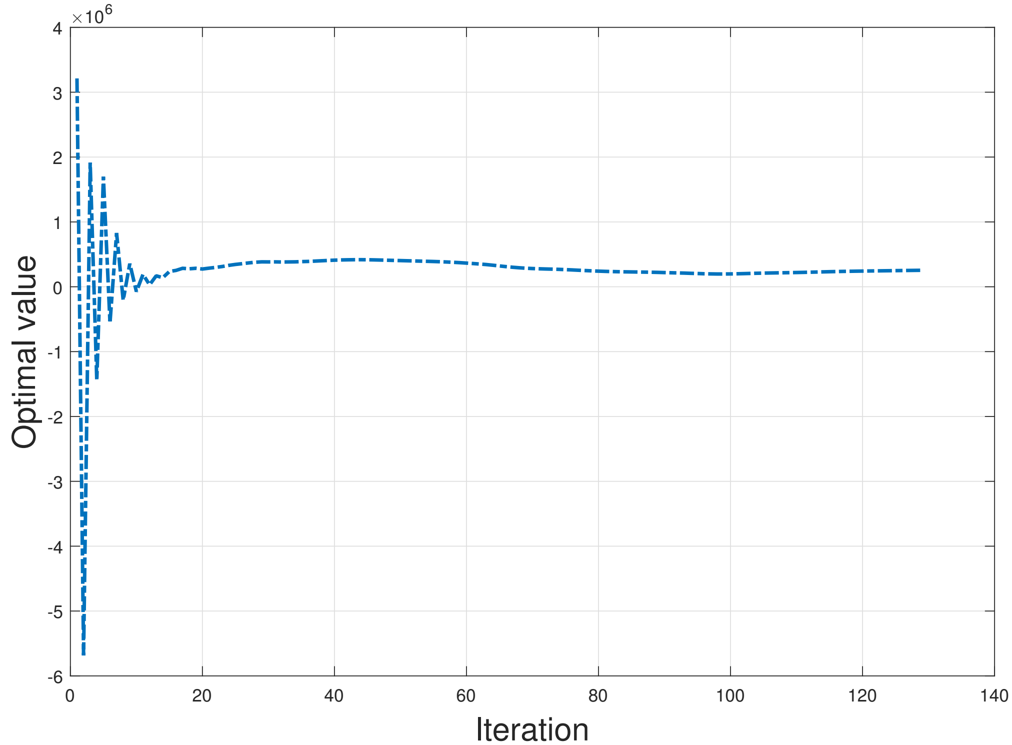


(a) Convergence to the Optimal Value

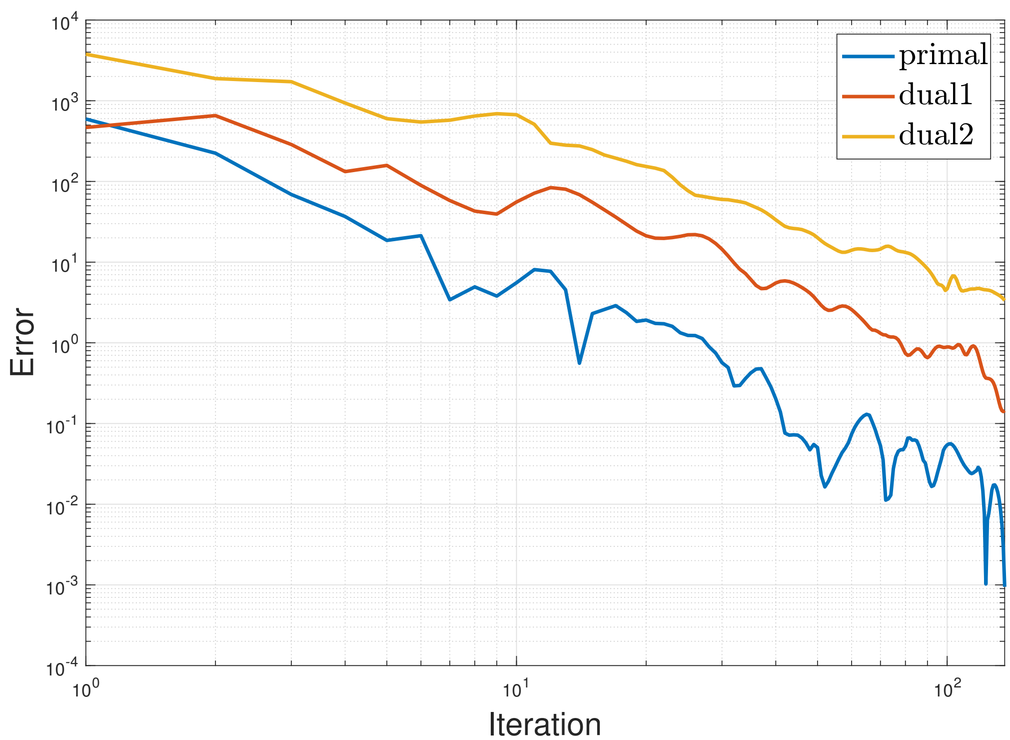


(b) Convergence of the Residuals

Figure 8.6: Convergence Behavior of ssn Problem



(a) Convergence to the Optimal Value



(b) Convergence of the Residuals

Figure 8.7: Convergence Behavior of 20term Problem

Table 8.3: Result Summary of the Comparison Between PH and ADMM Methods

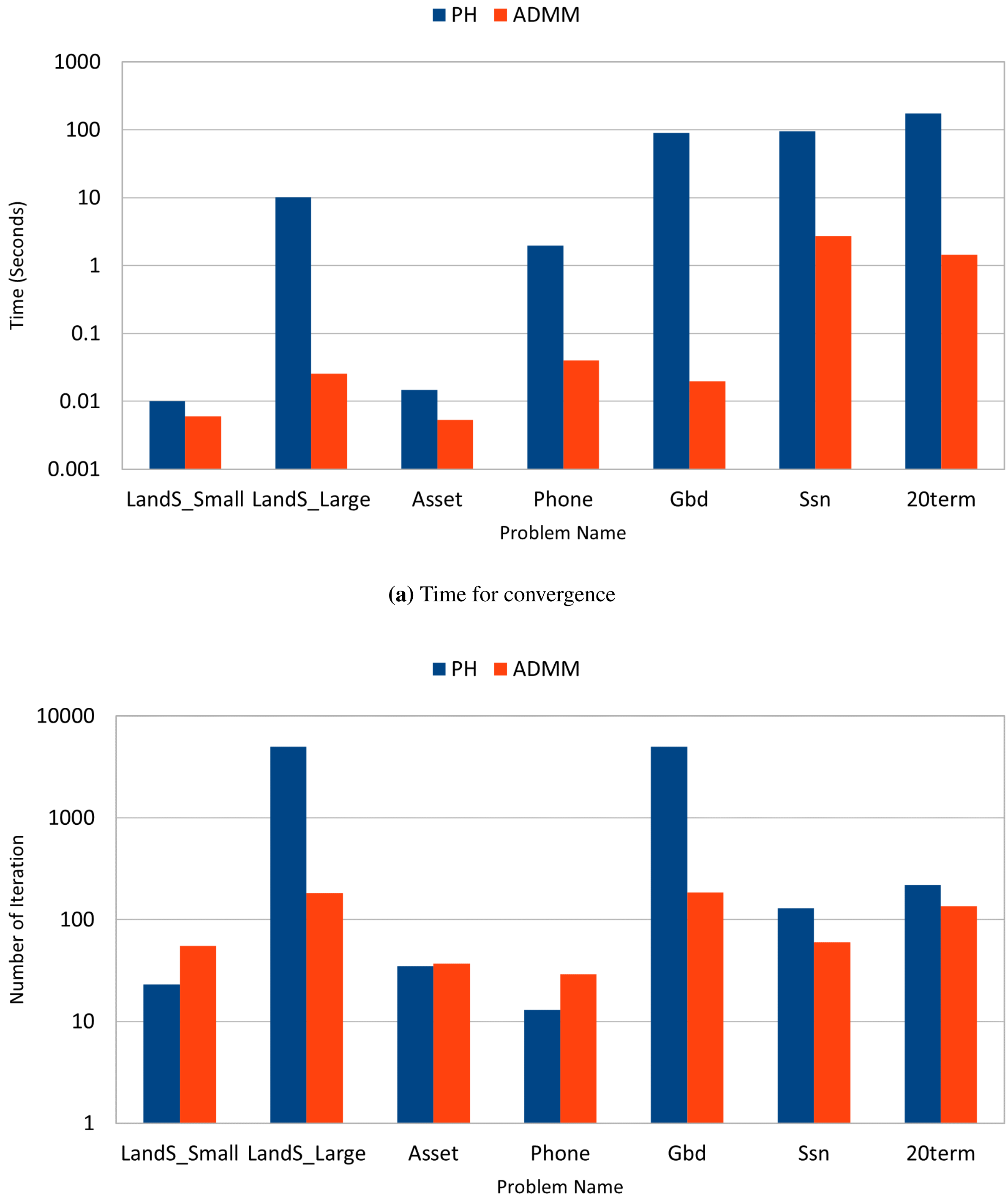
Problem	CPU Times (Seconds)		Number of Iterations		Termination		Percentage gap	
	PH	ADMM	PH	ADMM	PH	ADMM	PH	ADMM
LandS-Small	0.01	0.006	23	55	C	C	0.168%	0.047%
LandS-Large	10.1989	0.0254	5000	183	N	C	0.177%	0.13%
Asset	0.0148	0.0053	35	37	C	C	0.12%	0.09%
Phone	1.95	0.04	13	29	C	C	0.29%	0.001%
Gbd	89.56	0.0197	5000	184	N	C	0.27%	0.52%
Ssn	95.4	2.7476	130	60	C	C	0.625%	0.46%
20term	174.8	1.445499	220	136	C	C	1.655%	0.017%

CPU time and the number of iterations in Table 8.4 and Figure 8.9. Here, we call Algorithm 2 as ADMM for simplicity.

To assess the performance of ADMM and PH algorithms in terms of computational time, we have run both algorithms using the same programming language (MATLAB) using the same computer device. A direct comparison of the running time of the two methods is unfair due to the different meanings of the stopping criteria used by the algorithms. For this reason, we have used the following methodology: the two algorithms stop when the error (a combination of primal and dual residuals) is less than convergence tolerance $\epsilon = 10^{-3}$ or when they reach the maximum number of iterations.

In Table 8.3, columns 2-5 present the results of the two algorithms. Columns 2 and 3 contain the CPU times required for convergence; columns 4 and 5 present the number of iterations. We have used the same maximum iteration number for both algorithms to be 5000. The symbols N and C in columns 6 and 7 denote non-convergence and convergence cases, respectively. The percentage gaps, columns 8 and 9, are calculated using the formula $\left|\frac{\phi - z^*}{z^*}\right| \times 100$, where ϕ is the computed optimal value and z^* the known optimal.

Table 8.3 shows that ADMM converged in all the presented problems while PH failed to converge for two problems; ADMM gives excellent approximations to the optimal values. The percentage gap of ADMM is smaller than PH, which infers that the convergence of ADMM is superior to PH.



(a) Time for convergence

(b) Number of iterations

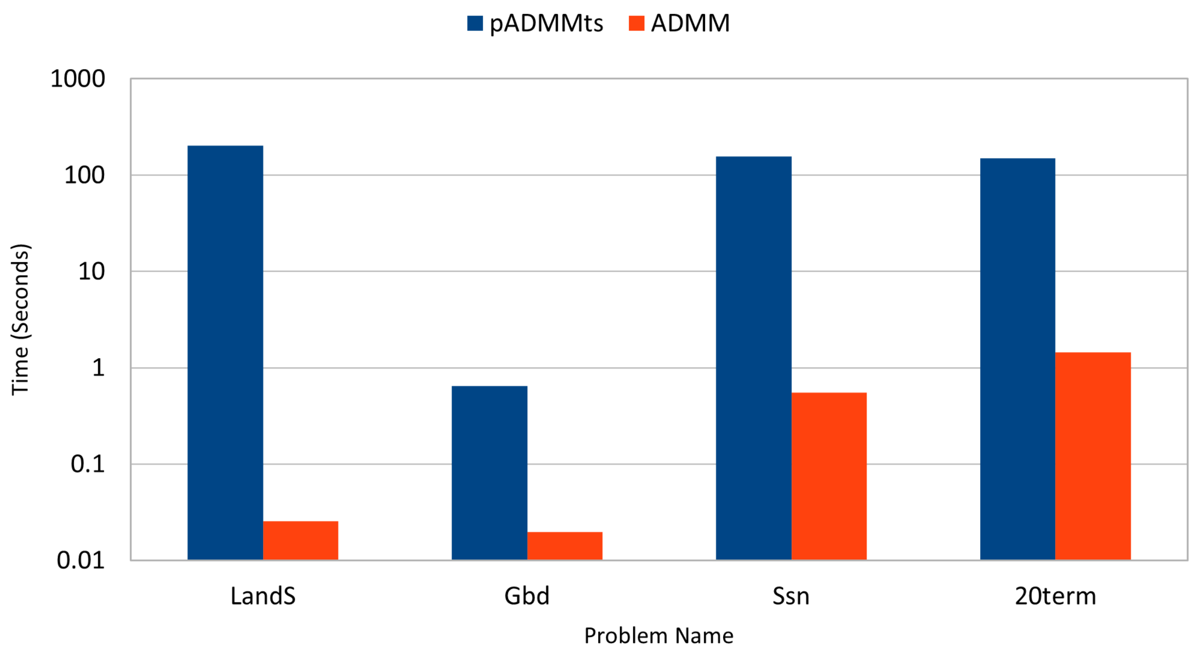
Figure 8.8: Comparison Between PH and ADMM Using Seven Problems from Table 8.1

The bars in Figure 8.8 are on a logarithmic scale because of the variations between the number of iterations and convergence time results of different problems. Figure 8.8a presents the time that algorithms require to stop. We observe that ADMM converges faster than PH because of the parallel implementation of ADMM. Figure 8.8b shows the number of iterations demonstrating that ADMM outperforms PH on four problems. On the other hand, PH has less number of iterations on the other two problems. Both methods are comparable in the number of iterations of the Asset problem. Even though it seems that PH has a less number of iterations on two problems, it takes a longer time to converge.

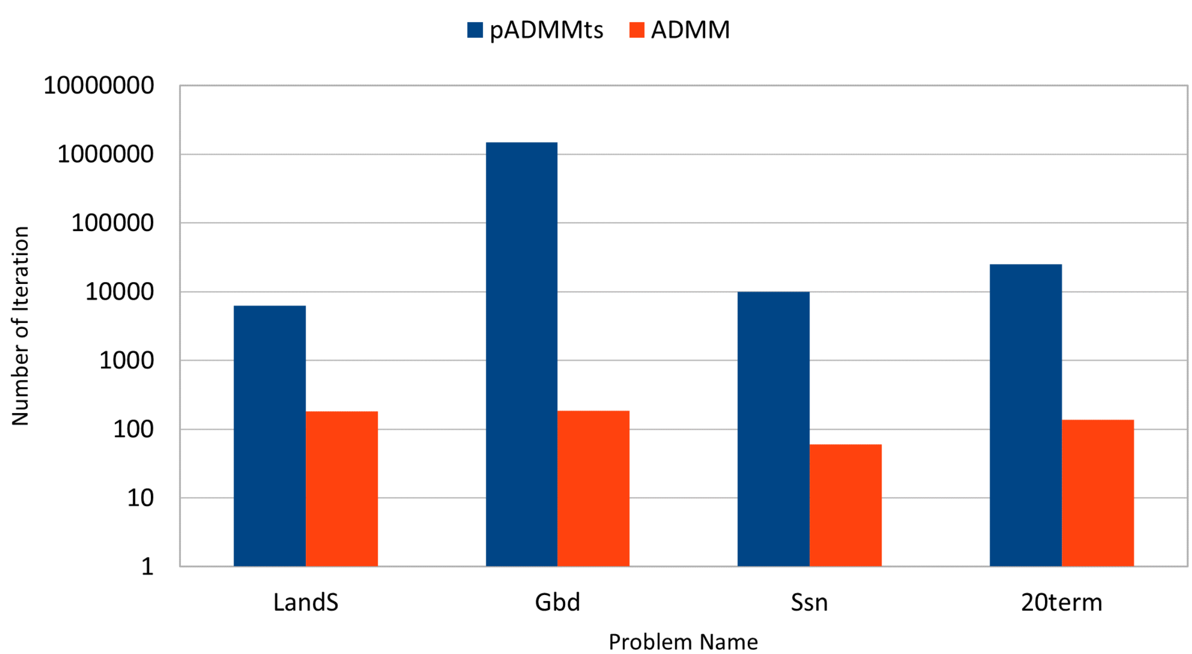
Now we assess the performance of ADMM and pADMMts in terms of computational time and the number of iterations using the same four problems used by Arpón et al. [81]. Both algorithms use the same concept of primal and dual residuals. For the fairness of comparison, we have used the threshold specified by Arpón et al. for the stopping criteria, which is $\epsilon = 10^{-3}$.

The graphs in Figure 8.9 are also on a logarithmic scale. Figure 8.9 and Table 8.4 present the numerical comparison between the two methods. Figure 8.9a shows the convergence time, and Figure 8.9b displays the number of iterations. We observe that ADMM converges faster than pADMMts with a smaller number of iterations in all the presented problems. Both algorithms attain the optimal objective value for each problem by convergence, but pADMMts take a significant number of iterations to converge.

In addition to showcasing the superior performance of the three-block ADMM over the PH method, this research carried significant implications for hybrid system sizing. The robustness and efficiency demonstrated by the three-block ADMM in solving optimization problems are also leveraged, in Chapter 6, to optimize the sizing of hybrid renewable energy systems integrated with energy storage. By utilizing ADMM's ability to handle complex and interconnected variables, system designers can more accurately determine the optimal sizing of different components within hybrid systems, considering cost, reliability, storage capacity, and load demand. The results of this study highlight the potential of ADMM as a valuable tool for supporting decision-making



(a) Time for convergence



(b) Number of iterations

Figure 8.9: Comparison of pADMMts with ADMM Using Four Problems from Table 8.1.

Table 8.4: Result Summary of the Comparison Between pADMMts and ADMM Methods

Problem	Scenarios	CPU Times (seconds)		# of Iterations	
		pADMMts	ADMM	pADMMts	ADMM
LandS	500	201	0.0254	6.3×10^3	183
Gbd	500	0.65	0.0197	1500×10^3	184
Ssn	100	155	0.54952	10×10^3	60
20term	100	149	1.445499	25×10^3	136

processes in hybrid system sizing, enabling the design of more efficient and cost-effective energy solutions.

Chapter 9

Conclusion

We have developed a three-block ADMM method to solve two-stage stochastic programming (SP) problems with finite scenarios. We have implemented the proposed algorithm to obtain the optimal hybrid renewable energy system using stochastic data derived from fluctuating load demands. The unpredictability of load demands is determined by the consumers' needs, which vary from one season to another and change during the day. Also, we have applied the model to seven benchmark problems from the literature, and we found that it always converges to the optimal solution reported for each problem. Our proposed three-block ADMM method has a parallel implementation, which provides faster computational time for large-sized problems.

We have proved that the proposed three-block ADMM algorithm always converges to the optimal solution with faster CPU time. We have established the theoretical convergence of the iterative three-block ADMM to the optimal solution of the two-stage stochastic programming problem. We have then implemented the method to the three-block formulation of the problem where its decomposable structure has been fully exploited in solving large-size problems. Numerical investigations of the algorithm using seven benchmark problems have justified the theoretical convergence of the proposed algorithm.

We have shown that the adaptive penalty parameter for the augmented lagrangian generates an accurate solution and always leads to convergence regardless of the initial choice. Also, we have compared the performance of three-block ADMM with the state-of-art Progressive Hedging (PH) algorithm using the probability of success in obtaining the optimal solution, the accuracy of the solution obtained, and the CPU times. The numerical experiments have shown that the three-block ADMM achieves a better convergence time than PH for all tested problems, but it might take more iterations for some problems. On the other hand, PH takes longer times but less number of iterations. It is because it calls the Linear Programming (LP) solver at each iteration, which requires more time to handle large-size problems. Also, we have compared ADMM and pADMMts in terms of the number of iterations and the CPU times. The comparison has shown that ADMM outperforms pADMMts.

We have designed an optimal sizing model for a proposed hybrid renewable energy system that will last for 20 years. Since we aim to reduce the environmental impact caused by conventional power generators, the proposed model is entirely renewable that uses solar, wind, hydrogen, and batteries. We have chosen the rural area Ga-Nkoane, Limpopo, South Africa, as a case study to apply our proposed model. The proposed algorithm blends all the available information to find the optimal configuration of renewable sources. It allows the user to specify the minimum and the maximum number of renewable plants. Since renewable systems depend solely on weather conditions, which might affect the system's reliability, we used the LPSP to regulate the reliability percentage. The LPSP controls the portion of power outages in the system.

The reliability of an energy system is strongly related to the system's energy storage capacity. In finding the optimal sizing of an energy system, it is possible to assign a higher weight to the reliability factor. When doing so, the optimization algorithm will prioritize the fulfillment of having a more reliable system. As a result, the algorithm will naturally adjust the sizing of the energy storage and renewable energy production components, ensuring they are adequately scaled to meet reliability requirements. This approach guarantees that the energy system can effectively handle variations in renewable energy generation and fluctuations in energy demand, ultimately leading to increased overall reliability. Therefore, in the optimization process, giving greater weight to reliability automatically drives the system towards larger storage and renewable production sizes, solidifying the reliability of the energy system.

Also, we have compared the optimal results produced by the ADMM to the Monte Carlo algorithms in Sakalauskas and Zilinskas's study [5]. We have found that the results are comparable, which proves the correctness of the algorithm alongside the mathematical convergence proof.

This study is beneficial in setting up a hybrid renewable energy system in areas with similar load demand to Ga-Nkoane. The proposed sizing algorithm is implementable in any location where load demand and weather data are available. It is important to note that the model's comparison and verification process primarily relied on a single real hybrid renewable system. However, it has also been validated using various benchmark problems from the literature. An intriguing direction for further research would involve analyzing the application of this algorithm to renewable energy systems with grid-connected power and conventional methods.

The previous approach of designing the lifetime project for the optimal configuration of the hybrid renewable energy system using three-block ADMM has the issue of unused components. That is because the entire system is built at the beginning of the project. We have proposed a new novel optimization approach to overcome this issue using a differential equations-based model. This approach solves the problem by determining the optimal cost and size of the renewable components annually based on the current demand. This approach makes it possible to build only the necessary components to meet the current demand reducing the cost and making the system more efficient. Furthermore, the total budget for the whole planned duration is not required to be available at the beginning of the project. The new procedure is more practical and feasible, considering the dynamic changes in demand and population growth. On the other hand, it is more costly to build the entire system upfront, which may not be fully used at the beginning of the project.

The results show that the optimal size and cost of the plants decrease year over year, reflecting the decision to expand new power plants based on the gradually expanding system. The differential equations-based model is superior in terms of capital cost when compared to the three-block ADMM approach. In our numerical experiments, we have found the differential equations-based model has decreased the total capital cost for the renewable components by 7.8%. It could be a significant deciding factor in choosing which method to use for long-term projects. Moreover, given that we conducted the comparison using only one real example, it is natural to extend the comparison to encompass multiple cases, considering various scenarios and variables that could influence the outcomes.

Furthermore, the differential equation-based model incorporates the cumulative PV area over three-year intervals. As a result, it generates a lower PV output during the initial two years compared to the ADMM approach. This is because the differential equation-based model optimally decides the necessary components gradually, based on the load demand, rather than sizing the entire project lifespan all at once.

One limitation of the differential equations-based model is solved using a pattern search optimization algorithm. This algorithm proves beneficial as it does not rely on derivatives for optimization, making it suitable for scenarios where derivative information is unavailable or difficult to compute. However, it is important to note that as the size of the decision variables increases, the search space of the algorithm also expands accordingly. Consequently, the algorithm may require more time to converge due to the increased complexity of exploring a larger solution space.

Also, we modeled the power curve and solar power prediction using machine learning procedures. We have used time-series data in the analysis of different models. We found that linear models produce poor results, which makes them unreliable for these predictions. The ensembles, LSTM, and regression trees show more accurate results than the parametric models considered. We have found that the LSTM is the best model for the power curve and solar power predictions among the tested procedures. This analysis provides insights for applying machine learning in time series forecasting and provides accurate models for predicting wind turbine power curves and solar power output. The machine learning models used for time-series data for the wind turbine power curve and solar power output are applicable to other data elsewhere in the world.

Our future research will focus on integrating the ADMM with multi-stage stochastic programming (MSP) to solve complex optimization problems. MSP is a powerful optimization method that models and solves problems with multiple stages with uncertain variables. However, MSP can be computationally intensive and may struggle to find the optimal solution in a reasonable amount of time, especially for large-scale problems. The application of ADMM to MSP problems aims to address these limitations by combining the strengths of both methods. ADMM is a fast and efficient optimization method that allows parallel computing, which can be used to solve large-scale problems, making it an ideal candidate for the optimization of MSP problems. Our future research will explore the use of ADMM to enhance the performance of MSP, particularly in terms of computation time and solution quality.

In addition, our future research will also focus on developing novel algorithms for solving MSP problems. These algorithms will be designed to consider the unique characteristics of MSP problems, such as the presence of multiple stages, uncertainty, and interstage dependencies. By developing these algorithms, we aim to make MSP more accessible and beneficial for solving complex optimization problems in various fields, including energy systems, finance, and transportation.

Overall, our future research on the application of ADMM to MSP problems promises to have a significant impact by providing a more efficient and effective approach for solving complex optimization problems with multiple stages and uncertain variables. Also, the advancements in integrating ADMM with MSP will have profound implications for hybrid renewable energy system sizing and optimization. The integration of these mathematical developments might offer a transformative approach to address the challenges faced in this domain. By leveraging the strengths of both ADMM and MSP, our future research aims to significantly improve the efficiency and effectiveness of hybrid renewable energy system optimization.

List of Submitted Papers

1. Mohammed, N., & Ali, M. M. (2022). Solving Two-stage Stochastic Programming Problems Using Alternating Direction Method of Multipliers. *Annals of Operations Research*. (IF: 4.82)

2. Mohammed, N., & Ali, M. M. (2022). Optimal Sizing of Hybrid Renewable Energy System Using Two-stage Stochastic Programming. *Renewable and Sustainable Energy Reviews*. (IF: 16.799)

3. Mohammed, N., & Ali, M. M. (2023). Differential Equations-Based Model In Sizing Hybrid Renewable Energy Systems. *IEEE TRANSACTIONS ON SUSTAINABLE ENERGY*. (IF: 8.31)

Bibliography

[1] Solar panel cost archives. https://www.solar.com/. Accessed: 2022-December-1.

[2] Lead acid vs lfp cost analysis: Cost per kwh battery storage. https://www.powertechsystems.eu/home/tech-corner/lithium-ion-vs-lead-acid-cost-analysis/. Accessed: 2022-December-1.

[3] How much do wind turbines cost? https://www.windustry.org/how_much_do_wind_turbines_cost. Accessed: 2022-December-1.

[4] Max Wei, Gregorio Levis, and Ahmad Mayyas. U.s. department of energy hydrogen program : Doe hydrogen program. https://www.hydrogen.energy.gov/pdfs/review20/fc332_wei_2020_o.pdf, Apr 2020.

[5] Leonidas Sakalauskas and Kestutis Žilinskas. Power plant investment planning by stochastic programming. *Technological and Economic Development of Economy*, 16(4):753–764, 2010.

[6] Jeff Linderoth, Alexander Shapiro, and Stephen Wright. The Empirical Behavior of Sampling Methods for Stochastic Programming. *Annals of Operations Research*, 142(1):215–241, 2006.

[7] S Solomon, D Qin, M Manning, Z Chen, M Marquis, K Averyt, M Tignor, and H Miller. Ipcc fourth assessment report (ar4). *Climate change*, 2007.

[8] South africa carbon (co2) emissions 1990-2023. https://www.macrotrends.net/countries/ZAF/south-africa/carbon-co2-emissions. Accessed: 2023-January-27.

[9] IEA PVPS. A snapshot of global pv (1992-2015). Technical report, tech. rep.(IEA Photovoltaic Power Systems Programme, 2016), 2016.

[10] John Coviello. Countries that generate 100% renewable energy electricity. https://turbofuture.com/industrial/Countries-That-Generate-100-Renewable-Energy-For-Their-Electricity-Needs, Aug 2019.

[11] Binayak Bhandari, Kyung-Tae Lee, Gil-Yong Lee, Young-Man Cho, and Sung-Hoon Ahn. Optimization of hybrid renewable energy power systems: A review. *International journal of precision engineering and manufacturing-green technology*, 2(1):99–112, 2015.

[12] Raul Banos, Francisco Manzano-Agugliaro, FG Montoya, Consolacion Gil, Alfredo Alcayde, and Julio Gómez. Optimization methods applied to renewable and sustainable energy: A review. *Renewable and sustainable energy reviews*, 15(4):1753–1766, 2011.

[13] Riad Chedid and Saifur Rahman. Unit sizing and control of hybrid wind-solar power systems. *IEEE Transactions on energy conversion*, 12(1):79–85, 1997.

[14] R Chedid, S Karaki, and A Rifai. A multi-objective design methodology for hybrid renewable energy systems. In *2005 IEEE Russia Power Tech*, pages 1–6. IEEE, 2005.

[15] AR De and L Musgrove. The optimization of hybrid energy conversion systems using the dynamic programming model—rapsody. *International Journal of Energy Research*, 12(3):447–457, 1988.

[16] Hongxing Yang, Wei Zhou, Lin Lu, and Zhaohong Fang. Optimal sizing method for stand-alone hybrid solar–wind system with lpsp technology by using genetic algorithm. *Solar energy*, 82(4):354–367, 2008.

[17] Hongxing Yang, Zhou Wei, and Lou Chengzhi. Optimal design and techno-economic analysis of a hybrid solar–wind power generation system. *Applied Energy*, 86(2):163–169, 2009.

[18] Eftichios Koutroulis, Dionissia Kolokotsa, Antonis Potirakis, and Kostas Kalaitzakis. Methodology for optimal sizing of stand-alone photovoltaic/wind-generator systems using genetic algorithms. *Solar energy*, 80(9):1072–1088, 2006.

[19] M Mohammadi, SH Hosseinian, and GB Gharehpetian. Ga-based optimal sizing of microgrid and dg units under pool and hybrid electricity markets. *International Journal of Electrical Power & Energy Systems*, 35(1):83–92, 2012.

[20] Daming Xu, Longyun Kang, Liuchen Chang, and Binggang Cao. Optimal sizing of standalone hybrid wind/pv power systems using genetic algorithms. In *Canadian Conference on Electrical and Computer Engineering, 2005.*, pages 1722–1725. IEEE, 2005.

[21] SM Hakimi and SM Moghaddas-Tafreshi. Optimal sizing of a stand-alone hybrid power system via particle swarm optimization for kahnouj area in south-east of iran. *Renewable energy*, 34(7):1855–1862, 2009.

[22] U Boonbumroong, N Pratinthong, S Thepa, C Jivacate, and W Pridasawas. Particle swarm optimization for ac-coupling stand alone hybrid power systems. *Solar Energy*, 85(3):560–569, 2011.

[23] M Mohammadi, SH Hosseinian, and GB Gharehpetian. Optimization of hybrid solar energy sources/wind turbine systems integrated to utility grids as microgrid (mg) under pool/bilateral/hybrid electricity market using pso. *Solar energy*, 86(1):112–125, 2012.

[24] A Kashefi Kaviani, GH Riahy, and SH M Kouhsari. Optimal design of a reliable hydrogen-based stand-alone wind/pv generating system, considering component outages. *Renewable energy*, 34(11):2380–2390, 2009.

[25] Giuseppe La Terra, Gagliano Salvina, and GM Tina. Optimal sizing procedure for hybrid solar wind power systems by fuzzy logic. In *MELECON 2006-2006 IEEE Mediterranean Electrotechnical Conference*, pages 865–868. IEEE, 2006.

[26] Tamer Khatib, Azah Mohamed, and K Sopian. Optimization of a pv/wind micro-grid for rural housing electrification using a hybrid iterative/genetic algorithm: Case study of kuala terengganu, malaysia. *Energy and buildings*, 47:321–331, 2012.

[27] Hongxing Yang, Lin Lu, and Wei Zhou. A novel optimization sizing model for hybrid solar-wind power generation system. *Solar energy*, 81(1):76–84, 2007.

[28] Seth B Darling, Fengqi You, Thomas Veselka, and Alfonso Velosa. Assumptions and the levelized cost of energy for photovoltaics. *Energy & environmental science*, 4(9):3133–3139, 2011.

[29] Bahtiyar Dursun and Ercan Aykut. An investigation on wind/pv/fuel cell/battery hybrid renewable energy system for nursing home in istanbul. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 233(5):616–625, 2019.

[30] Binayak Bhandari, Kyung-Tae Lee, Caroline Sunyong Lee, Chul-Ki Song, Ramesh K Maskey, and Sung-Hoon Ahn. A novel off-grid hybrid power system comprised of solar photovoltaic, wind, and hydro energy sources. *Applied Energy*, 133:236–242, 2014.

[31] M Fadaeenejad, MAM Radzi, MZA AbKadir, and H Hizam. Assessment of hybrid renewable power sources for rural electrification in malaysia. *Renewable and Sustainable Energy Reviews*, 30:299–305, 2014.

[32] Christopher Osita Anyaeche, Toyin Akappo, and Adefemi Omowole Adeodu. Optimisation of hybrid energy system production parameters for electricity power generation in nigeria. *Energy and Power Engineering*, 10(05):198, 2018.

[33] Anis Afzal, Mohibullah Mohibullah, and Virendra Kumar Sharma. Optimal hybrid renewable energy systems for energy security: a comparative study. *International Journal of Sustainable Energy*, 29(1):48–58, 2010.

[34] Gilbert M Masters. *Renewable and efficient electric power systems*. John Wiley & Sons, 2013.

[35] Jun Dong, Zhenhai Dou, Shuqian Si, Zichen Wang, and Lianxin Liu. Optimization of capacity configuration of wind–solar–diesel–storage using improved sparrow search algorithm. *Journal of Electrical Engineering & Technology*, 17(1):1–14, 2022.

[36] D Saheb Koussa, M Koussa, and S Hadji. Assessment of various wtg (wind turbine generators) production in different algerian's climatic zones. *Energy*, 96:449–460, 2016.

[37] Adel Brka. Optimisation of stand-alone hydrogen-based renewable energy systems using intelligent techniques. *Edith Cowan University*, 2015.

[38] Paweł Daszkiewicz, Beata Kurc, Marita Pigłowska, and Maciej Andrzejewski. Fuel cells based on natural polysaccharides for rail vehicle application. *Energies*, 14(4):1144, 2021.

[39] Peter P Edwards, Vladimir L Kuznetsov, William IF David, and Nigel P Brandon. Hydrogen and fuel cells: towards a sustainable energy future. *Energy policy*, 36(12):4356–4362, 2008.

[40] Krystina Lamb, Michael Dolan, and Danielle Kennedy. Ammonia for hydrogen storage; a review of catalytic ammonia decomposition and hydrogen separation and purification. *International Journal of Hydrogen Energy*, 44, 02 2019.

[41] Weiqiang Dong, Yanjun Li, and Ji Xiang. Optimal sizing of a stand-alone hybrid power system based on battery/hydrogen with an improved ant colony optimization. *Energies*, 9(10):785, 2016.

[42] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[43] Natashia Boland, Jeffrey Christiansen, Brian Dandurand, Andrew Eberhard, Jeff Linderoth, James Luedtke, and Fabricio Oliveira. Combining progressive hedging with a frank–wolfe method to compute lagrangian dual bounds in stochastic mixed-integer programming. *SIAM Journal on Optimization*, 28(2):1312–1336, 2018.

[44] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1):1–25, 1997.

[45] Charles Audet and John E Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on optimization*, 13(3):889–903, 2002.

[46] Pedro Alberto, Fernando Nogueira, Humberto Rocha, and Luís N Vicente. Pattern search methods for user-provided points: Application to molecular geometry problems. *SIAM Journal on Optimization*, 14(4):1216–1236, 2004.

[47] Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003.

[48] Musa Nur Gabere. *Prediction of antimicrobial peptides using hyperparameter optimized support vector machines*. PhD thesis, University of the Western Cape, 2011.

[49] Alhassan Ali Teyabeen, Fathi Rajab Akkari, and Ali Elseddig Jwaid. Mathematical modelling of wind turbine power curve. *International Journal of Simulation–Systems, Science & Technology*, 19(5), 2018.

[50] Vaishali Sohoni, SC Gupta, and RK Nema. A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. *Journal of Energy*, 2016, 2016.

[51] Rich H Inman, Hugo TC Pedro, and Carlos FM Coimbra. Solar forecasting methods for renewable energy integration. *Progress in energy and combustion science*, 39(6):535–576, 2013.

[52] Tao Hong, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J Hyndman. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond, 2016.

[53] Peder Bacher, Henrik Madsen, and Henrik Aalborg Nielsen. Online short-term solar power forecasting. *Solar energy*, 83(10):1772–1783, 2009.

[54] Renani Ehsan Taslimi. *Power prediction using the wind turbine power curve and data-driven approaches/Ehsan Taslimi Renani*. PhD thesis, University of Malaya, 2018.

[55] M Lydia, A Immanuel Selvakumar, S Suresh Kumar, and G Edwin Prem Kumar. Advanced algorithms for wind turbine power curve modeling. *IEEE Transactions on sustainable energy*, 4(3):827–835, 2013.

[56] Mohammed G Khalfallah and Aboelyazied M Koliub. Suggestions for improving wind turbines power curves. *Desalination*, 209(1-3):221–229, 2007.

[57] Vinay Thapar, Gayatri Agnihotri, and Vinod Krishna Sethi. Critical analysis of methods for mathematical modelling of wind turbines. *Renewable Energy*, 36(11):3166–3177, 2011.

[58] C Carrillo, AF Obando Montaño, J Cidrás, and Eloy Díaz-Dorado. Review of power curve modelling for wind turbines. *Renewable and Sustainable Energy Reviews*, 21:572–581, 2013.

[59] Andrew Kusiak, Haiyang Zheng, and Zhe Song. On-line monitoring of power curves. *Renewable Energy*, 34(6):1487–1493, 2009.

[60] Andrew Kusiak, Haiyang Zheng, and Zhe Song. Models for monitoring wind farm power. *Renewable Energy*, 34(3):583–590, 2009.

[61] Wind atlas for south africa (wasa) sanedi. http://www.wasaproject.info/. Accessed: 2022-August-12.

[62] V90-2.0 mw™. https://www.vestas.com/en/products/2-mw-platform/V90-2-0-MW. Accessed: 2023-February-06.

[63] Tenesol te220 datasheet. https://www.rectifier.co.za/tenesol/SPECS/TE220_240-60P+%20en.pdf. Accessed: 2023-January-03.

[64] Supervised learning. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning. Accessed: 2023-March-07.

[65] Keras Team. Keras documentation: Lstm layer. https://keras.io/api/layers/recurrent_layers/lstm/. Accessed: 2023-March-06.

[66] Cydal. Cydal/tsextract: Time series feature extraction for supervised learning modeling. https://github.com/cydal/tsExtract. Accessed: 2023-March-06.

[67] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[68] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[69] Rui Guo, Zhiqian Zhao, Tao Wang, Guangheng Liu, Jingyi Zhao, and Dianrong Gao. Degradation state recognition of piston pump based on iceemdan and xgboost. *Applied Sciences*, 10(18):6593, 2020.

[70] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[71] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

[72] Jing Liu, Yongrui Duan, and Min Sun. A symmetric Version of the Generalized Alternating Direction Method of Multipliers for Two-block Separable Convex Programming. *Journal of Inequalities and Applications*, 2017(1):1–21, 2017.

[73] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed Optimization and Statistical Learning via The Alternating Direction Method of Multipliers*. Now Publishers Inc, 2011.

[74] Z. Wang, B. Hall, J. Xu, and X. Shi. A Sparse Learning Framework for Joint Effect Analysis of Copy Number Variants. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(5):1013–1027, 2017.

[75] J. Wang, Z. Chai, Y. Cheng, and L. Zhao. Toward Model Parallelism for Deep Neural Network Based on Gradient-Free ADMM Framework. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 591–600, 2020.

[76] Daniel Gabay and Bertrand Mercier. A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[77] Roland Glowinski and A Marroco. Sur L'approximation, Par Éléments Finis D'ordre Un, Et La Résolution, Par Pénalisation-dualité D'une Classe de Problèmes de Dirichlet Non Linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.

[78] Defeng Sun, Kim-Chuan Toh, and Liuqin Yang. A Convergent 3-Block Semiproximal Alternating Direction Method of Multipliers for Conic Programming with 4-Type Constraints. *SIAM journal on Optimization*, 25(2):882–915, 2015.

[79] KA Ariyawansa and Andrew J Felt. On a New Collection of Stochastic Linear Programming Test Problems. *INFORMS Journal on Computing*, 16(3):291–299, 2004.

[80] R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and Policy Aggregation in Optimization Under Uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[81] Sebastián Arpón, Tito Homem-de Mello, and Bernardo K Pagnoncelli. An ADMM Algorithm for Two-stage Stochastic Programming Problems. *Annals of Operations Research*, 286(1):559–582, 2020.

[82] Fenghui Wang, Wenfei Cao, and Zongben Xu. Convergence of Multi-block Bregman ADMM for Nonconvex Composite Problems. *Science China Information Sciences*, 61(12):122101, 2018.

[83] Andrzej Ruszczyński. Decomposition methods in stochastic programming. *Mathematical programming*, 79(1):333–353, 1997.

[84] Jacek Gondzio, Pablo González-Brevis, and Pedro Munari. Large-scale Optimization with The Primal-dual Column Generation Method, 2016.

[85] R Tyrrell Rockafellar. *Convex analysis*, volume 18. Princeton university press, 1970.

[86] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[87] Sarah M Ryan, Roger J-B Wets, David L Woodruff, César Silva-Monroy, and Jean-Paul Watson. Toward Scalable, Parallel Progressive Hedging for Stochastic Unit Commitment. In *2013 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2013.

[88] BS He, Hai Yang, and SL Wang. Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities. *Journal of Optimization Theory and Applications*, 106(2):337–356, 2000.

[89] R Tyrrell Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[90] Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[91] Zheng Xu, Mario Figueiredo, and Tom Goldstein. Adaptive ADMM with Spectral Penalty Parameter Selection. In *Artificial Intelligence and Statistics*, pages 718–727. PMLR, 2017.

[92] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal Alternating Minimization and Projection Methods for Nonconvex Problems: An Approach Based on The Kurdyka-Łojasiewicz Inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[93] Hussain Ali Hatam Alshamri. *Optimal sizing of hybrid renewable systems to improve electricity supply reliability in Iraqi domestic dwellings*. PhD thesis, University of Leeds, 2019.

[94] Zhe Zhou, Jianyun Zhang, Pei Liu, Zheng Li, Michael C Georgiadis, and Efstratios N Pistikopoulos. A two-stage stochastic programming model for the optimal design of distributed energy systems. *Applied Energy*, 103:135–144, 2013.

[95] Fu-Cheng Wang, Yi-Shao Hsiao, and Yi-Zhe Yang. The optimization of hybrid power systems with renewable energy and hydrogen generation. *Energies*, 11(8):1948, 2018.

[96] Anastasios Tsoularis and James Wallace. Analysis of logistic growth models. *Mathematical biosciences*, 179(1):21–55, 2002.

[97] Data first. https://www.datafirst.uct.ac.za/. Accessed: 2022-March-16.

[98] Power project data sets. https://power.larc.nasa.gov/. Accessed: 2022-March-16.

[99] Robert M Freund. Optimization under uncertainty. *Massachusetts Institute of Technology*, pages 18–27, 2004.

[100] A. Gersho. Quantization. *IEEE Communications Society Magazine*, 15(5):16–16, 1977.

[101] Charles Audet and John E Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006.

[102] MF AlHajri, KM El-Naggar, MR AlRashidi, and AK Al-Othman. Optimal extraction of solar cell parameters using pattern search. *Renewable energy*, 44:238–245, 2012.

[103] LV Suresh Kumar, GV Nagesh Kumar, and Sreedhar Madichetty. Pattern search algorithm based automatic online parameter estimation for agc with effects of wind power. *International Journal of Electrical Power & Energy Systems*, 84:135–142, 2017.

[104] Pysmps. https://pypi.org/project/pysmps/. Accessed: 2023-January-27.

[105] François V Louveaux and Yves Smeers. Optimal Investments for Electricity Generateion: A Stochastic Model and A Test Problem. *Numerical Techniques for Stochastic Optimization*, 1988.

[106] Andy Felt. Test-Problem Collection for Stochastic Linear Programming. https://www4.uwsp.edu/math/afelt/slptestset/download.html, 2003. [Online; accessed 14-October-2020].

[107] George B Dantzig and Mukund N Thapa. *Linear Programming 2: Theory and Extensions*. Springer Science & Business Media, 2006.

[108] Suvrajeet Sen, Robert D Doverspike, and Steve Cosares. Network planning with random demand. *Telecommunication systems*, 3(1):11–30, 1994.

[109] Wai-Kei Mak, David P Morton, and R Kevin Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1-2):47–56, 1999.