©Copyright 2024
William Yang

# Robust Optimization Methods for Improving Virtual Power Plant Reliability and Classification Fairness

William Yang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:

Chaoyue Zhao, Chair

Shuai Huang

Chiwei Yan

Program Authorized to Offer Degree:
Industrial and Systems Engineering

University of Washington

**Abstract**

Robust Optimization Methods for Improving Virtual Power Plant Reliability and Classification Fairness

William Yang

Chair of the Supervisory Committee:
Chaoyue Zhao
Industrial and Systems Engineering

In this dissertation, we present three different robust optimization approaches: distributionally robust optimization (DRO), target-oriented robust optimization (TORO), and Rawlsian fairness. We apply DRO and TORO frameworks to address virtual power plant reliability, and Rawlsian fairness to address the fair binary classification problem.

A virtual power plant (VPP) is an entity that aggregates smaller solar and wind farms with other heterogeneous distributed energy resources (DERs) to increase their visibility to Independent System Operators (ISOs) and allows them to participate in the energy market. Typically, smaller solar and wind farms are unable to participate in the wholesale energy market, so a VPP's ability to integrate them into the energy market is a crucial step for reducing the global carbon footprint and combating climate change. Our proposed multi-stage DRO framework allows us to schedule VPP operations in the presence of intermittent renewable energy output by dynamically coordinating the heterogeneous DERs in a reliable and cost-effective manner. Our proposed TORO method helps us address another challenge that arises from increased renewable energy penetration, which is the assessment of the VPP's flexibility. The uncertainty brought by renewable energy makes it harder to balance energy supply and demand, and failing to do so can result in expensive renewable energy curtailment or blackouts. Our TORO method provides a flexibility assessment framework that identifies the maximum amount of net load deviation the system can tolerate.

Traditional binary classification algorithms are prone to producing unfair results that favor certain demographic groups over others. This inequity is often exacerbated in unbalanced datasets where the number of entries from a majority group significantly outweighs the entries from a minority group. We use a MIP framework to formulate our Rawlsian fairness to address these inequities.

Our methodology prioritizes the performance of the worst-off demographic group, and our specific formulation can produce interpretable solutions by directly optimizing sparsity. Additionally, it provides flexibility for users to achieve interpretable solutions in multiple ways.

# TABLE OF CONTENTS

Page

List of Figures . . . . . iii

List of Tables . . . . . iv

Chapter 1: Introduction . . . . . 1
1.1 Motivation . . . . . 1
1.2 Research Objectives . . . . . 3
1.3 Organization of the Dissertation . . . . . 4

Chapter 2: A Real-time Distributionally Robust Approach Towards Improving Cost-effectiveness and Reliability of Virtual Power Plant Scheduling . . . . . 5
2.1 Background . . . . . 5
2.2 VPP Model Formulation . . . . . 7
2.3 Methodology . . . . . 11
2.4 Experimental Results . . . . . 17
2.5 Conclusion . . . . . 29

Chapter 3: A Two-Stage Target-Oriented Robust Optimization Model to Assess the Flexibility of Virtual Power Plants with Quick-Start Storage Resources . . . . . 30
3.1 Background . . . . . 30
3.2 Flexibility Model Formulation . . . . . 32
3.3 Abstract Formulation . . . . . 38
3.4 Model Reformuation . . . . . 39
3.5 Primal Generation Algorithm . . . . . 43
3.6 Experimental Results . . . . . 47
3.7 Conclusion . . . . . 54

Chapter 4: A Rawlsian Mixed Integer Programming Approach for Fair Classification . . 55
4.1 Background . . . . . 55
4.2 Model Formulation . . . . . 58
4.3 Experimental Results . . . . . 61
4.4 Conclusion . . . . . 70

Chapter 5: Conclusion . . . . . 72
5.1 Summary . . . . . 72

5.2 Extensions to Current Work . . . . . 72
5.3 Future Research Areas . . . . . 73

## LIST OF FIGURES


| Figure Number | | Page |
|---|---|---|
| 1.1 | VPP schematic | 2 |
| 2.1 | DRDP algorithm | 17 |
| 2.2 | Impact of changing the radius | 20 |
| 2.3 | Hourly cumulative cost | 22 |
| 2.4 | Battery usage for differing radius values | 22 |
| 2.5 | Cumulative DR and LS utilization | 23 |
| 2.6 | Static and dynamic DR and LS usage for $r = 70$ | 25 |
| 2.7 | Comparing cost when ESS or DR is removed | 26 |
| 2.8 | Average cost with different component combinations | 27 |
| 3.1 | Flexibility region | 34 |
| 3.2 | Flexibility region for different expected RNRG levels | 49 |
| 3.3 | Sensitivity analysis | 50 |
| 3.4 | Sensitivity analysis (cont.) | 51 |
| 4.1 | Fairness for different penalization values | 64 |
| 4.2 | Fairness for different upper-bound parameter values | 65 |

## LIST OF TABLES


Table Number Page

2.1 Distributed generator parameter values . . . . . 18
2.2 ESS parameters . . . . . 18
2.3 Demand response parameter values ($/kW) . . . . . 18
2.4 Parameters for scenario generation distribution . . . . . 19
2.5 Average cost for dynamic and static VPP scheduling ($) . . . . . 24
2.6 Relationship between horizon and runtime . . . . . 28
2.7 Increased profit as radius increases . . . . . 28

3.1 Distributed generator parameters (kW) . . . . . 48
3.2 ESS parameter values (kW) . . . . . 48
3.3 Demand response parameter values . . . . . 48
3.4 Runtime analysis (tolerance = 0.1%) . . . . . 52
3.5 Runtime analysis (tolerance = 1%) . . . . . 53
3.6 PGA total flexibility (kW) for two tolerances . . . . . 53

4.1 Model comparison for Penguin Dataset . . . . . 62
4.2 Error and fairness when $b = 5$ and $\lambda \in \mathbb{Z}^{P+1}$ . . . . . 66
4.3 Results from [1]. . . . . 68
4.4 Results from [2] . . . . . 69

## ACKNOWLEDGMENTS

I would like to express my sincerest thanks to the following people. Firstly, my PhD advisor, Professor Chaoyue Zhao. Professor Zhao is one of the brightest people in the world, and I am blessed to have had the opportunity to work with her. She helped me understand the beauty and intricacies of operations research and showed me what it takes to be a world-class scholar. She challenged and encouraged me, which helped me get to where I am today. But perhaps most importantly, she inspired me to be the best version of myself and instilled in me a sense of love and pride for my work. I will forever be grateful to have a mentor like her in my life.

To the rest of my dissertation committee, Professors Shuai Huang, Chiwei Yan, and Baosen Zhang, for providing feedback on my dissertation. Their guidance and expertise have helped me improve immensely as a researcher.

To my other research collaborators, Hongda Li, Lei Fan, and Jun Song, for sharing brilliant ideas with me and helping me see things from different perspectives. I learned a tremendous amount from working with them all.

To the rest of my professors at the University of Washington, Aleksandr Aravkin, Jeff Ban, Yen-Chi Chen, Youngjun Choe, Tekla Cunningham, Dmitriy Drusvyatskiy, Archis Ghate, Fang Han, Kevin Jamieson, Charles Lee, Shan Liu, Ron Patterson, Thomas Rothvoss, and Zelda Zabinksy, for providing me with the knowledge, skills, and tools I need to conduct research at a high level.

To my colleagues at the Clean Energy Institue, Hannah Contreras, Mercie Hodges, Danica Hendrickson, Xuetao Ma, Cody Schlenker, and Madelynne Zornes, who gave me a chance to present my research and provided me with valuable guidance for improving my presentation skills. To my colleagues at NREL, Seong Choi, Weijia Liu, Santosh Veda, and Hongming Zhang, for exposing me to exciting research projects, and helping me deepen my understanding of power grid operations. To my colleague at Lotus Engineering and Sustainability, Ally Mark, for sharing her expertise on emerging challenges in the world of sustainable energy, which provided me with valuable perspective for my research. To my colleague, Jonathan Hayase, who helped me overcome the non-trivial challenge of getting my code to run properly.

To my labmates, Ryan Lin, Prasanna Raut, Jun Song, and Xinyi Zhao, for being tremendous teammates who helped with my research and put me in a position to succeed. To the rest of my fellow UW ISE graduate students, Grace Douglas, Fiete Krutein, Monika Kwapisz, Serin Lee, Julia Lensing, Jiaxin Li, Pariyakorn Maneekul, Matt Martell, Danielle Morey, Joe Osborne, Wengeng Pan, David Prendez, Chris Salazar, Sameer Sapre, Ribhu Sengupta, Ameer Hamza Shakur, Mutita Siriruchatanon, Michelle Song, Xiaonan Sun, Nick Terry, Yinsheng Wang, and Yilun Xing, for giving me a sense of community at UW, and for providing companionship during my time in graduate school.

To the ISE staff, Kelly Foong, Sheila Prusa, Neelu Rajvanshi, Trevor Smith, and Jennifer Tsai, for their hard work in supporting me and making my time at ISE as smooth as possible.

To the undergraduate students in my optimization and stochastic processes classes who came into my lectures with great attitudes and helped me further my knowledge and love for operations research.

To my professors at the University of Michigan, Eunshin Byon, Xiuli Chao, Mark Daskin, Marina Epelman, Brian Denton, Jon Lee, Amy Cohn, Ruiwei Jiang, Mariel Lavieri, Siqian Shen, and Cong Shi, for introducing me to operations research and encouraging me to pursue a PhD. To professors at other universities, Geunyeong Byeon, Gian Garcia, Daniel Otero-Leon, Chenang Liu, Kayse Lee Maass, Minseok Ryu, Karmel Shehadeh, Lauren Steimle, Emily Tucker, Adam VanDeusen, Yiling Zhang, and Zheng Zhang, for providing mentorship and guidance during my PhD journey.

To Calvin Chen, Gitika Talwar, and Ashley and Geoff Van Dragt, for helping me believe in myself during times I was struggling to do so. To my friends Emily Ahn, Jeff Chan, Aaron Chang, Jesse Chiem, Bryan Chisholm, Chris Combs, Alex Fang, Mark Hwang, Eric Juang, Sam Li, Addison Lin, Issac Loo, Ebenezer Prasanna, Andrew Shiau, David Simmons, Josephine Sulimin, Justin Tu, Kevin Tully, Sean Van Hulle, Andrew Wagenmaker, Grace Wan, Chad and MC Westra, Diane Yoon, and Emmie Zhang who provided me with a healthy balance in life.

And last but not least, to my family, for supporting me on this journey. To my parents, Larry and Jenny, for giving me the chance to pursue my dream of getting a PhD and for helping me realize I could do it. To my wife, Amy, for being the center of my life and keeping me grounded. I would not be where I am today without her love during all of my ups and downs.

## DEDICATION

to my wife Amy, my anchor during the stormiest of seas

# Chapter 1
# INTRODUCTION

## *1.1 Motivation*

### *1.1.1 Robust Optimization*

In this dissertation, we explore the use of robust optimization methods in different important application areas. Robust optimization is a broad class of optimization methods where the objective is to minimize the maximum value of an objective function. In standard robust optimization problems, the objective is to minimize cost, while the random variables in the system achieve their worst-case values within some set of possible outcomes. We consider three different forms of robust problems: distributionally robust optimization (DRO), target-oriented robust optimization (TORO), and Rawlsian fairness. DRO aims to minimize cost while assuming that random variables follow the worst-case scenario of probability distribution. TORO aims to characterize the maximum range of variations that the system can tolerate. Rawlsian fairness aims to minimize the cost for the worst-off group within some discrete set of groups.

In this dissertation, we consider two applications: virtual power plant reliability and binary classification. We implement our DRO and TORO methods on the first application, and our Rawlsian fairness methodology on the second. We motivate these two applications below.

### *1.1.2 Virtual Power Plant*

Distributed energy resources (DERs), such as solar panels, batteries, and demand response units, can play vital roles in meeting the nation's increasing energy demand, backing up the electricity grid in the event of outages, and peak shaving in the case of high demand scenarios. Contrary to distributed generators, which lie on transmission systems and are dispatched by Independent System Operators (ISOs), DERs are spread across distribution systems. Individual DERs are typically too small to participate in the electricity market because ISOs do not have access to the DER's supply information and cannot effectively control their operation.

Traditionally, the approach of connecting DERs in most cases is based on a so-called "fit and forget" regime where DERs are used as backup resources and designed to fulfill demand in the

worst-case scenarios like peak load, which may only occur occasionally [3]. This conservative approach rules out the possibility of having an effective share of the energy supply from the DERs. Furthermore, DERs that rely on renewable energy, such as wind farms or photovoltaic cells, have a high degree of stochasticity and could fail to meet demand as individual entities [4, 5, 6].

A more reliable method for connecting DERs is the use of a Virtual Power Plant (VPP), which aggregates a heterogeneous portfolio of DERs and their capacities which allows them to participate in the wholesale energy market as a single entity (see, Figure 1.1). Instead of grouping all DERs with a particular program under one umbrella, the VPP concept allows utilities to aggregate these programs by type and location in the distribution topology, which allows for better forecasting and analysis of customer information to ultimately improve operational decision-making [7]. Another advantage of VPPs is their ability to aggregate massive flexible resources, which has posed significant challenges to the operation of active distribution networks [8]. Aggregating flexible resources helps improve power system stability and security, especially when the system experiences peak loads during extreme weather events [9]. Notably, the VPP's ability to facilitate the integration of smaller renewable energy resources into the energy market is a crucial step in increasing global renewable energy penetration and reducing carbon emissions [10] [11].

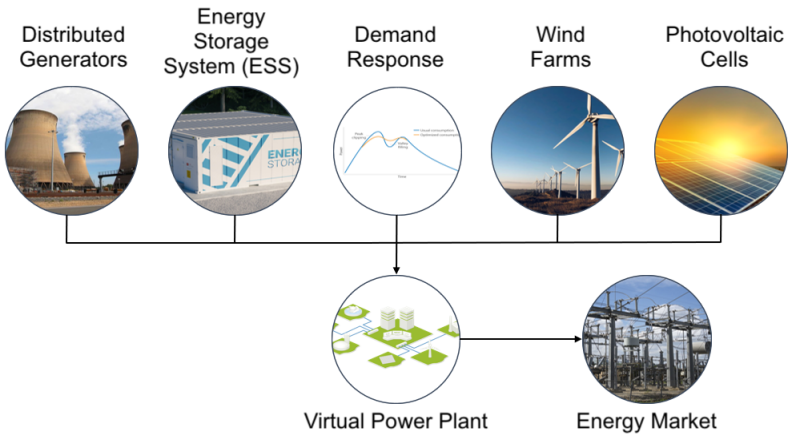



Figure 1.1: VPP schematic

Despite the VPP's advantages, incorporating renewable energy brings volatility into the system and makes it harder to reliably satisfy customer demand, leading to an increased blackout frequency over the past 20 years [12]. In a typical 24-hour period, customer demand is relatively steady, but the net load, which is defined as the difference between customer demand and renewable energy supply, can be volatile [13]. The volatile net load makes it harder for the VPP to adequately schedule its non-renewable generation resources. If the net load is lower than anticipated, this could lead to

expensive renewable energy curtailment. If it is higher than anticipated, this could lead to load-shedding or blackouts. Therefore, it is challenging to schedule and coordinate the VPP's DERs reliably and cost-effectively, and it is difficult for the VPP to assess its ability to reliably satisfy customer demand in the presence of volatile renewable energy resources.

### *1.1.3 Binary Classification*

Binary classification algorithms in machine learning categorize data into different classes and are widely used with a broad array of applications [14]. Despite many advances in the study of binary classification algorithms, this methodology still has some limitations. Namely, these algorithms are prone to producing unfair results in certain circumstances. A famous example of this is the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) algorithm [15], which was designed to label individuals as "low-risk" or "high-risk" for recidivism. The algorithm had a significantly higher rate of both false positives (labeling an individual as "high-risk" who did not recommit a crime) and false negatives (labeling an individual as "low-risk" who did recommit a crime) for black individuals compared to white individuals.

When an algorithm prioritizes the overall prediction accuracy across all individuals in the dataset, it could prioritize the prediction accuracy for members belonging to one demographic group over members in other groups as a result. Therefore, the accuracy for the worst-off demographic group could be dramatically worse than other groups in this scenario. Furthermore, this disparate treatment across demographic groups is commonly seen when members of one demographic group make up a large proportion of the dataset that the classification algorithm is used on [16].

## ***1.2 Research Objectives***

This dissertation explores the development and application of three different robust optimization methods:

1. Distributionally robust optimization, which provides a cost-effective and reliable method for VPP scheduling that facilitates the incorporation of smaller wind and solar farms into the energy market;

2. Target-oriented robust optimization, which provides a framework for the VPP to evaluate its ability to satisfy customer demand while incorporating the volatility of renewable energy;

3. Rawlsian fairness, which provides a flexible and interpretable methodology that makes data-driven classification assignments, while prioritizing the performance of the worst-off demographic group.

### *1.3 Organization of the Dissertation*

Chapter 2 addresses the first research objective by proposing a multi-stage distributionally robust method to provide an optimal scheduling policy for VPPs. This framework helps facilitate the inclusion of small renewable resources in the energy market in a way that is reliable and cost-effective. Chapter 3 addresses our second research objective by presenting a flexibility metric that is a comprehensive, reliable, and functional tool to help the power grid evaluate its ability to satisfy demand. We propose a Primal Generation Algorithm to solve our problem more efficiently than standard column-and-constraint-generation approaches. Chapter 4 addresses our third research objective by introducing a Rawlsian mixed-integer programming formulation for a classification algorithm that makes fair predictions. Our method prioritizes minimizing classification error for the worst-off group while providing several modeling advantages that enhance user interpretability and flexibility for users. Chapter 5 summarizes the dissertation, introduces possible extensions to the methods presented, and describes new research areas to explore in the future.

Chapter 2

# A REAL-TIME DISTRIBUTIONALLY ROBUST APPROACH TOWARDS IMPROVING COST-EFFECTIVENESS AND RELIABILITY OF VIRTUAL POWER PLANT SCHEDULING

## *2.1 Background*

VPPs have been extensively studied with respect to settings such as bidding strategy optimization [17], characteristic evaluation [18], and network security [19]. In this chapter, we tackle the problem of coordinating the VPP reliably and cost-effectively, which is known as the optimal scheduling problem for VPPs. Finding the optimal VPP scheduling strategy is a scientifically challenging question for the following three reasons. Firstly, since a VPP consists of various components of vastly different characteristics, coordinating their operation as one aggregated entity is difficult. Secondly, due to the fluid nature of power systems, there is a constant influx of available system information, and it is hard to smoothly incorporate such dynamics to inform real-time scheduling decisions. Lastly, it is challenging to coordinate a VPP in a way that mitigates risks due to a high degree of uncertainty brought by renewable energy, customer demand, and market prices.

There are various methods in the literature for integrating uncertainty in optimization models. One commonly used approach is stochastic optimization (SO) [20], which has been studied extensively for optimizing VPP scheduling [21, 22, 23, 24]. SO methods seek to maximize the expected profit or minimize the expected cost of a VPP over all random scenarios. One disadvantage of SO methods is that by optimizing the average performance, these models fail to consider catastrophically bad outcomes (e.g., significantly high costs or severe damages to the system) brought by some scenarios. Conditional value at risk (CVaR) [25] is a technique that adds more weight to more extreme random outcomes. Thus, CVaR is a more risk-averse approach than SO and has been used by [26, 27, 28, 29] for VPP scheduling.

Models that use SO and CVaR both rely on the assumption that the true distribution for random variables is known, so they can provide poor results if the distribution is estimated inaccurately. Robust optimization (RO) [30] is a method that optimizes the objective over the worst-case realization of a random variable and has been used by [31, 32, 33] for VPP scheduling. Unlike SO and CVaR, RO methods do not rely on distributional information and are risk-averse. However, RO

methods can perform poorly in practice because they can cause decision-makers to invest too many resources to handle extreme outcomes that are unlikely to occur in reality. Therefore, RO methods are prone to giving overly-conservative solutions.

Other methods, such as recent data-driven approaches like [34] and [35], have also been employed to tackle the VPP scheduling problem by leveraging available data for strategic decision-making. However, it is important to note that these methods often rely on strong assumptions regarding the distributions of the data used, which may restrict their applicability to more general stochastic systems. Distributionally robust optimization (DRO) [36] stands out as a method that offers distinct advantages compared to previous approaches. Unlike SO, CVaR, and the aforementioned data-driven methods, DRO does not rely on strong assumptions about the underlying distributional information. Instead, DRO considers the incomplete distributional information: the true distribution of uncertain parameters is unknown, but an ambiguity set that contains the true distribution can be derived. Then, DRO optimizes over such an ambiguity set of probability distributions to determine the worst-case probability distribution, thereby providing a robust solution that hedges against uncertainty. One key advantage of DRO is that it avoids the production of overly-conservative solutions often associated with robust optimization (RO) methods. By considering a range of possible distributions rather than a single worst-case scenario, DRO strikes a balance between robustness and conservatism. This flexibility allows for more effective decision-making under uncertainty.

There have been several instances of data-driven DRO methods employed to address optimal scheduling for virtual power plants (VPPs) and other applications. For example, [37] and [38] utilize chi-squared and norm-based DRO approaches, respectively. However, these methods have limitations in terms of flexibility in utilizing available data. In this chapter, we propose the utilization of a Wasserstein DRO method that enables a full utilization of data information. Specifically, by leveraging historical data on uncertain parameters, such as renewable energy output and demand, we can construct an ambiguity set that encompasses the true probability distribution of the random parameter. The robustness, or radius, of the ambiguity set is measured using the Wasserstein metric.

While previous attempts have been made to apply the Wasserstein DRO approach to solve VPP problems, such as in the works [39], [40], and [41], they are limited to a two-stage setting, where the scheduling decision is made 24 hours in advance before the start of the time horizon. In contrast, our proposed methodology introduces a novel combination of the Wasserstein DRO approach with a dynamic multi-stage framework for optimizing VPP scheduling. This dynamic multi-stage

framework allows us to model the VPP setting more accurately, considering the constant influx of information and enabling real-time hourly decision-making. By incorporating a multi-stage approach with dynamic programming (DP), our method captures the evolving nature of the VPP system and facilitates adaptive decision-making. While some dynamic programming approaches have been applied to multi-stage scheduling problems in VPPs or similar applications, such as [42], [43], and [44], it is worth noting that none of these methods incorporate distributional robustness. We propose a multi-stage distributionally robust optimization (MSDRO) that combines multi-stage optimization and DRO. MSDRO addresses the limitations of existing data-driven approaches, incorporates distributional robustness, and models uncertainties in the VPP scheduling problem more comprehensively. A more detailed summary of our main contributions is presented below:

1. Our model introduces a higher dimension of uncertainty compared to previous DRO literature. Specifically, we incorporate random demand, market prices, as well as wind and solar generation. This extension allows for a more comprehensive representation of uncertainties present in VPP scheduling.

2. We are the first to combine the Wasserstein DRO approach with a dynamic multi-stage framework to optimize VPP scheduling in a real-time economic dispatch setting.

3. We provide empirical results to illustrate the advantages of the distributional robustness and multi-stage modeling methods and compare our method to other data-driven approaches.

The rest of the chapter is as follows. In Section 2.2 we introduce and define the different VPP components considered in our problem and formulate the mathematical model for the VPP scheduling problem. In Section 2.3, we introduce the solution methodology and algorithm used to solve the problem formulated in Section 2.2. In Section 2.4, we present the experimental results that demonstrate the effectiveness of our multi-stage DRO model for VPP scheduling. In Section 2.5, we conclude the chapter.

## *2.2 VPP Model Formulation*

For our problem, we consider a VPP with six major components:

1. distributed generators

2. energy storage system (ESS)

3. demand response (DR)

4. load shedding (LS)

5. wind farms

6. photovoltaic cells

We consider the real-time economic dispatch problem where the VPP must supply enough energy to satisfy random customer demand, and they can update dispatch decisions each hour after observing random outcomes in the previous hour. First, we formulate the corresponding VPP scheduling problem as a linear program model. We use $[T] := \{1, .., T\}$ to represent the set of all positive integers up to $T$. Note that the formulation we present in this section does not account for the uncertainty and non-anticipativity of dispatch decisions. We will incorporate uncertainty and non-anticipativity in Section 2.3.

### *2.2.1 Objective Function*

$$\min \sum_{t}^{T} \left( \sum_{n}^{N} z_{nt}^{G} + \sum_{m}^{M} z_{mt}^{ESS} + z_{t}^{DR} + z_{t}^{LS} - \theta_{t}^{P} s_{t} \right) \tag{2.1}$$

The objective function (2.1) minimizes the total cost, which includes distributed generator cost $z^G$, battery cost $z^{ESS}$, demand response cost $z^{DR}$, and load shedding cost $z^{LS}$. The revenue that the VPP can earn by selling excess energy $s_t$ to the market is represented by $\theta_t^P s_t$. In our framework, the VPP is a price-taker, and we treat the market price, $\theta_t^P$, as a random variable. $N$ is the number of distributed generators, $M$ is the number of ESS units, and $T$ is the time horizon.

### *2.2.2 Distributed Generator Constraints*

In our setting, we consider distributed generators like the ones used in [18]. Since we consider a real-time setting, we assume that all distributed generators are turned on during the entire time horizon and that any unit commitment decisions have already been made. The distributed generators have quadratic fuel costs, which are approximated with a convex piecewise linear function similar to [18].

$$g^G_{nt} - g^G_{n,t-1} \leq RU_n \quad \forall n \in N, t = 2, \cdots T \tag{2.2}$$

$$g^G_{n,t-1} - g^G_{nt} \leq RD_n \quad \forall n \in N, t = 2, \cdots T \tag{2.3}$$

$$\underline{P}^G_n \leq g^G_{nt} \leq \overline{P}^G_n \quad \forall n \in N, t \in [T] \tag{2.4}$$

$$z^G_{nt} \geq \beta_{nk} + \alpha_{nk} g^G_{nt} \quad \forall n \in N, t \in [T], k \in [K] \tag{2.5}$$

$g_{nt}$ is the geneartion level for distributed generator $n$ at time $t$. $\overline{P}^G_n$ and $\underline{P}^G_n$ are the minimum and maximum generation levels, respectively. $RU_n$, and $RD_n$ are the ramp up and down limits, respectively. $\alpha_{nk}$ and $\beta_{nk}$ are quadtraic fuel cost approximation parameters, and $K$ is the number of breakpoints for the piecewise-linear approximation. (2.2) and (2.3) represent ramping up and ramping down limits, respectively. (2.4) represents the generation level capacities. (2.5) represents a piecewise linear approximation for the quadratic fuel cost with respect to the generation level.

### *2.2.3 ESS Constraints*

The energy storage system allows the VPP to store excess energy to be used to satisfy demand in future periods.

$$h_{mt} = h_{m,t-1} + \nu^+_m g^{E+}_{m,t-1} - \nu^-_m g^{E-}_{m,t-1} \quad \forall m \in M, t = 2, \cdots T \tag{2.6}$$

$$h_{mt}, g^{E+}_{mt}, g^{E-}_{mt} \leq \overline{P}^{ESS}_m, \overline{P}^+_m, \overline{P}^-_m \quad \forall m \in M, t \in [T] \tag{2.7}$$

$$z^{ESS}_{mt} = \lambda_{ESS}(g^{E+}_{mt} + g^{E-}_{mt}) \quad \forall m \in M, t \in [T] \tag{2.8}$$

$h_{mt}$ is the state of charge of ESS $m$ at time $t$. $g^{E+}_{mt}$ and $g^{E-}_{mt}$ are the energy charged and discharged at ESS $m$ at time $t$, respectively. $\overline{P}^{ESS}_m, \overline{P}^+_m, \overline{P}^-_m$ are the storage limit, charging limit, and discharging limit for ESS $m$, respectively. $\nu^+_m$ and $\nu^-_m$ are the respective charging and discharging efficiency factors for ESS $m$. $\lambda_{ESS}$ is the unit price for charging or discharging. (2.6) is an energy balance constraint indicating that the current state of charge (SOC), denoted by $h_{mt}$, is equal to the previous SOC plus the energy charged minus the energy discharged, adjusted by the efficiency factor. (3.21) shows the respective capacity limits for the state of charge, charging, and discharging, respectively. We assume that the market price is non-negative, therefore, these constraints are sufficient without including a binary variable to indicate whether the ESS is charging or discharging. Finally, (2.8) represents the battery cost calculation.

### *2.2.4 Demand Response and Load Shedding Constraints*

The cost of implementing demand response is related to the amount of demand change, so we adopt a step-wise DR cost curve, similar to the one used in [45].

$$0 \leq u_{lt}^{DR} \leq R_{t,l} - R_{t,l-1} \quad \forall t \in [T], l \in [L] \tag{2.9}$$

$$u_t^{DR} = \sum_{l}^{L} u_{lt}^{DR} \quad \forall t \in [T] \tag{2.10}$$

$$z_t^{DR} = \sum_{l}^{L} \rho_{lt} u_{lt}^{DR} \quad \forall t \in [T] \tag{2.11}$$

$$z_t^{LS} = V u_t^{LS} \quad \forall t \in [T] \tag{2.12}$$

$$u_t^{DR} + u_t^{LS} \leq \min\{DR_{max}, \theta_t^D\} \quad \forall t \in [T] \tag{2.13}$$

We consider $L$ demand response price intervals. $u_{lt}^{DR}$ is the demand response at level $l$ at time $t$. $u_t^{DR}$ and $u_t^{LS}$ are the total demand response and load shedding at time $t$, respectively. $R_{t,l}^{DR}$ is the upper limit for demand response in pricing interval $l$. $DR_{max}$ is the maximum allowable demand response, and $\theta_t^D$ is the random customer demand at time $t$. $V$ is the value of lost load, which is the unit price for load shedding. The cost at time $t$ of using demand response within in the $l^{th}$ interval between $R_{t,l}^{DR}$ and $R_{t,l-1}^{DR}$ is $\rho_{lt}$. This relationship is captured by constraints (3.16)-(2.11). The load shedding cost is represented by (2.12), where $V$ represents the unit penalty of load shedding, which is typically greater than the demand response cost. (3.18) ensures that the sum of total demand response and load shedding cannot exceed the demand response limit and realized demand.

### *2.2.5 Demand Balance Constraint*

Lastly, (2.14) is a demand balance constraint that requires that the total supply and total demand are equal.

$$\theta_t^W + \theta_t^{PV} + g_t^{E-} - g_t^{E+} + \sum_{n \in N} g_{nt}^G = \theta_t^D - u_t^{DR} - u_t^{LS} + s_t \quad \forall t \in [T] \tag{2.14}$$

$\theta_t^W$ and $\theta_t^{PV}$ are the random wind and solar output at time $t$, respectively. Output from renewable energy and distributed generator output with net ESS discharge comprise the total supply, while observed demand, demand response, and load shedding comprise the total demand. If supply exceeds demand, any excess energy can be sold to the market via $s_t$. All decision variables are

non-negative. Note that because we use random variables $\theta_t^W$ and $\theta_t^{PV}$ to aggregately capture the total wind and solar generation levels respectively, we can efficiently model systems with a large number of renewable DERs, such as the Tesla [46] or Sunrun [47] VPPs that may have thousands of renewable energy units.

#### *2.2.6 Abstract Formulation*

For brevity, we let $x_t(\theta^t)$ represent a vector of decision variables, such as generation output, charging and discharging levels, and demand response, which depend on the realization of the random vector $\theta^t := (\theta_t^W, \theta_t^{PV}, \theta_t^D, \theta_t^P)$ representing random wind output, solar output, customer demand, and the market price at time $t$, respectively. Also, we define $x_0$ as the initial state of the system. The abstract form of the above formulation can then be represented as:

$$\begin{aligned} \min \ & \sum_{t=1}^{T} c_t(\theta^t) x_t(\theta^t) \\ s.t. \ & A_t x_t(\theta^t) = B_t x_{t-1} + b_t(\theta^t) \quad \forall t = 2, \cdots, T \\ & x_t(\theta^t) \geq 0 \quad \forall t \in [T] \end{aligned} \tag{2.15}$$

$c_t(\theta^t)$ is the objective coefficient which also depends on the random variable $\theta^t$ that captures the objective function's dependence on the random market price in our setting. $b_t(\theta^t)$ is a constant parameter vector in the constraints, which also depends on $\theta^t$, that captures the dependence on random renewable energy and demand in our setting. This abstract formulation provides us with a versatile framework that allows for the incorporation of multiple dimensions of uncertainty in both the constraints and the objective coefficients. This distinguishes our approach from many previous methods, such as [48], which primarily focus on considering randomness solely in the constraints. Next, we will present the tractable reformulation of (3.27) and propose an efficient algorithm to solve the reformulation correspondingly.

### *2.3 Methodology*

#### *2.3.1 Reformulation Using Wasserstein Ambiguity Set*

To present the MSDRO formulation, we first define the ambiguity set, which is a set of distributions that are considered in our model. There are two standard methods of defining the ambiguity set. The first is a moment-based ambiguity set, which includes all distributions that satisfy moment requirements on their first and second moments [49]. The second type of ambiguity set is a

distribution-based ambiguity set, which considers all distributions that are sufficiently close to a reference distribution, such as the empirical distribution learned from data [50].

We adopt the second method, which better utilizes available data information, and we use the Wasserstein distance [51] as a metric to measure how close a distribution is to the reference distribution. Formally, the Wasserstein distance $d_W(P_t, \hat{P}_t)$ between two distributions $P_t, \hat{P}_t$ is defined as:

$$d_W(P_t, \hat{P}_t) := \inf_{z \in Z} \int_{\Xi \times \Xi} \kappa(x, y) dz(P_t, \hat{P}_t) \tag{2.16}$$

where $(\Xi, d)$ is the metric space, $\kappa$ is a general distance between random variables $x$ and $y$, and $Z$ is the set of all joint distributions.

Next, we define the ambiguity set $D_t$ that includes all distributions $P_t$, such that the Wasserstein distance between $P_t$ and the reference distribution $\hat{P}_t$ is less than some pre-defined constant $r \in \mathbb{R}$, which we call the "radius" of $D_t$:

$$D_t := \{P_t : d_W(P_t, \hat{P}_t) \leq r\}. \tag{2.17}$$

In our setting, we consider discrete random variables and $J$ random scenarios. We let $P_t = \{p_j^t, j \in [J]\}$ be the probability mass function (pmf) of random variable $\theta^t$ at time $t$, where $p_j^t$ represent the probability that the $j^{th}$ random scenario $\theta_j^t$ occurs at time $t$. Similarly, $\hat{P}_t = \{\hat{p}_j^t, j \in [J]\}$ is the pmf for the reference distribution for $\theta^t$ at time $t$, where $\hat{p}_j^t$ represent the probability that the $j^{th}$ random scenario $\theta_j^t$ occurs at time $t$. Therefore, (2.16) can be reformulated as:

$$\begin{aligned} d_W(P_t, \hat{P}_t) = & \min_z \sum_{i=1}^{J} \sum_{j=1}^{J} \hat{d}_{ij}^t z_{ij} \\ s.t. \ & \sum_{j=1}^{J} z_{ij} = \hat{p}_i^t \quad \forall i \in [J] \\ & \sum_{i=1}^{J} z_{ij} = p_j^t \quad \forall j \in [J] \\ & z_{ij} \geq 0 \quad \forall i, j \in [J] \end{aligned} \tag{2.18}$$

where $\hat{d}_{ij}^t$ represents the $L_p$ distance between $\theta_i^t$ and $\theta_j^t$ for $p \geq 1$. In this work, we use $L_2$ distance:

$$\hat{d}_{ij}^t := \sqrt{\sum_{k=1}^{n} (\theta_{i,k}^t - \theta_{j,k}^t)^2} \tag{2.19}$$

where $n$ is the number of components for the random vectors $\theta_i^t$ and $\theta_j^t \in \mathbb{R}^n$. $\theta_{i,k}^t$ and $\theta_{i,k}^t$ represent the $k^{th}$ component of $\theta_i^t$ and $\theta_j^t$, respectively.

### *2.3.2 Multi-Stage DRO Formulation*

Using the definition of the ambiguity set, we now provide a multi-stage reformulation of the original problem, (3.27). This framework provides the decision-maker with the flexibility to make dynamic or real-time decisions, allowing them to make sequential decisions based on observed random outcomes at each time period. This stands in contrast to a "static" decision-making setting commonly represented by a two-stage day-ahead framework. In such static settings, a scheduling decision is made 24 hours in advance before the start of the time horizon and remains fixed throughout the horizon. We consider an hourly decision-making framework, however, it is important to emphasize that this framework can be generalized to accommodate sequential decision-making at different time scales, such as every 15 minutes or monthly intervals.

We start by defining $Q_t(x_{t-1}, \theta^t)$ as the "cost-to-go" function, which is the cumulative cost from time $t$ to the end of the horizon at time $T$ if $x_{t-1}$ was chosen in time period $t-1$ and $\theta^t$ was observed. This multi-stage formulation captures the non-anticipativity of dispatch decisions because realistically, the VPP operator can adjust dispatch decisions based on observing the random solar output, wind output, demand, and market price information in the current period, but cannot observe these outcomes in future periods.

In our distributionally robust setting, the objective is to minimize the cost in the current time period plus the total expected future cost under the worst-case distribution in $D_t$. Accordingly, we can formulate distributionally robust Bellman's equations [52] (2.20) and (2.21) as follows:

$$Q_t(x_{t-1}, \theta^t) = \min_{x_t} c_t(\theta^t)^T x_t + \max_{P_{t+1} \in D_{t+1}} \mathbb{E}_{P_{t+1}}[Q_{t+1}(x_t, \theta^{t+1})] \tag{2.20a}$$

$$\text{s.t. } A_t x_t = B_t x_{t-1} + b_t(\theta^t) \tag{2.20b}$$

$$x_t \geq 0 \tag{2.20c}$$

for $t \in [T]$, and

$$Q_{T+1}(x_T, \theta^{t+1}) = 0 \tag{2.21}$$

It is important to notice that the worst-case distribution $P_{t+1}$ is the solution to the inner-maximization problem, so it is an output that cannot be known before solving the problem. In our specific setting, it is challenging to ascertain the specific distribution that would lead to the worst-case expected cost, primarily due to the presence of multiple factors in our model that contribute to cost increases (e.g., generation cost, demand response cost, or load shedding cost). This lack of knowledge about the exact worst-case distribution can be effectively addressed by the DRO approach: it still provides a solution with respect to the worst-case distribution without necessitating prior knowledge of its exact form.

In our setting, since we consider a discrete set of $J$ possible realizations for $\theta^t$,$\mathbb{E}_{P_{t+1}}[Q_{t+1}(x_t, \theta^{t+1})]$ can be rewritten as:

$$\sum_{j=1}^{J} p_j^{t+1} Q_{t+1}(x_t, \theta_j^{t+1}) \tag{2.22}$$

Now, using the definition of Wasserstein distance given in (2.18), we can reformulate the inner-maximization problem (which is the second term in (2.20a)) as the following:

$$\max_{z, p^{t+1}} \sum_{j=1}^{J} p_j^{t+1} Q_{t+1}(x_t, \theta_j^{t+1}) \tag{2.23a}$$

$$s.t. \quad \sum_{i=1}^{J} \sum_{j=1}^{J} \hat{d}_{ij}^{t+1} z_{ij} \leq r \tag{2.23b}$$

$$\sum_{j=1}^{J} z_{ij} = \hat{p}_i^{t+1} \quad \forall i \in [J] \tag{2.23c}$$

$$\sum_{i=1}^{J} z_{ij} - p_j^{t+1} = 0 \quad \forall j \in [J] \tag{2.23d}$$

$$z_{ij} \geq 0 \quad \forall i, j \in [J] \tag{2.23e}$$

The solution to the optimization problem (2.23) is the worst-case distribution. By taking the dual of (2.23), we can convert it to a minimization problem and combine it with the outer minimization problem (2.20) to get the following:

$$Q_t(x_{t-1},\theta^t) = \min_{x_t,\phi^t,\psi^t} c_t(\theta^t)x_t + r\phi^t + \sum_{j=1}^{J} \hat{p}_j^{t+1}\psi_j^t \tag{2.24a}$$

$$s.t.\ \ A_t x_t = B_t x_{t-1} + b_t(\theta^t) \tag{2.24b}$$

$$\hat{d}_{ij}^{t+1}\phi^t + \psi_i^t + \omega_j^t \geq 0 \quad \forall i,j \in [J] \tag{2.24c}$$

$$-\omega_j^t \geq Q_{t+1}(x_t,\theta_j^{t+1}) \quad \forall j \in [J] \tag{2.24d}$$

$$x_t, \phi^t \geq 0 \tag{2.24e}$$

where $\phi^t$, $\psi^t$, and $\omega^t$ are dual variables of constraints (2.23b),(2.23c), and (2.23d) respectively. Note here that constraints (2.24c) and (2.24d) can be combined into a single constraint:

$$\hat{d}_{ij}^{t+1}\phi^t + \psi_i^t \geq Q_{t+1}(x_t,\theta_j^{t+1}), \forall i,j \in [J] \tag{2.25}$$

Let $F(x_{t-1},\theta^t)$ denote the feasible region of $x_t$ in (2.20). Without loss of generality, we can assume $F(x_{t-1},\theta^t) \neq \emptyset$ for $\forall t \in [T]$, because if otherwise, we can add a penalty term in the objective function to guarantee feasibility. Furthermore, $F(x_{t-1},\theta^t)$ is compact because it is a closed polyhedral set and all decision variables are bounded by (2.14). [48] proved that if $F_t(x_{t-1},\theta^t)$ is nonempty and compact for any feasible $x_{t-1}$ and $\theta^t$, then the function $Q_{t+1}(x_t,\theta^{t+1})$ is a piecewise linear convex function on $F_t(x_{t-1},\theta_t)$ with a finite number of pieces. Using this fact, we can find a reformulation for (2.24) as follows:

$$Q_t(x_{t-1},\theta^t) = \min_{x_t,\phi_t,\psi^t} c_t(\theta^t)x_t + r\phi_t + \sum_{j=1}^{J} \hat{p}_j^{t+1}\psi_j^t \tag{2.26a}$$

$$s.t.\ \ A_t x_t = B_t x_{t-1} + b_t(\theta^t) \tag{2.26b}$$

$$\hat{d}_{ij}^{t+1}\phi^t + \psi_i^t \geq M_{j,k}^t x_t + h_{j,k}^t \quad \forall j \in [J], k \in \Xi_j^{t+1} \tag{2.26c}$$

$$x_t, \phi_t \geq 0 \tag{2.26d}$$

Here, $\Xi_j^{t+1}$ is the index set for all pieces that describe $Q_{t+1}(x_t,\theta^{t+1})$ as a function of $x_t$ with gradients $M_{j,k}^t$ and intercepts $h_{j,k}^t$. For all $k \in \Xi_j^t$, for a fixed solution $\hat{x}_t$ and a fixed scenario $\theta_j^{t+1}$, the formula for gradient and intercept are:

$$M_{j,k}^{t} = \eta_{jk}^{t+1} B_{t+1} \tag{2.27a}$$

$$h_{j,k}^{t} = Q_t(\hat{x}_t, \theta_j^{t+1}) - \eta_{jk}^{t+1} B_{t+1} \hat{x}_t \tag{2.27b}$$

where $\eta_{jk}^{t+1}$ is the optimal dual variable of constraint (2.26b) in period $t+1$. The cardinality of $\Xi_j^{t+1}$ is equal to the number of extreme points of the dual problem of (2.26) for $t+1$. As a result, solving (2.26) can be computationally challenging due to the large number of constraints in (2.26c). Instead of adding all constraints in (2.26c) corresponding to the entire set $\Xi_j^{t+1}$, we relax (2.26) by iteratively adding constraints in (2.26c). Therefore, we have a relaxation of (2.26), which we denote as $\tilde{Q}_t(x_{t-1}, \theta^t)$, by replacing constraint (2.26c) with:

$$\hat{d}_{ij}^{t+1} \phi^t + \psi_i^t \geq \tilde{M}_{j,k}^{t} x_t + \tilde{h}_{j,k}^{t}, \forall i, j \in [J], k \in \tilde{\Xi}_j^{t+1} \tag{2.28}$$

where $\tilde{\Xi}_j^{t+1} \subseteq \Xi_j^{t+1}$. We denote this problem as $P_t^k$.

### *2.3.3 Distributionally Robust Dynamic Programming Algorithm*

Using the framework presented above, we can solve for $\tilde{Q}_t(x_{t-1}, \theta^t)$ using the following distributionally robust dynamic programming (DRDP) algorithm, motivated by [48]. Here, $K$ is the total number of iterations, which is equivalent to the number of constraints (2.28) we add. For each iteration, the algorithm performs a forward pass where it solves the problem $P_t^k$ from periods $t = 1, ..., T-1$ by sampling $\hat{\theta}_1, ..., \hat{\theta}_{T-1}$ from the set of possible scenarios (in our experimental setting we consider 100 scenarios for each time period). The optimal solution at each stage, $\hat{x}_1, ..., \hat{x}_{T-1}$, can be obtained. Then, the algorithm performs a backward pass where it goes from $t = T, ..., 2$ and it solves $P_t^k(\hat{x}_{t-1}, \theta^j)$ for all $j$ and generates constraints (2.28) for $P_{t-1}^k$. The main purpose of the backward pass is to generate the cuts and add them to the problem in the previous time period. This is done by obtaining the value of $\eta_{tk}^j$, which represents the dual variable from the optimization problem $P_t$. $\eta_{tk}^j$ is used via (2.27a) and (2.27b) to define the cut of the form (2.28) to be added to the problem $P_{t-1}$. The full details of the algorithm are presented in Figure 2.1.

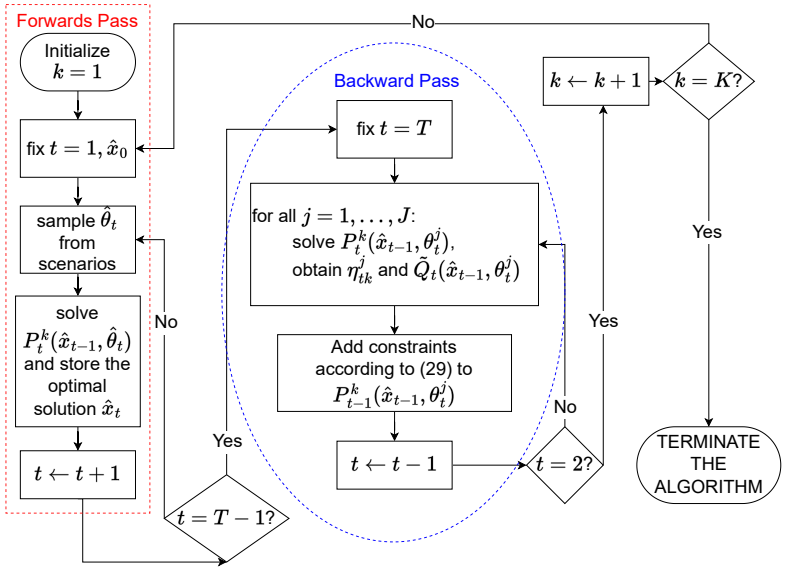



Figure 2.1: DRDP algorithm

Our dynamic Wasserstein DRO framework utilizes two streams of data to inform decision-making. The first stream is historical data, which establishes the reference distribution for random parameters and guides the derivation of the optimal decision policy. The second stream is real-time data, which reflects the actual realization of random parameters at each time period or simulates the dynamic nature of the real-world environment. Both historical data and real-time data can be synthetic or real, such as simulated scenarios or historical records of renewable energy output and demand trends.

### *2.4 Experimental Results*

We perform four types of experiments. The first is to test how the system's robustness level impacts the total cost. The second is to measure how much a dynamic approach improves performance relative to a static approach. The third is to investigate the importance of different components of the VPP on system performance. Finally, we perform experiments to explore the scalability and efficiency of our algorithm.

We consider a system with three distributed generators, one ESS, one PV, one wind farm, a demand response program that allows load curtailment, and a load that allows load shedding. The data for the model parameters are presented in Tables 2.1-2.3. We use the Gurobi solver to solve the model on a 1.6 GHz Dual-Core Intel Core i5 processor. For all experiments, we consider a time

horizon of $T = 24$ hours, and we set the number of cuts to $K = 20$. The computation time was roughly 5 seconds to implement one simulation of the DRDP algorithm for any value of $r$. Table **??** includes the fuel cost parameters, where the fuel cost for each generator is $f(p) = \hat{a}p^2 + \hat{b}p + \hat{c}$ for generation level $p$. This quadratic function is approximated with constraint (2.5) with $K = 5$.

| Gen | $\underline{P}_n^G$ (kW) | $\overline{P}_n^G$ (kW) | $RU_n, RD_n$ (kW/h) | $\hat{a}$ (\$/kW$^2$) | $\hat{b}$ (\$/kW) | $\hat{c}$ (\$) |
|---|---|---|---|---|---|---|
| 1 | 0 | 20 | 10 | 0.0697 | 29.24 | 31.67 |
| 2 | 0 | 50 | 25 | 0.0098 | 22.04 | 58.81 |
| 3 | 0 | 100 | 50 | 0.0004 | 6.00 | 50.00 |

Table 2.1: Distributed generator parameter values

| $\nu$ | $\overline{P}_m^{ESS}$(kW) | $\overline{P}_m^+$ (kW) | $\overline{P}_m^-$(kW) |
|---|---|---|---|
| 0.9 | 1000 | 100 | 100 |

Table 2.2: ESS parameters

| Level | $\rho_{lt}$ (\$/kW) | $R_{t,l}$ (kW) |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 100 | 100 |
| 2 | 200 | 200 |
| 3 | 300 | 300 |

Table 2.3: Demand response parameter values (\$/kW)

Lastly, $DR_{max} = 300$ kW and $V = \$500$. To test the model performance, we implement a training procedure and an out-of-sample testing procedure. The training procedure serves the purpose of running the DRDP algorithm, which generates the necessary cuts to define the reformulated distributionally robust Bellman's equation (2.26). In this context, our aim is not to train any specific parameters, but rather to train the constraints (or cuts) of the distributionally robust Bellman's

| hour | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| price LB ($) | 10 | 30 | 10 | 12 | 14 | 17 | 22 | 30 | 45 | 55 | 90 | 110 |
| price UB ($) | 30 | 50 | 30 | 32 | 34 | 37 | 42 | 50 | 65 | 75 | 110 | 130 |
| load LB (kW) | 200 | 200 | 200 | 200 | 225 | 225 | 250 | 300 | 350 | 350 | 350 | 400 |
| load UB (kW) | 275 | 275 | 275 | 275 | 300 | 300 | 325 | 325 | 400 | 400 | 425 | 500 |
| wind LB (kW) | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| wind UB (kW) | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| solar LB (kW) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 | 100 | 200 | 300 |
| solar UB (kW) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25 | 100 | 200 | 300 | 400 |
| hour | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| price LB ($) | 120 | 115 | 108 | 95 | 89 | 82 | 71 | 62 | 40 | 30 | 20 | 10 |
| price UB ($) | 140 | 135 | 128 | 115 | 109 | 102 | 91 | 82 | 60 | 50 | 40 | 30 |
| load LB (kW) | 350 | 350 | 350 | 350 | 350 | 400 | 500 | 550 | 500 | 450 | 350 | 250 |
| load UB (kW) | 500 | 500 | 500 | 500 | 500 | 600 | 700 | 800 | 700 | 600 | 500 | 400 |
| wind LB (kW) | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| wind UB (kW) | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| solar LB (kW) | 500 | 400 | 400 | 400 | 400 | 140 | 200 | 150 | 100 | 100 | 100 | 100 |
| solar UB (kW) | 700 | 500 | 500 | 500 | 400 | 300 | 200 | 100 | 25 | 0 | 0 | 0 |

Table 2.4: Parameters for scenario generation distribution

equation problem. Subsequently, during the testing procedure, the distributionally robust Bellman's equation is solved sequentially with the cuts generated in the training stage. For each time period, a random realization is sampled from a distribution that differs from the one used during the training procedure. This sampling process simulates a real-life dynamic VPP scheduling environment.

For the training procedure, we implement the DRDP algorithm with respect to 100 scenarios that are sampled from a uniform distribution. The upper-bound (UB) and lower-bound (LB) for these uniform distributions for prices, load, and renewables (solar plus wind) are displayed in Table 2.4.

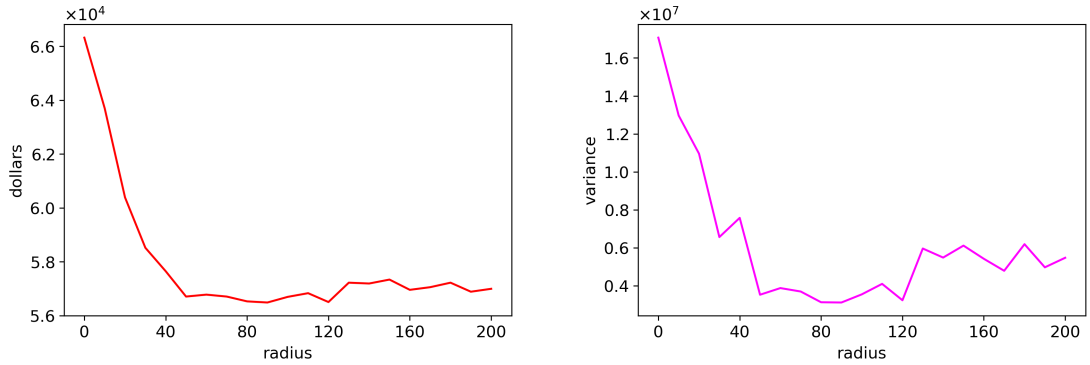
For our out-of-sample testing procedure, we test the policy generated in the training procedure by recording the average cost over 100 sample paths. To test the robustness of the proposed model,

we allow the out-of-sample uncertainty to have a higher mean demand and higher demand variability than anticipated. Specifically, for the testing procedure, we sample from a new uniform distribution where the load upper bound is increased by 150 kW and the load lower bound is decreased by 50 kW.

We perform the training and testing procedure for different values of the Wasserstein radius $r$, and for each radius level, we implement the training and testing procedure for 100 simulations. The initial generation levels of all generators and the state of charge of ESS are set to zero.

### *2.4.1 Robustness Analysis*

The objective of this experiment is to investigate how different radius values for the Wasserstein ambiguity set affect the VPP scheduling cost. We test different radius values between $r = 0$ and $r = 200$, and analyze the behavior of the average cost in the testing procedure across these different radius values. The results are shown below in Figures 2.2a and 2.2b.




(a) Average cost vs. radius

(b) Variance vs. radius

Figure 2.2: Impact of changing the radius

The value of $r$ represents the maximum allowable distance between the reference and true distributions, which measures the conservativeness level that the decision-maker wishes to incorporate into the decision-making process. Intuitively, a larger radius indicates a greater willingness to accommodate outcomes that deviate significantly from the reference distribution. In practice, the choice of the radius value should align with the decision-maker's level of risk aversion. When $r = 0$, this implies that the ambiguity set contains only one distribution: the true distribution. This setting

indicates that there is no distributional uncertainty, and the true distribution is known perfectly. Therefore, when $r = 0$, MSDRO is equivalent to SO.

The results from Figure 2.2a show that the average cost decreases as $r$ increases from 0 to about 70. That is because when $r$ is small, the model is not capable of considering extreme scenarios, and therefore has limited ability to handle them when they occur. When $r$ becomes larger, the model takes into account more distributional ambiguity and thus can plan for more extreme scenarios to occur. This allows the system to schedule more robustly and reduce the impact of costly events. Therefore, increasing $r$ allows our model to consider more distributional robustness and allows the DRO model to outperform the SO model. From about $r = 60$ to 120, we do not observe any significant change in cost, which means that at this point, considering more distributional ambiguity does not change the model's behavior. Finally, for radius values greater than about 120, the cost is greater. This behavior is the result of overly-conservative behavior where the model spends extra money to prepare for more extreme scenarios that are considered because of the larger radius value, which exemplifies behavior that is common for RO methods.

These results show that by choosing appropriate radius values (between roughly 60 and 120), our MSDRO method considers the appropriate amount of distributional ambiguity to give a solution that is cost-effective, but not overly-conservative.

From Figure 2.2b, we can see that as $r$ increases to about 50, there is a sharp decrease in variance, and the variance remains at this level until about $r = 120$. Therefore, in addition to decreasing cost, increasing the radius decreases the variability, and thus improves system reliability. For $r \geq 120$, the variance is larger which indicates that further increasing the distributional robustness may introduce more variability.

Next, we compare the performance of the stochastic-VPP (SVPP) and distributionally robust VPP (DRVPP) on an hourly basis. Specifically, to represent the SVPP case in our experimental setup, we implement a stochastic dual dynamic programming method, which is a common method used for modeling dynamic programs with uncertainty (e.g., in [42] and [43], among others). For our comparison, we use $r = 70$ as a representative case of DRVPP. Figure 2.3 shows the difference in the cumulative cost between the two methods.

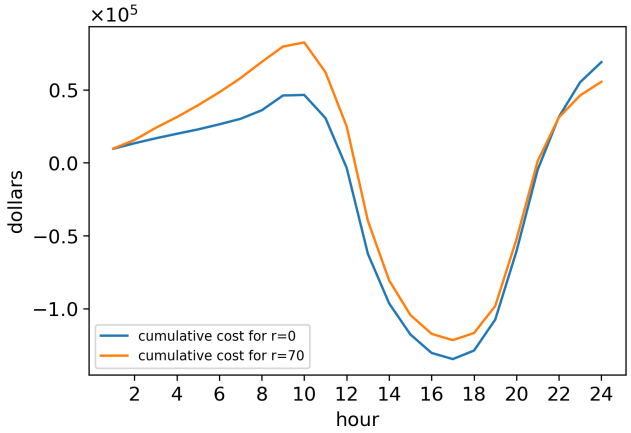



Figure 2.3: Hourly cumulative cost

We can see from Figure 2.3, that for the first 22 hours, the SO case outperforms the DRO case (note that a negative cost means a positive profit), but the DRO case yields a lower *cumulative* cost over the entire time horizon. This behavior highlights the nature of our algorithm, which is not based on a greedy approach. Instead, it strategically sacrifices a relatively small amount of profit earlier in the time horizon to ensure a lower overall cost by the end of the time horizon.

We can further observe how the optimal policy generated in the training phase differs between the SO and DRO cases. We compare the optimal values of ESS storage, demand response utilization, and load shedding, shown in Figures 2.4 and 2.5, respectively.

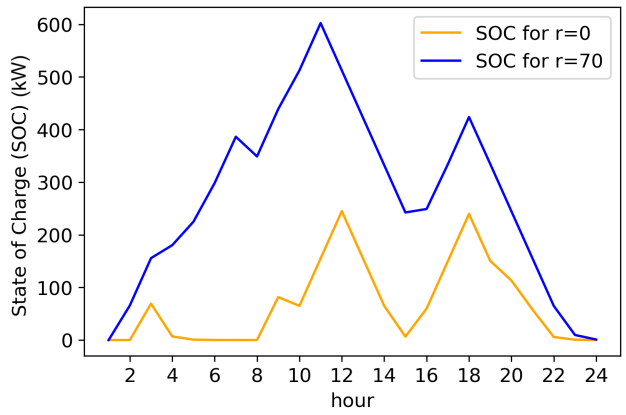



Figure 2.4: Battery usage for differing radius values

Figure 2.4 shows the state of charge (SOC) value of the ESS, so upward slopes and downward

slopes in the graph represent periods of charging and discharging, respectively. We can observe that the DRVPP tends to store more energy overall than the SVPP, and the DRVPP stores more energy leading up to the demand spike at $t = 20$ and 21. These results can explain why the SVPP has lower cumulative costs earlier in the time horizon, as observed in Figure 2.3. Essentially, by storing more energy earlier in the time horizon, the DRVPP sacrifices opportunities to sell to the market or satisfy demand without DR when compared to the SVPP, but it makes up for it in the end by being better prepared to handle peak demand.

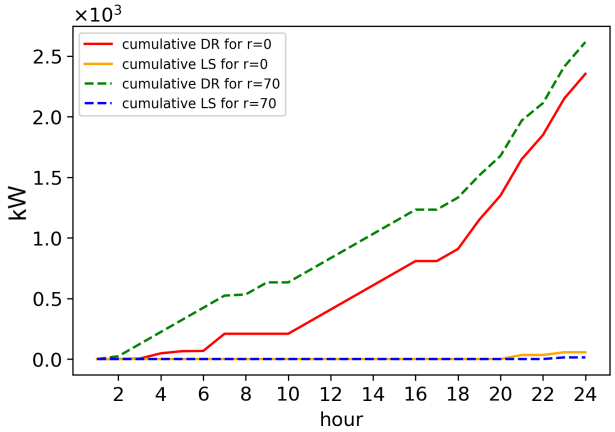



Figure 2.5: Cumulative DR and LS utilization

We can also observe the differences between how the SVPP and DRVPP utilize demand response and load shedding in Figure 2.5. We can see that the DRVPP chooses to spend more on DR to avoid load shedding. This causes the SVPP to load shed more during peak hours due to the insufficient utilization of DR. As a result, the DRVPP saves money compared to the SVPP because load shedding is significantly more expensive than DR.

These experiments demonstrated how our method is robust to distributional ambiguity. In our testing phase, since we conducted simulations where the samples are drawn from a testing distribution with higher mean demand and increased demand variability compared to the training distribution, the sampled scenarios during the testing phase can exhibit more extreme conditions than what would typically be expected. From the results, we can see that our DRO model (i.e., $r > 0$) outperformed the SO model (i.e., $r = 0$). That is, our model performs better even when the true distribution is unknown or inaccurately estimated, and exhibits greater robustness to extreme realizations of random events.

### *2.4.2 Dynamic vs. Static Comparison*

In this section, we compare the performance of our DRDP algorithm with a static-DRO algorithm to showcase the advantage of our dynamic DRO method. For the static-DRO algorithm, we implement a two-stage Wasserstein DRO method, similar to the method implemented in [39] and [40], for example. This static-DRO method also considers the worst-case distribution in the Wasserstein ambiguity set defined in (2.17), so the static DRO algorithm also gives a risk-averse solution by hedging against distributional ambiguity.

However, the static model must decide its generation profile 24 hours in advance before the start of the time horizon, so it cannot observe any information about the uncertainty and it cannot adjust it dynamically during the day. In our setup, the decision-maker in the static implementation must commit to distributed generation levels, battery charging, and discharging, and demand response 24 hours in advance. The decision-maker can only adjust load shedding and the amount of energy sold to the market in real-time after uncertain parameters are revealed.

If the true outcomes of random parameters deviate significantly from the predicted outcomes in the planning stage, the static-DRO model could give costly results. We illustrate this by comparing the performance of the static-DRO and dynamic-DRO algorithms across different radius values. We record the average cost for each case and present the results in Table 2.5. When the training and testing distributions are different, the static-DRO case performs worse than its dynamic counterpart, because it plans for random outcomes in the training phase that differ from the actual outcomes observed in the testing phase, and cannot adjust as flexibly as the dynamic-DRO method.

| $r$ | Static | Dynamic | $r$ | Static | Dynamic |
|---|---|---|---|---|---|
| 0 | 305,659.22 | 66,319.00 | 60 | 292,412.73 | 56,776.51 |
| 10 | 290,079.47 | 63,699.55 | 70 | 292,451.29 | 56,704.61 |
| 20 | 291,241.00 | 60,387.36 | 80 | 292,903.08 | 56,526.30 |
| 30 | 292,649.48 | 58,516.35 | 90 | 292,908.91 | 56,484.20 |
| 40 | 292,382.67 | 57,645.99 | 100 | 292,997.48 | 56,548.45 |
| 50 | 292,469.12 | 56,704.97 | – | – | – |

Table 2.5: Average cost for dynamic and static VPP scheduling ($)

The most significant contributor to the increased cost for the static case is the heavy reliance

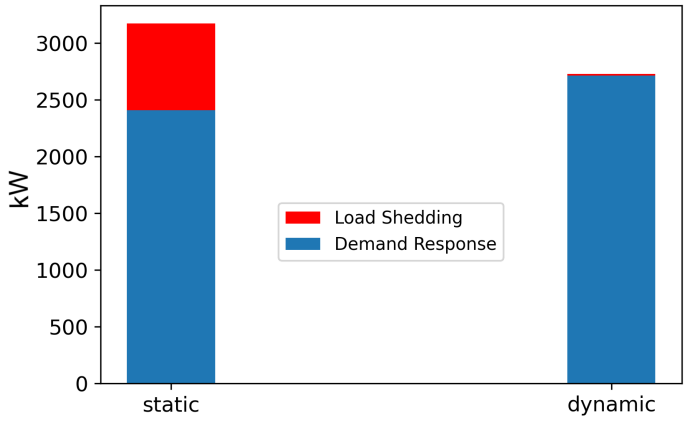



Figure 2.6: Static and dynamic DR and LS usage for $r = 70$

on load shedding. Figure 2.6 compares the amounts of DR and LS for the case when $r = 70$ for the static and dynamic cases. We can observe that the dynamic case spends more on demand response, and by doing so, its reliance on load shedding is negligible compared to the static case's. Because the static case is unable to adjust in real-time, it is unable to increase its demand response usage when it observes higher-than-expected demand, so it must resort to more load shedding.

These experiments demonstrate the importance of the dynamic nature of our DRDP method. By allowing the system to adjust to uncertainty in a real-time manner, the DRDP method outperforms the static method in cases where the realization of random outcomes is different from the predicted outcomes.

### *2.4.3 Component Impact Quantification*

In the following experiments, we compare model performance when different VPP components are removed. In particular, we compare the following cases:

1. Full VPP (a VPP with all components, as defined in Section 2.2)

2. Without ESS

3. Without DR

Note that "without DR" means that demand response is removed from the model, but load shedding can still be utilized. Figure 2.7 compares the average cost across 50 simulations for three

different cases.

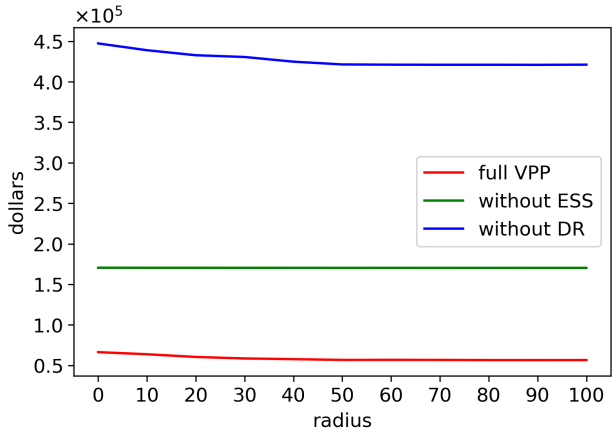
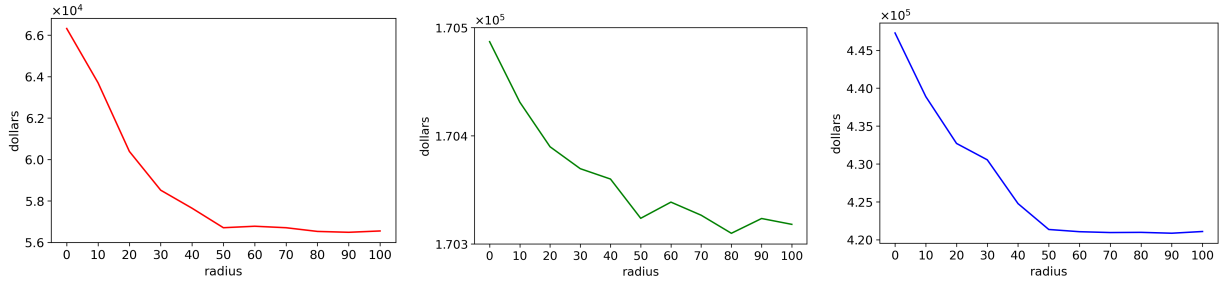



Figure 2.7: Comparing cost when ESS or DR is removed

It is clear that Case 1 (full VPP) yields the best performance, followed by Case 2 (without ESS), and then Case 3 (without DR) performs the worst. These results emphasize the importance of demand response as an effective way to reduce the load when demand exceeds supply. Without DR, the system must rely on load shedding, which is much more expensive. The results in Figure 2.7 indicate that ESS also plays an important role in reducing costs. The ESS can charge and discharge energy, which gives the VPP flexibility in scenarios with high or low net loads. However, due to the ESS's capacity constraints, it is unable to mitigate peak load scenarios as effectively as DR.

We can further explore how the model performs by looking at Case 2 and Case 3 separately. Figures 2.8a, 2.8b and 2.8c show the average cost with respect to radius for Cases 1, 2, and 3, respectively for $r = 0$ to 100.




(a) Full VPP (b) Without ESS (c) Without DR

Figure 2.8: Average cost with different component combinations

Figures 2.8a, 2.8b and 2.8c all show a negative correlation, which means that considering more uncertainty in the planning process improves the performance in all 3 cases. However, as radius increases from 0 to 100, we observe a much more dramatic change in cost for Case 3 ($\sim \$30,000$) than in Case 1 ($\sim \$10,000$). Thus, in the absence of DR, there is a greater performance improvement when more distributional robustness is incorporated, compared to the full VPP. On the other hand, when $r$ is increased from 0 to 100, we see a less dramatic change in cost for Case 2 ($\sim$ \$200). Therefore, after removing ESS, the model performance improves to a lesser degree compared to the full VPP, as more distributional robustness is incorporated.

While both ESS and DR play important roles in mitigating costs for the VPP, DR plays a more significant role. Without DR, the model relies more on increasing distributional robustness to achieve higher performance compared to when ESS is removed.

### *2.4.4 Scalability Analysis*

*Extended Time Horizon*

To evaluate the scalability of our DRDP algorithm, we conducted experiments on extended time horizons beyond the previously tested 24 hours. Specifically, we tested horizons ranging from 1 to 4 weeks, equivalent to 168 to 672 hours, with hourly decision intervals. The objective was to measure the execution time required for a single simulation, encompassing both the training and testing phases. In our algorithm, the number of iterations in each forward and backward path run is directly related to the size of the time horizon. As a result, we anticipated observing a linear relationship between the horizon length and the execution time. Empirical evidence supporting this

trend is presented in Table 2.6, which displays the results of our experimental analysis.

| horizon (weeks) | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| runtime (seconds) | 56.71 | 70.52 | 101.22 | 130.07 |

Table 2.6: Relationship between horizon and runtime

Our results demonstrate that our methodology is capable of accommodating large time horizons, providing decision-makers with the ability to make sequential decisions further into the future or with narrower decision intervals, surpassing the limitations of the typical 24-hour setting.

Moreover, it is important to note that the runtime reported in Table 2.6 represents the total time required to run an entire simulation, including both the training and testing procedures for the entire time horizon. In practice, only the training procedure would need to be executed prior to the start of the horizon. Subsequently, at each hour, the decision-maker would only need to solve the distributionally robust Bellman's equation (2.26) for that specific time period. Since the computational time is linearly dependent on the horizon length, it is extremely efficient and can be anticipated to solve in a much shorter time.

*Test Case on South Carolina 500-bus Power System Model*

We also tested our algorithm on the synthetic South Carolina 500-bus power system model [53] to see how increasing the number of generators affects the model performance. We used the same experimental setup as previously. Across all simulations run with this dataset, the average runtime for the DRDP algorithm was 48 seconds. Even for large-scale data, our model runs efficiently because the reformulation method used to derive the algorithm relies only on solving linear programs for each simulation. These results show that our methodology can efficiently handle problems with an increased number of generators in addition to increasing the time horizon. We can also observe the average profit over 100 simulations with different radius values in Table 2.7.

| radius | 0 | 25 | 50 | 75 | 100 |
|---|---|---|---|---|---|
| profit | 1,425,803.54 | 1,426,324.71 | 1,427,070.09 | 1,427,967.58 | 1,428,727.00 |

Table 2.7: Increased profit as radius increases

We see a similar trend to what we observed previously. As the radius increases, we see an increase in profit because the system can account for more uncertainty with larger radius values. Note that the South Carolina 500-bus system allows the system to earn a profit instead of incurring a cost, which was observed in the previous setting. From a mathematical modeling perspective, this is just a difference between a positive and negative objective value. Therefore, our methodology can improve performance by incorporating uncertainty in the decision-making process regardless of the sign of the objective function.

### *2.5 Conclusion*

Our MSDRO method has four major benefits. The first is that it hedges against uncertainty by providing more conservative solutions than SO approaches. This is important for VPPs because extreme events can be costly. The second advantage is that, unlike RO methods, it provides solutions that are not overly-conservative by optimizing over the worst-case distribution rather than a single worst-case scenario. This property enables our model to hedge against uncertainty without overspending to prepare for a single extreme scenario that may occur rarely in practice. Third, unlike approaches that use SO or CVaR, our MSDRO approach does not assume the true distribution is known, which is important because wind output, solar output, demand, and market price distributions are difficult to estimate accurately. Lastly, utilizing a dynamic multi-stage framework allows the system to incorporate the constant influx of information to support real-time decision-making.

Our findings revealed that considering the appropriate amount of distributional ambiguity allowed us to reduce the average cost and variance. Even with incomplete distributional information, the MSDRO method provided a solution that is reliable, cost-effective, and robust to more extreme realizations of our random events. We also showed the benefit of incorporating a dynamic framework, which adjusts the dispatch decisions in a real-time manner after the values of uncertain parameters are realized. These advantages allowed our model to outperform the static model that had to commit to a solution 24 hours in advance before realizing any uncertain parameter values. Furthermore, our component impact analysis quantified the impact of DR and ESS on the VPP's effectiveness. These experiments demonstrated the relative abilities of ESS and DR to handle extreme demand cases. Finally, we conducted experiments that demonstrate our methodology's scalability by evaluating its ability to efficiently solve problems with large time horizons, and large-scale systems.

Chapter 3

# A TWO-STAGE TARGET-ORIENTED ROBUST OPTIMIZATION MODEL TO ASSESS THE FLEXIBILITY OF VIRTUAL POWER PLANTS WITH QUICK-START STORAGE RESOURCES

## *3.1 Background*

There are many degrees of stochasticity associated with VPPs, which makes them challenging to operate reliably. The VPP can satisfy random customer demand on the distribution level and provide ancillary services to the transmission level based on power trajectories provided by transmission system operators (TSOs) [54]. The VPP also relies on volatile renewable energy penetration [13], which adds another degree of stochasticity. To address this challenge, we propose a flexibility framework that helps the VPP assess its ability to reliably satisfy customer demand and power trajectories from TSOs while utilizing renewable energy.

In the literature, flexibility is defined in various ways. One category of flexibility metrics is stochastic flexibility. These metrics assess the probability of violating system constraints or they optimize the average flexibility based on specific probability distributions. For instance, the insufficient ramping resource expectation (IRRE) quantifies the expected number of observations when a power system cannot cope with the changes in renewable generation or demand [55]. The loss of wind estimation (LOWE) index represents the estimated probability of wind curtailment occurring in a system within a year [56]. Lack of ramp probability (LORP) represents the probability that the available system ramping capability from dispatched generators will not satisfy the load [57]. One limitation of these metrics lies in their probabilistic nature. They cannot guarantee the fulfillment of the demand without any violations because they solely ensure load satisfaction in terms of expectation or probability. Moreover, studies that employ simulation methods, like [58], for assessing system flexibility, also lack the assurance of load satisfaction. Similarly, [59] and [60] utilize stochastic optimization techniques to minimize the expected total cost with the integration of flexible resources, such as energy storage, into microgrid optimization problems. However, these methodologies may not provide robust solutions that cover all scenarios reliably.

Other methods define flexibility as a function of various risk metrics. By incorporating risk in their flexibility definition, these papers aim to create a conservative flexibility metric that hedges

against uncertainty. For example, operational risk is a flexibility metric that includes expected operational loss for wind generation curtailment as well as load shedding [61]. The expected flexibility shortfall is the conditional expectation of load loss due to insufficient flexibility, given that it is beyond the Value-at-Risk level of flexibility [62]. These papers present risk-averse flexibility metrics by incorporating risk in their flexibility definition. However, risk is an inherently abstract concept, so risk-based flexibility metrics could be difficult to understand and interpret in practice.

In contrast to the metrics described above, we use a reliable and functional flexibility definition: *flexibility is the largest amount of deviation from the expected required non-renewable generation across all time periods that the system can tolerate.* Required non-renewable generation (RNRG) is defined as the difference between the customer demand and power trajectories requested by TSOs and the expected renewable generation level. This flexibility metric helps the VPP satisfy RNRG more reliably by measuring the deviations from the expected RNRG it can satisfy.

It is a reliable flexibility definition because it guarantees that the RNRG will be satisfied surely instead of probabilistically or on average as in [55]-[60]. Furthermore, it is functional because it is easy to understand and interpret, and can be communicated more easily to practitioners compared to the more abstract risk flexibility definitions in [61] and [62]. Another advantage of our flexibility metric is its ability to model complex systems like VPPs which have multiple heterogenous generation resources and multiple forms of stochasticity.

To model flexibility, we use a formulation that is similar to two-stage robust optimization (RO) [63], [64], and [65]. These methods minimize cost while assuming the random system components take the worst-case value within an uncertainty set. We use a two-stage target-oriented robust optimization (TORO) method [66], which has one key difference, however. In our formulation TORO, we restrict the cost to be lower than a pre-specified target cost, and the size of the uncertainty set is a decision variable in the objective instead of a fixed parameter in the constraints as it is in standard two-stage RO methods. Therefore, our model does not have to accurately measure the boundaries of the possible random outcomes, which is challenging, especially when random variables are volatile. Thus, our method is easier to implement in practice compared to two-stage RO methods.

A similar definition and formulation of flexibility has been studied by [67], [68], and [69], however, they do not consider second-stage integer variables. We include second-stage integer variables, which allow us to model quick-start resources such as storage units more accurately. Adding second-stage integer variables makes the problem more difficult to solve because the duality-based approaches used in [68] and [69] are no longer applicable. To the best of our knowledge, we are the first to

evaluate flexibility with second-stage binary decision variables. We formulate our problem as a two-stage optimization problem with second-stage integer variables, and we use a variable uncertainty set to model the VPP's flexibility.

A similar formulation with second-stage integer variables and a variable uncertainty set is studied in [70], although they don't directly optimize flexibility. They propose an algorithm based on the column-and-constraint-generation (CCG) approach presented in [71] that requires solving mixed-integer subproblems. We solve our model with a novel primal generation algorithm (PGA) after reformulating it as a mixed-integer bilinear program, which outperforms the aforementioned CCG method. Some advantages that contribute to our method's efficiency are that it does not rely on auxiliary binary variables and it has a faster termination criteria compared to typical CCG methods. We verify our algorithm's efficiency by conducting experiments that showcase our algorithm's ability to evaluate VPP flexibility more efficiently than a standard CCG approach. We summarize our main contributions as the following:

1. We assess the flexibility of a VPP by formulating a robust two-stage uncertainty-set optimization problem with second-stage integer variables.

2. We propose a novel PGA that solves our model efficiently while maintaining a small optimality gap by utilizing a bilinear search heuristic and leveraging the problem structure to reduce the convergence time.

3. We conduct experiments to demonstrate our PGA's ability to evaluate the flexibility of a VPP, and its ability to outperform standard CCG methods.

The rest of this chapter is organized as follows. In Section 3.2, we present the VPP flexibility formulation. We present an abstract formulation for notational brevity in Section 3.3. In Section 3.4 we present the model reformulation technique used to transform our problem into a structure that can be solved by our PGA, presented in Section 3.5. Section 3.6 discusses our experimental results. Section 3.7 concludes the chapter.

### *3.2 Flexibility Model Formulation*

We consider the unit commitment and economic dispatch setting where the VPP's goal is to decide which generation resources to use to supply enough energy to satisfy random customer demand. In

this section, we formulate the VPP flexibility problem which we refer to as the original problem (OP). Our VPP consists of distributed generation units, demand response (DR), energy storage systems (ESS), and wind and solar generation resources. We present the entire OP formulation in this section. The objective of the OP is to evaluate the VPP's flexibility.

We evaluate the flexibility with respect to two stages of constraints. The first-stage constraints describe the on/off status decisions for the distributed generators that the VPP commits to. Once the first-stage decisions are fixed, the second-stage constraints describe feasible dispatch actions for all of the VPP components.

The uncertainty set $U_\gamma$ represents the set of RNRG values that can be fulfilled by using the VPP's non-renewable generation resources without violating operational constraints. We represent the RNRG as $\xi := d + d' - s_w - s_p$ where $d, d', s_w, s_p$ represent the random customer demand, power trajectory, wind output, and photovoltaic output, respectively.

$$U_\gamma := \bigtimes_{t=1}^{T} [\hat{\xi}_t - \gamma_t, \hat{\xi}_t + \gamma_t] \tag{3.1}$$

Here, $\bigtimes_{t=1}^{T}$ represents the Cartesian product of sets across all time periods $t = 1, ..., T$. So $U_\gamma$ is the Cartesian product of each interval $[\hat{\xi}_t - \gamma_t, \hat{\xi}_t + \gamma_t]$ across all periods in our time horizon. $\hat{\xi} \in \mathbb{R}^T$ is the expected RNRG, which is the center of the uncertainty set at each time $t$. The decision variable $\gamma$ represents flexibility by describing the deviation from $\hat{\xi}$ that the VPP can satisfy with its non-renewable generation resources. The goal of the OP is to find the largest such $\gamma$ such that the VPP can feasibly satisfy $\forall \xi \in U_\gamma$. Thus, for some expected RNRG value, $\hat{\xi}$, the OP outputs the maximum allowable deviation from $\hat{\xi}$ the VPP can tolerate.

### *3.2.1 Objective Function and Demand Balance Constraints*

$$\max \sum_{t=1}^{T} \gamma_t \tag{3.2}$$

$$\text{s.t.} \sum_{n \in G} p_n^t + \sum_{s \in \mathcal{S}} \left( g_{st}^{(-)} - g_{st}^{(+)} \right) + \mu_t = \xi_t, \tag{3.3}$$

$$\forall \xi_t \in [\hat{\xi}_t - \gamma_t, \hat{\xi}_t + \gamma_t], \forall t \in [T]. \tag{3.4}$$

The objective function (3.2) is equal to the flexibility of the VPP, which is defined as the sum of $\gamma_t$ across all time periods. Thus, the objective represents the maximum allowable deviation from the $\hat{\xi}$ the system can tolerate. We illustrate another way to understand the objective in Figure 3.1,

which represents a case where the time horizon $T = 2$. The purple rectangle of dimensions $\gamma_1$ and $\gamma_2$ represents an uncertainty set $U_\gamma$. It can be thought of as a "flexibility region" that is a hyperbox centered at $\hat{\xi}$ such that every value within the hyperbox can be satisfied by the VPP. This flexibility region is an approximation of the true feasibility boundary represented by the yellow line in the figure, where a point is feasible if and only if it is within this boundary. Our objective, then, is to maximize the side lengths $\gamma_1, ..., \gamma_T$ of such a hyperbox.

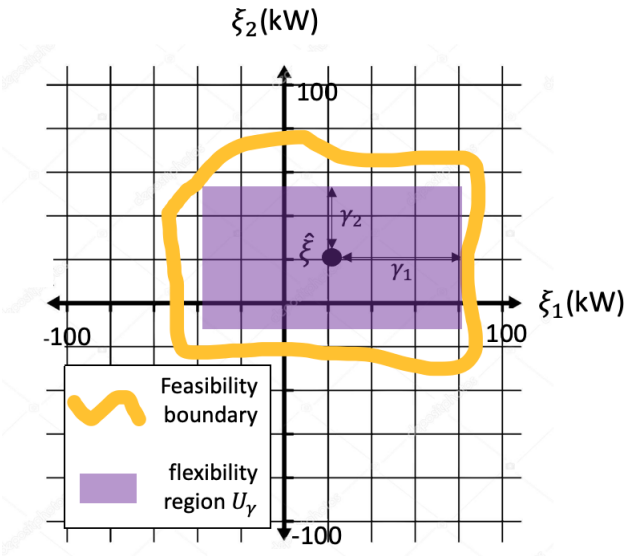



Figure 3.1: Flexibility region

Constraint (3.3) says that the RNRG must equal output from non-renewable resources for all possible values of $d$ within our uncertainty set (3.4). Specifically, the left-hand side of (3.3) includes distributed generation output $p_n^t$, net ESS charge $g_{st}^{(-)} - g_{st}^{(+)}$, and demand response $\mu_t$, which must equal the $\xi_t$ on the right-hand side. This demand balance must be satisfied for all $\xi_t$ in our uncertainty set $[\hat{\xi}_t - \gamma_t, \hat{\xi}_t + \gamma_t]$ for all time periods $t \in [T]$ according to (3.4), where $[T] := 1, ..., T$.

### *3.2.2 First-stage Constraints*

The first-stage problem consists of the unit commitment decisions for the distributed generators, which describe the binary on/off decisions for the generators.

$$x_n^t + z_n^t \leq 1, \forall n \in \mathcal{G}, \forall t \in [T], \tag{3.5}$$

$$y_n^t - y_n^{t-1} = x_n^t - z_n^t, \forall n \in \mathcal{G}, \forall t \in [T], \tag{3.6}$$

$$\sum_{\tau=t-T_n^{\text{minu}}+1}^{t} x_n^\tau \leq y_n^t, \forall n \in \mathcal{G}, \forall t \in \{T_n^{\text{minu}}, \cdots, T\}, \tag{3.7}$$

$$\sum_{\tau=t-T_n^{\text{mind}}+1}^{t} z_n^\tau \leq 1 - y_n^t, \forall n \in \mathcal{G}, \forall t \in \{T_n^{\text{mind}}, \cdots, T\}, \tag{3.8}$$

$$x_n^t, y_n^t, z_n^t, \in \{0, 1\}, \forall n \in \mathcal{G}, \forall t \in [T]. \tag{3.9}$$

$x_n^t$ and $z_n^t$ are start-up and shut-down indicator variables for distributed generator $n$ and time $t$, respectively. $y_n^t$ is an on-off indicator variable for distributed generator $n$ at time $t$. $T_n^{mind}$ and $T_n^{minu}$ are the minimum down and up time for generator $n$, respectively. (3.5) says that a generator cannot be turned on and turned off at the same time. (3.6) enforces the relationship between the on/off status and the decision to turn a generator on or off. (3.7) and (3.8) enforce the minimum up and down time, respectively.

### *3.2.3 Second-stage Constraints*

The second-stage constraints describe feasible dispatch actions for all of the VPP components based on the on-off decisions made for the distributed generators in the first stage.

*Budget Constraint*

$$\sum_{t=1}^{T} \left( \sum_{n \in \mathcal{G}} \text{SU}_n x_n^t + \text{SD}_n z_n^t + c_n^t + \sum_{l=1}^{L} \pi_{lt} \mu_{lt} \right) \leq \phi, \tag{3.10}$$

$T$ is the time horizon, $\mathcal{G}$ is the set of generators, and $L$ is the number of demand response price levels. $SU_n$ and $SD_n$ are the start-up and shut-down costs for generator $n$, respectively. $c_n^T$ is the fuel cost for generator $n$ at time $t$. $\pi_{lt}$ is the price of demand response within interval $l$ and $\mu_{lt}$ is the amount of demand response used in interval $l$ at time $t$. $\phi$ is the allowable budget. (3.10) says that the total VPP operating cost must satisfy the pre-determined target budget $\phi$. The left-hand side of (3.10) consists of the start-up cost, shut-down cost, distributed generation fuel cost, and demand response cost. The start-up and shut-down are fixed costs incurred any time a distributed generator is turned on or off, respectively. The fuel cost for the distributed generators is a quadratic function approximated as a piecewise linear function in (3.11) [18]. Lastly, the cost of implementing demand

response is related to the amount of demand change, so we adopt a step-wise DR cost curve [45], where $\pi_{lt}$ and $\mu_{lt}$ are the price and amount of demand response, respectively, at level $l$ and at time $t$.

*Distributed Generation Constraints*

$$c_n^t \geq \alpha_n^k p_n^t + \beta_n^k y_n^t \quad \forall n \in \mathcal{G}, \forall k \in [K] \;\; \forall t \in [T], \tag{3.11}$$

$$p_n^t - p_n^{t-1} \leq RU_n y_n^{t-1} + \overline{RU}_n x_n^t, \forall n \in \mathcal{G}, \forall t \in [T], \tag{3.12}$$

$$p_n^{t-1} - p_n^t \leq RD_n y_n^t + \overline{RD}_n z_n^t, \forall n \in \mathcal{G}, \forall t \in [T], \tag{3.13}$$

$$p_n^t \leq P_n^{\max} y_n^t, \forall n \in \mathcal{G}, \forall t \in [T], \tag{3.14}$$

$$p_n^t \geq P_n^{\min} y_n^t, \forall n \in \mathcal{G}, \forall t \in [T]. \tag{3.15}$$

$\alpha_n^k$ and $\beta_n^k$ are quadratic fuel cost approximation parameters, where we consider $K$ breakpoints for the piecewise approximation. $P_n^{max}$ and $P_n^{min}$ are maximum and minimum generation limits for generator $n$, respectively. $RU_n$ and $RD_n$ are the maximum ramp-up and ramp-down rates for generator $n$, respectively. $\overline{RD}_n$ and $\overline{RD}_n$ are the maximum start-up ramp-up and shut-down ramp-down rates for generator $n$, respectively. $p_n^t$ is the generation level from generator $n$ at time $t$. (3.12)-(3.15) describe the physical constraints of the distributed generation resources. Constraints (3.12) and (3.13) represent the ramp-up and ramp-down constraints, respectively. (3.14) and (3.15) describe upper and lower capacity limits, respectively.

*Demand Response Constraints*

$$0 \leq \mu_{lt} \leq R_{t,l} - R_{t,l-1}, \forall t \in [T], l \in [L], \tag{3.16}$$

$$\mu_t = \sum_{l}^{L} \mu_{lt}, \forall t \in [T], \tag{3.17}$$

$$\mu_t \leq DR_{max}, \forall t \in [T]. \tag{3.18}$$

Demand response allows the VPP to reduce customer demand, which allows the VPP's other generation resources to satisfy the power trajectory from the transmission system more easily. $\mu_{lt}$ is the amount of demand response used in the interval between $R_{t,l}$ and $R_{t,l-1}$, which is captured by (3.16). $\mu_t$ is the total demand response amount, and $DR_{max}$ is the maximum allowable demand response. (3.17) establishes the relationship between the total demand response and the demand

response used at each level, and (3.18) ensures that the total demand response cannot exceed the demand response limit for all time periods.

*Energy Storage System Constraints*

$$h_s^1 = h_s^0 \quad \forall s \in \mathcal{S}, \tag{3.19}$$

$$h_s^{t+1} = h_s^t + \nu_s^{(+)} g_{st}^{(+)} - \nu_s^{(-)} g_{st}^{(-)}, \forall t \in [T], \forall s \in \mathcal{S}, \tag{3.20}$$

$$h_s^t \leq \bar{H}_s \quad \forall t \in [T], \forall s \in \mathcal{S}, \tag{3.21}$$

$$g_{st}^{(+)} \leq \bar{G}_s^{(+)} \sigma_{st}^{(+)} \quad \forall t \in [T], \forall s \in \mathcal{S}, \tag{3.22}$$

$$g_{st}^{(-)} \leq \bar{G}_s^{(-)} \sigma_{st}^{(-)} \quad \forall t \in [T], \forall s \in \mathcal{S}, \tag{3.23}$$

$$\sigma_{st}^{(+)} + \sigma_{st}^{(-)} \leq 1 \quad \forall t \in [T], \forall s \in \mathcal{S}. \tag{3.24}$$

The ESS can charge or discharge energy. Here, $\mathcal{S}$ is the set of all ESS units. $h_s^t$ is the state of charge at time $t$ for ESS unit $s$, and $h_s^0$ is the initial storage level of ESS $s$. $g_{st}^{(+)}$ and $g_{st}^{(-)}$ are the charging and discharging level for ESS $s$ at time $t$, respectively. $\bar{H}_s$ is the storage capacity of ESS $s$. $\bar{G}_s^{(+)}$ and $\bar{G}_s^{(-)}$ are the charging and discharging limits for ESS $s$, respectively. $\nu_s^{(+)}$ and $\nu_s^{(-)}$ are the respective charging and discharging efficiency for ESS $s$. $\sigma_{st}^{(+)}$ and $\sigma_{st}^{(-)}$ are binary decision variables that indicate whether the battery is charging or discharging. (3.19) represents the initial charge of the ESS at the beginning of the time horizon. (3.20) represents the charge balance constraints. (3.21),(3.22), and (3.23) represent the storage, discharging, and charging capacities for the ESS, respectively. (3.24) says that the ESS cannot discharge and charge simultaneously. Constraints (3.22)-(3.24) introduce the binary variables $\sigma_{st}^{(+)}$ and $\sigma_{st}^{(-)}$ into the second-stage problem.

Finally, we have the following constraints (3.25) and (3.26) to define the non-negative continuous variables and integer variables, respectively.

$$p_n, dr_t, \mu_{lt}, g_{st}^{(+)}, g_{st}^{(-)} \geq 0, \tag{3.25}$$

$$\sigma_{st}^{(+)}, \sigma_{st}^{(-)} \in \{0, 1\}, \tag{3.26}$$

$$\forall n \in \mathcal{G}, \forall t \in [T], \forall s \in \mathcal{S}, \forall l \in [L].$$

Moreover, note that in our TORO formulation, the size of the uncertainty set is a decision variable and not a fixed parameter in the constraints like in standard two-stage (RO) methods [63], [64], and [65]. Therefore, our model does not have to accurately measure the limits of possible random renewable generation scenarios, which can be difficult in practice.

### *3.3 Abstract Formulation*

For notational brevity, we present an abstract formulation for the above VPP flexibility problem to be used for the rest of the chapter. We let $w$ be a vector of first-stage binary decision variables $x, y$, and $z$. We use $u$ and $q$ to represent second-stage continuous and integer decision variables, respectively. $A$ and $b$ are the constraint matrix and right-hand-side for the first-stage binary decision variables, respectively. $B$ represents the constraint matrix for binary decision variables in the second stage. We let $C$ and $G$ be the constraint matrices of second-stage continuous and integer decision variables, respectively. $H$ is a constraint matrix for demand. Lastly, we let $h$ be the constants on the right-hand side of the second-stage constraints. Our overall objective is to find the largest uncertainty set where we can find a $w$ in the first stage that leads to a feasible second-stage decision $q$ and $u$ for all $d$ in the uncertainty set. In other words, the OP is:

$$\max \sum_{t=1}^{T} \gamma_t \tag{3.27a}$$

$$\text{s.t.} \quad Aw \leq b, \tag{3.27b}$$

$$Bw + Cu + Gq \leq H\xi + h, \tag{3.27c}$$

$$\forall \xi \in [\hat{\xi} - \gamma, \hat{\xi} + \gamma]. \tag{3.27d}$$

The objective (3.27a) is the same as (3.2). (3.27b) is the first-stage operational constraints (3.5)-(3.9). (3.27c) is the second-stage operational constraints (3.3), and (3.10)-(3.26). Finally, (3.27d) is the uncertainty set (3.4), which says that we must consider all possible scenarios, $d$, within a deviation of $\gamma$ away from $\hat{\xi}$. Without loss of generality, we assume that all constraints are inequality constraints.

One advantage of our TORO formulation is that the size of the uncertainty set is a decision variable and not a fixed parameter in the constraints like in standard two-stage (RO) methods [63], [64], and [65]. Therefore, our model does not have to accurately measure the limits of possible random renewable generation scenarios, which can be difficult in practice.

However, there are a few properties of the model that make it challenging to solve. The binary variables in the second-stage problem make it challenging to solve because duality approaches that work for continuous two-stage optimization problems are no longer applicable.

We can see that the OP (3.27) is a bilevel semi-infinite program with second-stage integer

variables, making it a challenging problem to solve because the discrete variables make it difficult to use duality approaches. Duality-based methods such as [68] and [69] are common for solving two-stage optimization problems, but they rely on strong duality which only holds for a continuous feasible region in the second stage.

Therefore, these methods cannot be applied to our problem, motivating our PGA, which can efficiently solve our problem formulation with second-stage integer variables. In the next section, we present the techniques used to reformulate to allow it to be solved by our PGA.

## *3.4 Model Reformuation*

### *3.4.1 Overall Flow*

Before we introduce the details of how to solve the OP, we first outline the overall flow of the proposed method:

1. Solve MP for first-stage variables $\tilde{w}, \tilde{\gamma}$
2. Solve the Feasibility Check Problem using the PGA to check if $\tilde{w}, \tilde{\gamma}$ leads to a feasible second-stage problem
   (a) if we have solved the Feasible Check Problem $N$ times, terminate and return $\tilde{w}, \tilde{\gamma}$
   (b) if feasible, terminate the algorithm. $\tilde{w}, \tilde{\gamma}$ is optimal
   (c) if not feasible, add feasibility cut constraints to MP and return to Step 1

$N$ represents the maximum number of times the Feasibility Check Problem is solved. If $N = \infty$, then step 2a is nullified.

### *3.4.2 Master Problem*

To solve the OP, we first solve the following Master Problem (MP) to find an initial first-stage solution, which we denote $\tilde{\gamma}$ and $\tilde{w}$.

$$\max \sum_t \gamma_t \tag{3.28a}$$

$$\text{s.t.} \quad Aw \leq b, \tag{3.28b}$$

$$\sum_t \gamma_t \leq M. \tag{3.28c}$$

Solving the MP is equivalent to finding the largest uncertainty set size $\gamma$ while only considering first-stage constraints. (3.28c) ensures that the MP (3.28) is never unbounded by choosing some large $M$ because initially, there are no feasibility cut constraints, which are constraints added later in the process.

### *3.4.3 Feasibility Check Problem*

Next, we check if there exists a feasible second-stage solution or not, given that $\tilde{\gamma}$ and $\tilde{w}$ are chosen in the first stage. We do this by solving the Feasibility Check Problem (FCP):

$$\psi(\tilde{w}, \tilde{\gamma}) := \max_{\xi \in U_{\tilde{\gamma}}} \min_{q,u,v} v \tag{3.29a}$$

$$\text{s.t.} \quad Cu + Gq - v\mathbf{1} \leq H\xi + h - B\tilde{w}, \tag{3.29b}$$

$$v \geq 0. \tag{3.29c}$$

where $U_{\tilde{\gamma}} = \bigtimes_{t=1}^{T} [\hat{\xi}_t - \tilde{\gamma}_t, \hat{\xi}_t + \tilde{\gamma}_t]$. The FCP takes $\tilde{w}$ and $\tilde{\gamma}$ as inputs and determines if there is a second-stage constraint violation for some $\xi \in U_{\tilde{\gamma}}$. We introduce the decision variable $v \in \mathbb{R}$ to represent the amount of violation of the second-stage constraints (3.29b). The objective consists of an inner-minimization problem and an outer-maximization problem. The objective of the inner-minimization problem is to find a solution that minimizes the amount of constraint violation represented by variable $v$ for a fixed value $\xi$. The outer-maximization problem's objective is to find some value $d$ in the uncertainty set $U_{\tilde{\gamma}}$ that maximizes the objective of the inner-minimization problem. Thus, the objective function (3.29a), denoted $\psi(\tilde{w}, \tilde{\gamma})$, can be used to determine if there exists some $d \in U_{\tilde{\gamma}}$ that leads to an infeasible second-stage problem. We can use the value of $\psi(\tilde{w}, \tilde{\gamma})$ to determine whether $\tilde{w}$ and $\tilde{\gamma}$ are feasible for all $d$ in the uncertainty set. Specifically, the following will hold:

$$\begin{cases} \psi(\tilde{w}, \tilde{\gamma}) = 0 & \iff \tilde{w}, \tilde{\gamma} \text{ is feasible} \\ \psi(\tilde{w}, \tilde{\gamma}) > 0 & \iff \tilde{w}, \tilde{\gamma} \text{ is infeasible} \end{cases} \tag{3.30}$$

If $\psi(\tilde{w}, \tilde{\gamma}) = 0$, then the objective of (3.29) is 0, so the optimal value of the decision variable $v$ is 0. $v = 0$ indicates that there is no violation of the second-stage constraints (3.29b), and therefore, the decision of $\tilde{w}$ and $\tilde{\gamma}$ in the first-stage leads to a feasible second-stage problem, and we say "$\tilde{w}, \tilde{\gamma}$

is feasible". Otherwise, $\psi(\tilde{w}, \tilde{\gamma}) > 0$, so the optimal value of $v$ is positive, indicating that there is a violation of the second-stage constraints (3.29b). This means the first-stage decision $\tilde{w}$ does not yield a feasible second-stage solution. In this case, we say that $\tilde{w}$ and $\tilde{\gamma}$ are infeasible.

We see that the FCP (3.29) is a robust optimization problem by the min/max objective (3.29a). These problems are typically solved with duality approaches, but these methods will not work because of the existence of second-stage integer variables. We propose our PGA to handle such a problem, but in order to utilize the PGA, we must separate the continuous and integer second-stage variables and eliminate the min/max objective structure. To accomplish this, the first step is to reformulate the FCP as a 3-stage problem by separating the binary and continuous variables to split up the inner minimization problem into two distinct minimization problems:

$$\max_{\xi} \min_{q \in \Xi} \min_{u,v} v \tag{3.31a}$$

$$\text{s.t.} \quad \hat{\xi} - \tilde{\gamma} \leq \xi \leq \hat{\xi} + \tilde{\gamma}, \tag{3.31b}$$

$$Cu - v\mathbf{1} \leq H\xi + h - B\tilde{w} - Gq, \tag{3.31c}$$

$$v \geq 0. \tag{3.31d}$$

where $\Xi$ is the feasible region for discrete variables, $q$. In our VPP setting, $q$ represents the binary charge/discharge decisions for the ESS, so $\Xi$ is a finite set if we consider a finite number of ESS units. The inner-most minimization problem is a linear program (LP) because it has only continuous decision variables $u$ and $v$, so duality methods are applicable. We denote this inner-most minimization problem below as (3.32) for conciseness, which is an optimization problem parameterized by $q$ and $\xi$, with objective value $\theta(q, \xi)$.

$$\theta(q, \xi) := \min_{u,v} v \tag{3.32a}$$

$$\text{s.t.} \quad Cu - v\mathbf{1} \leq H\xi + h - B\tilde{w} - Gq, \tag{3.32b}$$

$$v \geq 0. \tag{3.32c}$$

Now, (3.31) can be represented as (3.33)

$$\max_{\xi} \min_{q} \theta(q, \xi) \tag{3.33a}$$

$$\text{s.t.} \quad \hat{\xi} - \tilde{\gamma} \leq \xi \leq \hat{\xi} + \tilde{\gamma}. \tag{3.33b}$$

We further reformulate the FCP (3.34) by introducing an auxiliary decision variable $\eta$ to represent the value of $\min_q \theta(q, \xi)$. The new objective function is to maximize the value of $\eta$ while upper-bounding it by each of the finitely many possible values of $\theta(q, \xi)$ for all $q \in \Xi$.

$$\max_{\eta, \xi} \eta \tag{3.34a}$$

$$\text{s.t.} \quad \eta \leq \theta(q, \xi) \quad \forall q \in \Xi, \tag{3.34b}$$

$$\hat{\xi} - \tilde{\gamma} \leq \xi \leq \hat{\xi} + \tilde{\gamma}. \tag{3.34c}$$

Now, the objective function is solely a maximization problem without the min/max structure from the original formulation, 3.29. Moreover, (3.34) is a bilevel optimization problem because the constraints include $\theta(q, \xi)$, which is the objective of a distinct optimization problem. Recall from (3.32), the objective function of (3.32) is to minimize the scalar decision variable $v$. Therefore, $\theta(q, \xi)$ is the optimal value of the decision variable $v$, which we denote $v^*$.

The LP (3.32) and its dual are feasible since (3.32) is feasible and bounded. (3.32) is feasible because there always exists some non-negative $v$ to satisfy (3.32b), and it is bounded because the objective is to minimize the value of $v$ and $v \geq 0$. Therefore, $v^*$ is the $v$ that satisfies the primal feasibility, dual feasibility, and strong duality constraints, so $\theta(q, \xi)$ is equal to the value of the decision variable $v$ that satisfies the constraints:

$$Cu - v\mathbf{1} \leq H\xi + h - B\tilde{w} - Gq, \tag{3.35a}$$

$$\lambda^T C \leq 0, \tag{3.35b}$$

$$\lambda^T \mathbf{1} \geq -1, \tag{3.35c}$$

$$\lambda^T (H\xi + h - B\tilde{w} - Gq) = v, \tag{3.35d}$$

$$\lambda \leq 0. \tag{3.35e}$$

Here, $\lambda$ is a dual decision variable, (3.35a) are primal constraints, (3.35b) and (3.35c) are dual constraints, and (3.35d) is a strong duality constraint.

We can obtain an equivalent reformulation of the FCP by replacing $\theta(q, \xi)$ in (3.34b) with $v$ and adding the constraints (3.35) for each possible discrete element in the set $\Xi$:

$$\psi(\tilde{w},\tilde{\gamma}) = \max_{\eta,\xi,\lambda,v} \tag{3.36a}$$

$$\text{s.t.} \quad \eta \leq v_r, \tag{3.36b}$$

$$Cu^r - v_r\mathbf{1} \leq H\xi + h - B\tilde{w} - Gq^r, \tag{3.36c}$$

$$(\lambda^r)^T C \leq 0, \tag{3.36d}$$

$$(\lambda^r)^T\mathbf{1} \geq -1, \tag{3.36e}$$

$$(\lambda^r)^T(H\xi + h - B\tilde{w} - Gq^r) = v_r, \tag{3.36f}$$

$$\hat{\xi} - \tilde{\gamma} \leq \xi \leq \hat{\xi} + \tilde{\gamma}, \tag{3.36g}$$

$$v \geq 0, \lambda^r \leq 0, \tag{3.36h}$$

$$\forall r = 1, ..., |\Xi|.$$

Here, $q^r$ is a constant that represents the $r^{th}$ element in $\Xi$. Therefore (3.36) has one set of constraints of the form (3.36b)-(3.36f) for every element in $\Xi$. Since $\Xi$ is the set equal to every possible second-stage integer variable, $|\Xi|$ could be large, making (3.36) challenging to solve directly. Furthermore, (3.36) can not be solved with LP solvers because of the nonlinearity caused by the bilinear term, $\lambda^T H\xi$, in constraint (3.36f). In the next section, we propose our PGA, which can efficiently solve the above problem.

## *3.5 Primal Generation Algorithm*

In this section, we introduce the details of the PGA, which solves the FCP (3.36), by which iteratively generates constraints of the form (3.36b)-(3.36f) to the primal problem. Before we introduce the algorithm, we define two problems that are used in the algorithm: the feasibility sub-problem and feasibility master problem, which provide lower and upper bounds of the FCP, respectively.

### *3.5.1 Feasibility sub-problem*

The feasibility sub-problem (FSP) is the same as the FCP but for a fixed $\xi = \tilde{\xi}$:

$$\underline{\psi}(\tilde{w}, \tilde{\gamma}, \tilde{\xi}) := \min_{u,v,q} v, \tag{3.37a}$$

$$\text{s.t.} \quad Cu - v\mathbf{1} \leq H\tilde{\xi} + h - B\tilde{w} - Gq, \tag{3.37b}$$

$$v \geq 0. \tag{3.37c}$$

The FSP (3.37) is a mixed-integer program (MIP) that evaluates the feasibility of $\tilde{\xi}$ that can be solved with off-the-shelf MIP solvers. (3.37) determines if the second-stage constraints can be satisfied for $\tilde{\xi}$ if $\tilde{w}$ and $\tilde{\gamma}$ were decided in the first-stage. Therefore, $\underline{\psi}(\tilde{w}, \tilde{\gamma}, \tilde{\xi}) \leq \psi(\tilde{w}, \gamma)$ because (3.37) obtains a feasible solution to (3.36).

### *3.5.2 Feasibility Master Problem*

The FCP is bounded above by the feasibility master problem (FMP):

$$\overline{\psi_R}(\tilde{w}, \tilde{\gamma}) = \max \eta \tag{3.38a}$$

$$\text{s.t.} \quad (3.36\text{b}) - (3.36\text{f}) \tag{3.38b}$$

$$(3.36\text{g}) - (3.36\text{h}) \tag{3.38c}$$

$$\forall r = 1, ..., R.$$

The FMP (3.38) is a relaxation of (3.36) because it considers only $R$ instances of constraints (3.36b)-(3.36f), where $R \leq |\Xi|$. Its objective value is denoted $\overline{\psi_R}(\tilde{w}, \tilde{\gamma})$, where the subscript $R$ indicates the number of included constraints of the form (3.36b)-(3.36f). There still exists the bilinear term, which means we cannot rely on linear solvers to solve (3.38). The bilinear term will be addressed with the bilinear search heuristic (BSH), which is similar to the approach used in [72]. Before we introduce the BSH details, we first introduce some notation. We represent the $R$ dual variables $\lambda^1, ..., \lambda^R$ as a matrix $\Lambda_R := \begin{bmatrix} | & & | \\ \lambda^1 & ... & \lambda^R \\ | & & | \end{bmatrix}$. We also introduce $\overline{\psi_R^1}(\tilde{w}, \tilde{\gamma}, \xi)$ and $\overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \Lambda_R)$ to represent the objective of (3.38) after fixing the value of $\xi$ and $\Lambda_R$, respectively. Since they represent a feasible solution to the FMP (3.38), we have that $\overline{\psi_R^1}(\tilde{w}, \tilde{\gamma}, d), \overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \Lambda_R) \leq \overline{\psi_R}(\tilde{w}, \tilde{\gamma})$.

### *3.5.3 Primal Generation Algorithm*

The details of the PGA are presented as Algorithm 1. The algorithm determines whether $\psi(\tilde{w}, \tilde{\gamma})$ is positive, which describes the first-stage variables' feasibility according to (3.30).

**Algorithm 1:** Primal Generation Algorithm

**Input:** $\epsilon$, $J, \tilde{w}, \tilde{\gamma}$

$m \leftarrow 1$, random $\tilde{\xi}$ and $\tilde{q}$
add variables $(u^1, v^1, \lambda^1)$ and constraints (3.36b)-(3.36f) with $q^1 = \tilde{q}$ to the FMP (3.38)
$R \leftarrow 1$
**while** $n < J$ **do**
  Execute BSH with $\tilde{\xi}$ as input and obtain $\xi^*, \Lambda_R^*$
  $\tilde{\xi} \leftarrow \xi^*, \tilde{\Lambda}_R \leftarrow \Lambda_R^*$
  **if** $\max\{\overline{\psi_R^1}(\tilde{w}, \tilde{\gamma}, \tilde{\xi}), \overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \tilde{\Lambda}_R)\} > \epsilon$ **then**
    $n \leftarrow 1$
    solve (3.37) for $\underline{\psi}(\tilde{w}, \tilde{\gamma}, \tilde{\xi})$ and obtain solution $q^*$
    **if** $\underline{\psi}(\tilde{w}, \tilde{\gamma}, \tilde{\xi}) > \epsilon$ **then**
      add cut constraint to MP: $Bw + Cu + Gq + H\hat{\xi} + H(\gamma \circ \frac{1}{\tilde{\gamma}_t}(\tilde{\xi}_t - \hat{\xi}_t)) \leq h$
      **return**
    **else**
      add variables $(u^{r+1}, v^{r+1}, \lambda^{r+1})$ and constraints (3.36b)-(3.36f) with $q^{r+1} = q^*$ to the FMP (3.38)
      $R \leftarrow R + 1$;
    **end if**
  **else**
    $n \leftarrow n + 1$;
    choose a new RNRG $\tilde{\xi}$
  **end if**
**end while**
**return** $\tilde{\gamma}$

**Algorithm 2:** Bilinear Search Heuristic

**Input:** $\delta, \tilde{\xi}, J' > 1$

1: $j \leftarrow 1, P \leftarrow \infty, Q \leftarrow 0$
2: **while** $j < J'$ **do**
3: Solve (3.38) for $\overline{\psi_R^1}(\tilde{w}, \tilde{\gamma}, \tilde{\xi})$, obtain solution $\tilde{\Lambda}_R$
4: Solve (3.38) for $\overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \tilde{\Lambda}_R)$, obtain solution $\tilde{\xi}$
5: $Q \leftarrow \overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \tilde{\Lambda}_R)$
6: **if** $|P - Q| \geq \epsilon$ **then**
7: $P \leftarrow Q$
8: **else**
9: $\xi^* \leftarrow \tilde{\xi}, \Lambda_R^* \leftarrow \tilde{\Lambda}_R$
10: **return** $\xi^*, \Lambda_R^*$
11: **end if**
12: **end while**
13: $\xi^* \leftarrow \tilde{\xi}, \Lambda_R^* \leftarrow \tilde{\Lambda}_R$
14: **return** $\xi^*, \Lambda_R^*$

The PGA's include the user's violation tolerance $\epsilon \geq 0$, a maximum iteration counter $J$, and the first-stage decisions $\tilde{w}$, $\tilde{\gamma}$. We initialize the iteration counter $m$ to 1, and we randomly select a RNRG value $\tilde{\xi}$ and a second-stage integer solution $\tilde{q}$. Next, we initialze the FMP (3.38) by adding a new set of variables $(u^1, v^1, \lambda^1)$ and constraints (3.36b)-(3.36f) to the FMP (3.38), and setting $q^1 = \tilde{q}$. $R$ is the number of such constraints included in (3.38), therefore, its initial value is 1.

In line 5 of Algorithm 1, $\xi^*$ and $\Lambda_R^*$ are obtained using the BSH, outlined in Algorithm 2. The inputs for the BSH are convergence criteria $\delta$ and maximum iteration counter $J'$. $P$ and $Q$ are placeholders for $\overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \tilde{\Lambda}_R)$ for two subsequent iterations. The BSH sequentially solves (3.38) while alternating between fixing $\xi = \tilde{\xi}$ and $\Lambda_R = \tilde{\Lambda}_R$. The BSH only requires solving LPs, so $\overline{\psi_R^1}(\tilde{w}, \tilde{\gamma}, \tilde{\xi})$ and $\overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \tilde{\Lambda}_R)$ can be obtained efficiently, and there are three mutually exclusive outcomes:

1. $\underline{\psi}(\tilde{w}, \tilde{\gamma}, \tilde{\xi}) > 0$

2. $\underline{\psi}(\tilde{w}, \tilde{\gamma}, \tilde{\xi}) = 0$ and $\overline{\psi_R^1}(\tilde{w}, \tilde{\gamma}, \tilde{\xi})$ or $\overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \tilde{\Lambda}_R) > \epsilon$

3. $\underline{\psi}(\tilde{w}, \tilde{\gamma}, \tilde{\xi}) = 0$ and $\overline{\psi_R^1}(\tilde{w}, \tilde{\gamma}, \xi)$ and $\overline{\psi_R^2}(\tilde{w}, \tilde{\gamma}, \tilde{\Lambda}_R) \leq \epsilon$

In Case 1, $\psi(\tilde{w},\tilde{\gamma}) > 0$ since $\underline{\psi}(\tilde{w},\tilde{\gamma},\tilde{\xi}) \leq \psi(\tilde{w},\tilde{\gamma})$. Therefore, the first-stage decision variables, $\tilde{w}$ and $\tilde{\gamma}$, lead to an infeasible second-stage problem according to (3.30). In line 11, we add a cut constraint $Bw+Cu+Gq+H\hat{\xi}+H(\gamma\circ\frac{1}{\tilde{\gamma}_t}(\tilde{\xi}_t-\hat{\xi}_t)) \leq h$ to the MP to cut off $\tilde{w}$ and $\tilde{\gamma}$ from its feasible region. Here, the notation $\circ$ represents the element-wise multiplication of two vectors. Then, in line 12, Algorithm 1 terminates, and the flow from section 3.3.1 is followed.

In Case 2, we cannot determine whether $\psi(\tilde{w},\tilde{\gamma})$ is positive or 0. Therefore, in line 14 of Algorithm 1, we update (3.38) by adding a set of variables $(u^{r+1}, v^{r+1}, \lambda^{r+1})$ and constraints (3.36b)-(3.36f) to the FMP (3.38), and setting $q^{r+1} = \tilde{q}$. This tightens the relaxation of the FCP (3.36) provided by the FMP (3.38). The parameter $R$ is incremented accordingly in line 15.

Finally, in Case 3, the sign of $\overline{\psi_R}(\tilde{w},\tilde{\gamma})$ is undetermined, so we attempt to obtain a different value of $\overline{\psi_R^1}(\tilde{w},\tilde{\gamma},\tilde{\xi})$ and $\overline{\psi_R^2}(\tilde{w},\tilde{\gamma},\tilde{\Lambda}_R)$ by randomly choosing a different $\tilde{\xi}$ in line 19. For the results presented here, we selected a uniformly random $\tilde{\xi}$ along the boundary of the uncertainty set in this step. This new $\tilde{\xi}$ is evaluated in Lines 5-7. If we are unable to find such a $\tilde{\xi}$ that leads to $\overline{\psi_R^1}(\tilde{w},\tilde{\gamma},\tilde{\xi})$ or $\overline{\psi_R^2}(\tilde{w},\tilde{\gamma},\tilde{\Lambda}_R) > 0$ after $J$ attempts, we conclude that $\overline{\psi_R}(\tilde{w},\tilde{\gamma}) = 0$ and terminate in line 22. In this case, we conclude that $\tilde{\gamma}$ is the optimal flexibility vector, and the entire process flow described in subsection 3.3.1 can be terminated.

## *3.6 Experimental Results*

We performed various experiments to demonstrate our methodology's ability to efficiently assess the VPP's flexibility. We first performed a sensitivity analysis to explore how changing different system parameters affects the VPP's flexibility. Secondly, we performed computational experiments that demonstrate our algorithm's efficiency. Before we present our results, we first introduce the data used for our experiments.

### *3.6.1 Data*

All experiments were performed using Gurobi on an Intel i9-10900 processor. For the algorithm parameters, we choose $N = 20$, $J = 10$, $J' = 20$, and $\epsilon = \delta = 0.1$. The rest of the parameter values used for all experiments are introduced in Tables 3.1-3.3. The quadratic fuel cost is approximated in the same manner described in the previous chapter.

| Gen | $P_n^{min}$ | $P_n^{max}$ | $\overline{RD}_n$ | $RD_n$ | $\overline{RU}_n$ | $RU_n$ | $a$ (\$/kW$^2$) | $b$ (\$/kW) | $c$ (\$) | $SD_n$ (\$) | $SU_D$ (\$) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 35 | 35 | 3.5 | 35 | 3.5 | 0.0089 | 13.47 | 74.33 | 5 | 0 |
| 2 | 10 | 20 | 20 | 2 | 20 | 2 | 0.0044 | 10.29 | 39 | 20 | 0 |
| 3 | 5 | 15 | 15 | 1.5 | 15 | 1.5 | 0.0032 | 12.93 | 41 | 25 | 0 |
| 4 | 10 | 35 | 35 | 3.5 | 35 | 3.5 | 0.00831 | 14.51 | 72.94 | 5 | 0 |
| 5 | 10 | 35 | 35 | 3.5 | 35 | 3.5 | 0.003 | 10.76 | 32.96 | 5 | 0 |
| 6 | 5 | 15 | 15 | 1.5 | 15 | 1.5 | 0.0024 | 12.33 | 28 | 30 | 0 |

Table 3.1: Distributed generator parameters (kW)

| ESS no. | Efficiency | Capacity | Charge Limit | Discharge Limit |
|---|---|---|---|---|
| 1 | 0.9 | 100 | 100 | 100 |

Table 3.2: ESS parameter values (kW)

| DR level | price (\$) | limit (kW) |
|---|---|---|
| 1 | 50 | 10 |
| 2 | 100 | 20 |
| 3 | 150 | 30 |

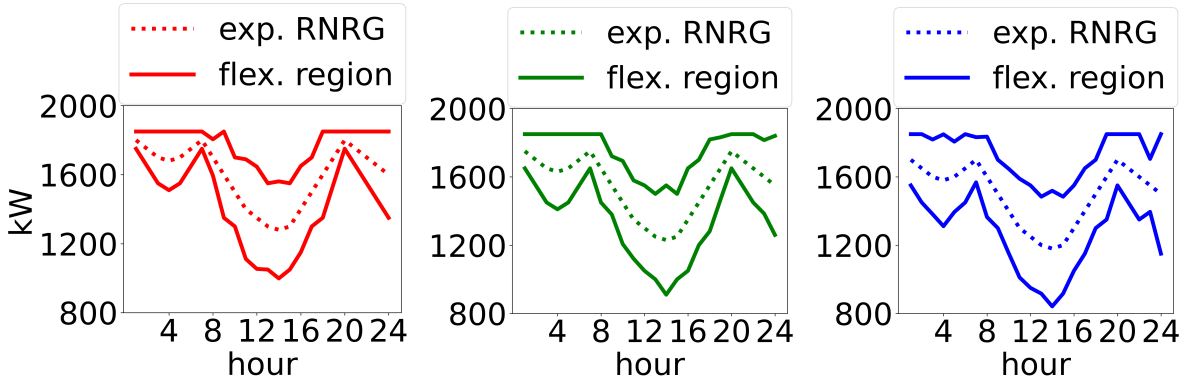
Table 3.3: Demand response parameter values

### *3.6.2 Sensitivity Analysis*

We performed a sensitivity analysis to gain insight into how the system operators can improve their system flexibility with limited resources. We investigate how changes in the expected RNRG, distributed generation capacity, ramping limit, budget, and demand response impact the system's flexibility.

*Expected RNRG*

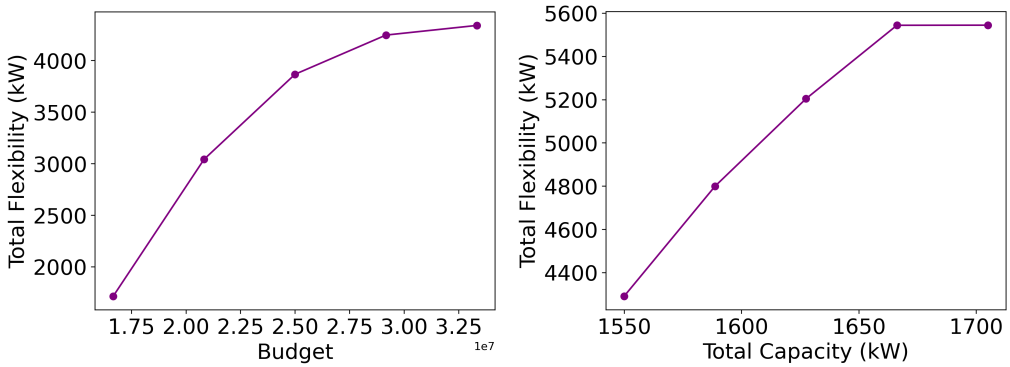
We are interested in evaluating how flexibility changes with respect to expected RNRG. For these experiments, we considered three different profiles of $\hat{\xi}$ as shown in Figure 3.2a, 3.2b, and 3.2c and we let the highest of these values serve as the default expected RNRG for the remaining sensitivity analysis experiments.




(a) High (b) Medium (c) Low

Figure 3.2: Flexibility region for different expected RNRG levels

Figures 3.2a, 3.2b, and 3.2c display the flexibility range for our three different cases for $\hat{\xi}$. From these figures, we observe that the flexibility region is narrowest for the highest expected RNRG case and widest for the lowest expected RNRG case. Particularly, this difference is most dramatic during the periods of higher expected RNRG in hours 8 and 20. This indicates that the system is more flexible under low expected RNRG cases, and less flexible under higher expected RNRG cases. We can conclude that a higher $\hat{\xi}$ value puts more strain on the system's ability to satisfy deviations from the expected RNRG. The total flexibility across the entire horizon for the low, medium, and high expected RNRG is 5608.49, 4988.45, and 4291.11 kW, respectively.




(a) Budget

(b) Generation capacity

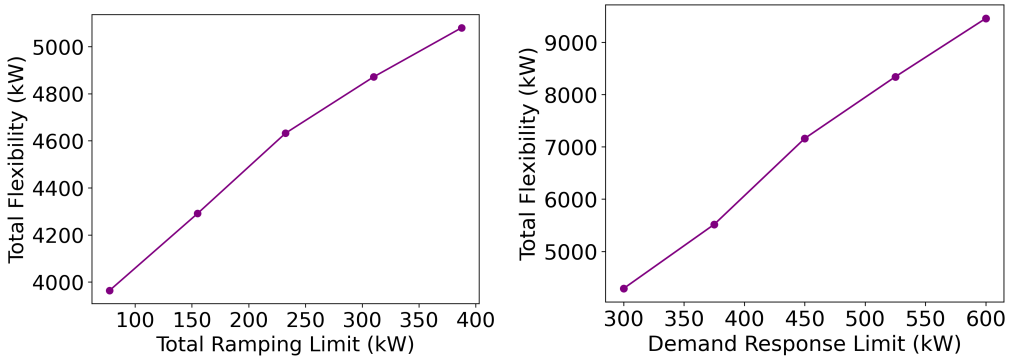
Figure 3.3: Sensitivity analysis

*Budget*

The relationship between flexibility and system budget is shown in Figure 3.3a. For smaller budgets, increasing the budget increases the flexibility significantly because the VPP can spend more on generation resources or demand response to handle larger deviations from the expected RNRG. However, as the budget gets larger, the flexibility experiences a diminished increase because the flexibility is limited by operational constraints such as capacity or ramping. Therefore, increasing the budget does not significantly improve the flexibility at this point.

*Generation Capacity*

The total flexibility across all periods for different generation capacities is shown in Figure 3.3b. These results show that as the capacity increases, the flexibility also increases, but the rate of increase gets smaller for larger capacity values. Therefore, flexibility does not have a linear relationship with the capacity. This is because increasing the capacity prepares the system more adequately for changes in expected RNRG. However, when the capacity is larger, increasing it minimally affects flexibility because the generator's contribution to flexibility is restricted by its ramping limit.




(a) Ramping limit

(b) Demand response

Figure 3.4: Sensitivity analysis (cont.)

*Ramping Limit*

Figure 3.4a presents the flexibility for different ramping limit values. It appears that there is a linear relationship between flexibility and the ramping limit. If the VPP has more ramping capability, it is better prepared to accommodate greater deviations from the expected RNRG, which increases the flexibility of the VPP.

*Demand Response*

The total flexibility for different demand response limits is shown in Figure 3.4b. These results show a similar trend to the linear trend in Figure 3.4a. With greater demand response capabilities, the VPP has a greater ability to reduce customer demand, allowing it to handle a larger deviation from the expected RNRG from its expected value. We further attempted to evaluate the impact of demand response by measuring the VPP's flexibility in two scenarios: one with demand response capabilities and one without for the three expected RNRG profiles shown in Figure 3.2. Our results showed that the system was infeasible when the demand response was removed completely from the system, demonstrating how crucial demand response is in satisfying customer demand.

The results of the sensitivity analysis demonstrate the relative impacts of changing different system parameters on the flexibility, which provides insight that can help system planners determine the value of investing in different resources.

### *3.6.3 Computational Experiments*

In this section, we report the results of experiments conducted to showcase our algorithm's computational efficiency. We compared our PGA with a benchmark CCG algorithm to demonstrate our algorithm's relative ability to solve our problem efficiently without sacrificing optimality. We further demonstrate our PGA's scalability by testing it on a larger problem instance.

*Benchmarking*

We tested the efficiency of our algorithm by comparing it to a standard CCGA algorithm similar to [73] as a benchmark. This method introduces auxiliary binary variables to address the bilinear term. We compared our PGA method with this CCG algorithm for different problem sizes ranging from a time horizon of $T = 4$ to $T = 24$. The results are shown in Table 3.4, where CCGA represents the CCG algorithm we use as a benchmark.

| $T$ | 4 | 8 | 12 | 16 | 20 | 24 |
|---|---|---|---|---|---|---|
| tol = 0.1% | 1 | 3 | 120 | 2381 | × | × |
| tol = 1% | 8 | 9 | 25 | 671 | 1362 | 1792 |
| opt. gap (%) | 0.0 | 0.0 | 0.0 | 1.0 | – | – |

Table 3.4: Runtime analysis (tolerance = 0.1%)

The algorithms have comparable performance up to $T = 8$ but for $T = 12$ and 16 the runtime of the CCG algorithm increases dramatically. Finally, for $T = 20$, and 24 the algorithm is not able to solve the problem in under 2 hours while our method can still solve the problem in less than a minute. We also observed a small optimality gap, which we define as the difference between the PGA solution and the optimal solution divided by the optimal solution. The CCG algorithm we tested is an exact method, so these results show that our method can achieve a near-optimal solution.

The results presented in Table 3.4 were obtained using an optimality gap tolerance of 0.1%, which means the solver terminates when it finds a solution within 0.1% of the optimal solution every time it solves a MIP. We performed the same experiments setting the tolerance equal to 1%, and we present the runtimes for the CCGA and PGA as well as the optimality gap for problems with different time horizons below in Table 3.5.

| $T$ | 4 | 8 | 12 | 16 | 20 | 24 |
|---|---|---|---|---|---|---|
| tol = 0.1% | 5 | 8 | 136 | × | × | × |
| tol = 1% | 24 | 78 | 90 | 81 | 134 | 281 |
| opt. gap (%) | 0.0 | 0.0 | 0.0 | – | – | – |

Table 3.5: Runtime analysis (tolerance = 1%)

For the smaller time horizons from $t = 4$ to 12 we observe that for both the CCGA and PGA, the runtime is greater when the tolerance is increased to 1%. However, for the remaining time horizons, the runtime is faster when the tolerance is increased to 1%. The increased tolerance allows the MIP to be solved faster since it only needs to achieve a 1% instead of 0.1% optimality gap. This change does not make a difference in runtime for smaller problem instances because the MIPs are small and can be solved quickly regardless of the tolerance level. However, for the larger time horizons, the MIPs are larger, so a significant amount of time is saved by increasing the tolerance.

In both cases displayed in Figures 3.4 and 3.5, the CCG algorithm was not able to solve a model with a time horizon of $T = 24$. This is because the CCGA relies on solving MIPs because it introduces auxiliary binary variables to address the bilinear term. In contrast, our PGA relies on solving LPs as a result of the bilinear search heuristic.

| $T$ | 4 | 8 | 12 | 16 | 20 | 24 |
|---|---|---|---|---|---|---|
| tol = 0.1% | 470.00 | 920.00 | 1945.00 | 2951.11 | 3541.66 | 4218.15 |
| tol = 1% | 470.00 | 920.00 | 1945.00 | 3031.11 | 3612.56 | 4291.11 |
| avg gap (%) | 0.00 | 0.00 | 0.00 | 0.17 | 0.10 | 0.07 |

Table 3.6: PGA total flexibility (kW) for two tolerances

We can also observe how optimal flexibility is affected by increasing the tolerance. Table 3.6 shows the total flexibility for when the MIP the tolerance levels are 0.1% and 1% for different sized problem instances. We also record the average gap for each hour in the time horizon, where the gap is defined as the difference between the flexibility with the 1% tolerance and 0.1% under the solution and the optimal solution divided by the flexibility with the 0.1% tolerance. For $t = 4$ to 12, both flexibility values are identical. For $t = 16$ to 24, the average gap between the flexibility

from the two tolerance levels is small with values of 0.17%, 0.10%, and 0.07%. These results show that the significant time saved by increasing the optimality tolerance from 0.1% to 1% does not significantly sacrifice the optimality of the obtained flexibility.

*Scalability Testing*

Furthermore, we tested our algorithm on the Synthetic South Carolina 200-bus Power System Model [53] to demonstrate its scalability. We solved a 24-hour problem with our PGA with optimality tolerance values of 0.1% and 1% and achieved runtimes of 70.40 minutes and 275 seconds, respectively. for the South Carolina 200-bus model compared to the 6-generator case. The smaller tolerance makes each MIP slower to solve, and this is exacerbated by the fact that there are more binary variables in the MIP with the South Carolina 200-bus model. However, this runtime is sufficiently small for typical day-ahead planning settings, which our methodology is intended for. But if a faster runtime is desired, this can be achieved by increasing the optimality tolerance.

For the large tolerance of 1%, we do not see any significant change in runtime. This could be because not only is each MIP faster to solve, but the complexity of increasing the number of binary variables only affects the first-stage problem, so the runtime of only the Master Problem is affected, but not the FMP. Finally, we observed that increasing the tolerance yielded an average gap of 0.24%, so the optimal flexibility obtained with a tolerance of 1% is close to the flexibility when the tolerance is 0.1%

## *3.7 Conclusion*

We formulated our flexibility problem as a robust two-stage uncertainty-set optimization problem with second-stage integer variables. The second-stage integer variables allow us to model quick-start storage systems, but they make our model challenging to solve because duality approaches are not applicable. To address this, we reformulate the problem as a MIP with a bilinear term in the constraints. Then, the problem is solved efficiently by our PGA, which utilizes a BSH and leverages the problem's special structure. Our sensitivity analysis demonstrated how our methodology can provide insight to VPP operators by evaluating how flexibility is affected by changing system parameters. Further results showed that our algorithm performs more efficiently than a standard CCG algorithm without sacrificing optimality.

# Chapter 4

# A RAWLSIAN MIXED INTEGER PROGRAMMING APPROACH FOR FAIR CLASSIFICATION

## *4.1 Background*

Binary classification algorithms are widely used machine learning algorithms that classify data into separate classes or groups. While classification algorithms can be powerful tools, they can produce unfair results that favor certain groups while hurting others. The COMPAS [15] algorithm described in Chapter 1 is a famous example of this phenomenon.

There are many different ways to evaluate fairness with classification algorithms. A common notion of fairness is the assurance that individuals of different demographic groups are treated similarly. To this end, three of the most common fairness metrics used in conjunction with classification algorithms are statistical parity, equal opportunity, and equalized odds [74]. Statistical parity ensures that subjects in different groups have an equal probability of receiving the same classification. Equal opportunity ensures that subjects in different demographic groups have the same probability of being classified as a negative class if they are truly in the positive class. Equalized odds ensures that the probability of being correctly assigned to the positive class for members of the positive class and the probability of being incorrectly assigned to the positive class for members of the negative class are the same across all groups.

Satisfying these fairness metrics often leads to significant deterioration of the classification's prediction error. This phenomenon is known as the fairness-accuracy trade-off [75]. We propose a Rawlsian (also often referred to as minimax) fairness metric based on [76] that is designed to prioritize the model's performance for the worst-off group. This methodology does not sacrifice error to the extent that the above metrics do because it does not restrict the prediction accuracy for any demographic group in an attempt to achieve similar performance across all demographic groups.

Fairness can be incorporated into classification algorithms in three different ways: pre-processing, in-processing, and post-processing. Pre-processing methods adjust the data before feeding it into any classification algorithm to ensure fairer results [77]. In-processing algorithms do not change the data beforehand, and instead create customized algorithms that are designed to output fair

results [78]. Finally, post-processing methods adjust the output of a classification algorithm to yield fairer results [1]. Pre-processing and post-processing methods can be used with any classification algorithm. However, these methods can lead to high variability with the ultimate performance error [79], and especially, the post-processing methods can lead to sub-optimal results [80]. Our proposed method is an in-processing method to ensure fairness to be used for any dataset and does not require any pre or post-processing. Furthermore, our model is particularly suited to handle unbalanced datasets, which present specific challenges to fair classification problems.

Unbalanced datasets have one class (the majority class) that has significantly more entries than other classes (the minority classes). Such datasets often yield unfair outcomes, as conventional classification algorithms tend to favor the majority class, relegating minority classes to noise. Consequently, during the training phase, the influence of minority classes is severely limited [81]. To address these issues, common approaches involve altering the data sampling for the algorithm. Random over-sampling (ROS) involves sampling minority entries more frequently while under-sampling (RUS) reduces the frequency of majority entries [82, 83]. However, ROS can introduce additional noise to the model due to the inclusion of extra samples, potentially leading to overfitting. Conversely, RUS may remove crucial training data, resulting in underfitting. Alternatively, some methods refrain from altering the sampling procedure but provide specialized loss functions tailored for unbalanced datasets [84] [85] [86]. However, these loss functions are often nonlinear and challenging to solve.

Our Rawlsian fairness approach mitigates the effects of unbalanced datasets by ensuring that each demographic group does not have their error compromised regardless of how much of the dataset they comprise. Rawlsian fairness has been considered used in a variety of applications such as supply-chain management [87], healthcare [88], and resource allocation [89]. Some examples of Rawlsian fairness used with classification in the literature include [90], which formulates a Rawlsian loss function, but they don't solve it directly, and only provide bounds for it. [91] also provides a Rawlsian objective function, but their model is only solvable for special cases when the second-order statistics are known. [92] and [93] also propose Rawlsian fairness frameworks, however, their models are limited in their ability to increase flexibility and interpretability, compared to our method.

We combine our Rawlsian fairness method with a mixed integer programming (MIP) framework to formulate our model as a constrained optimization problem with a linear objective and constraints. Our MIP framework has certain advantages compared to common classification methodologies. One common classification method is the use of a score function with a threshold [94].

These algorithms assign a score to each data entry and classify them based on whether the score is above or below a certain threshold. One issue with these methods is that the threshold is an input that must be decided by the user, and it is not obvious what threshold value will give the best result. Area under the curve (AUC) [95] is another metric used for classification problems that evaluate a score function's ability to correctly classify data for all possible threshold values. Optimizing the AUC is robust to threshold selection since the user does not need to figure out what the best threshold is. However, optimizing AUC can lead to overly-conservative results since it considers thresholds that might lead to poor prediction accuracy. Furthermore, directly optimizing AUC is an NP-hard problem, so it must be approximated by other functions [96]. In our model, the threshold is a decision variable, so its optimal value is output upon solving the optimization problem. Therefore, the burden of choosing the threshold value as an input before solving the model is alleviated. Furthermore, the results will not be overly-conservative like the AUC optimization methods, since we consider the optimal threshold value minimizes prediction error instead of considering all possible threshold values.

In contrast to these classification methods, MIP frameworks like the Supersparse Linear Integer Model (SLIM) can solve classification problems while guaranteeing optimal balance between sparsity and error without the need for additional post-processing [97]. These models also allow the user to obtain integer coefficients which could be desirable depending on the application the model is used for. Another advantage of these methods is that they directly provide a predicted outcome instead of a probabilistic outcome because they do not require a threshold value as a model input (the optimal threshold is implicitly solved for) [98].

In general, a Rawlsian fairness framework is difficult to implement since it includes a discrete robust objective. Such an objective can be reformulated by moving the loss function to the constraints and optimizing over an auxiliary variable, but the resulting formulation could be difficult to solve if the original loss function is nonlinear. By using a MIP framework, however, we can linearize our loss function and reformulate our model as a MIP with a singular minimization objective. Then, the resulting problem can be solved by off-the-shelf solvers. Furthermore, using a MIP framework provides modeling flexibility to adjust sparsity or interpretability depending on the user's preferences [99]. We summarize our contributions as follows:

1. We propose a novel Rawlsian MIP method that is specifically suited to handle unbalanced datasets and provide fair classification results by prioritizing the worst-off group

2. Our methodology provides a single optimal threshold that generates deterministic classification results. Furthermore, it provides flexibility by allowing the user to optimize sparsity in multiple ways and choose between real or integer coefficients

3. We provide experimental results that demonstrate our model's ability to produce fair results without sacrificing classification error in comparison to other fair classification algorithms

In the remainder of this chapter, we introduce our model formulation in Section 4.2. In section 4.3, we introduce the datasets used in the experimental results presented in section 4.4. In section 4.5 we conclude the chapter.

## *4.2 Model Formulation*

### *4.2.1 Preliminaries*

First, we introduce some preliminary notation. We consider a dataset with $N$ entries with $P$ features and $K$ demographic groups. For example, race could be a demographic group we could consider, so our methodology would ensure that none of racial groups $1, ..., K$ suffers disparate classification error, by prioritizing the worst-off group. $(1, x_1^i, ..., x_P^i)$, is a feature vector for the $i^{th}$ datapoint, $y_i \in \{-1, 1\}$ is true labels for data point $i$, and $\hat{y}_i = sign(x^T\lambda)$ is the predicted label for data point $i$. $\lambda = (\lambda_0, ..., \lambda_P)$ is a vector of coefficients used to define a linear score function. $\epsilon$ is a small perturbation term that represents how much we want to penalize small negative terms in the objective (larger value corresponds to less penalization). $\Lambda$ is the maximum allowable value for $\lambda$ and $M$ is a large number.

We include $L_0$ and $L_1$ regularization in our objective, and $C_0$ and $C_1$ are the respective penalty coefficients. $L_0$ regularization allows our model to directly optimize sparsity, but minimizing the $L_0$ norm is a non-convex problem [100]. However, our MIP formulation provides a framework that can address this non-convexity and allows us to exactly minimize the $L_0$ norm. The $L_1$ norm is typically used as an approximation of $L_0$, but it can be more stable and robust in certain settings [101], and combining the two regularization terms can improve variable selection and stability [102]. Furthermore, if the coefficients must be integer values, using $L_1$ regularization in addition to $L_0$ can ensure that the optimal coefficients are coprime integers [103]. Integer coefficients are desirable for linear scoring systems, which are commonly used in applications such as education [104] or healthcare [105].

### *4.2.2 Original Loss Function*

The original loss function used in [97] minimizes classification by minimizing the number of incorrect classifications, which is equivalent to maximizing classification accuracy:

$$\min_{\lambda \in \mathcal{X}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{y_i (x^i)^T \lambda \leq 0\} + C_0 \|\lambda\|_0 + C_1 \|\lambda\|_1 \tag{4.1}$$

This loss function, however, does not incorporate any fairness.Here, $\mathcal{X}$ is a set of feasible solutions, where $\mathcal{X} \in \{\mathbb{R}^{P+1}, \mathbb{Z}^{P+1}\}$. In other words, we allow $\lambda$ to be a real or integer-valued $P+1$-dimensional vector. $\mathbb{1}$ is an indicator function whose value is 1 if its input is true and 0 otherwise. The objective is to choose a $\lambda$ to minimize the number of incorrect classifications. A classification, in this context, is determined by the sign of $(x^i)^T \lambda$. Formally, the classification method is the following:

$$\hat{y}_i = \begin{cases} 1 & \text{if } (x^i)^T \lambda > 0 \\ -1 & \text{if } (x^i)^T \lambda \leq 0 \end{cases} \tag{4.2}$$

In other words, if $(x^i)^T \lambda > 0$ and $y_i = 1$ or $(x^i)^T \lambda < 0$ and $y_i = -1$, this is considered a correct classification. Otherwise, it is considered incorrect. The $C_0 \|\lambda\|_0$ and $C_1 \|\lambda\|_1$ terms exist to penalize any $\lambda$ with a large $L_0$ or $L_1$ norm, respectively.

Note that because $x_1^i = 1 \ \forall i$ by definition, $\lambda_0$ can be thought of as an intercept variable whose value adjusts the overall score function $(x^i)^T \lambda$ by a scalar. Therefore, $\lambda_0^*$, which is the $1^{st}$ component of the optimal solution $\lambda^*$ acts as a threshold for the score function. This can be understood by observing the equivalent representation of the classification method (4.2):

$$\hat{y}_i = \begin{cases} 1 & \text{if } \sum_{j=1}^{P} \lambda_j x_j^i \geq -\lambda_0 \\ -1 & \text{if } \sum_{j=1}^{P} \lambda_j x_j^i \leq -\lambda_0 \end{cases} \tag{4.3}$$

Thus, the objective is to choose $\lambda_0$ such that data point $i$ is assigned to to class 1 if $\sum_{j=2}^{P+1} \lambda_j x_j^i \geq -\lambda_0$ or to class $-1$ if $\sum_{j=2}^{P+1} \lambda_j x_j^i \leq -\lambda_0$. $\lambda_0$ is a decision variable in the MIP, so the optimal threshold $\lambda_0^*$ is an output of the problem. Therefore, its value need not be decided by the user before implementation. Because our methodology outputs the optimal threshold value, it has an advantage over methods that rely on predetermined thresholds that could be sub-optimal, such as logistic regression [106].

### *4.2.3 Rawlsian Loss Function*

Next, we introduce our novel Rawlsian loss function, which minimizes the classification error for the worst-off demographic group.

$$\min_{\lambda \in \mathcal{X}^{P+1}} \left\{ \max_{k \in [K]} \{ \frac{1}{N_k} \sum_{j=1}^{N_k} \mathbb{1}\{ y_{k(i)} (x^{k(i)})^T \lambda \leq 0 \} + C_0 \|\lambda\|_0 + C_1 \|\lambda\|_1 \} \right\} \tag{4.4}$$

Here, the notation $[K]$ represents the index set $\{1, ..., K\}$. $N_k$ represents the number of data entries for demographic group $k$. The index $k(i)$ represents the $j^{th}$ entry within the $k^{th}$ group. Therefore, (4.4) is equivalent to minimizing the classification error rate (which can be thought of as the average classification error) for the worst-off demographic group. (4.4) is a robust optimization problem because it has an outer-minimization objective paired with an inner-maximization objective. It is difficult to solve because the inner-maximization objective function has a nonlinear expression, and it is optimized over a discrete support. Therefore, duality methods, which are commonly applied to reformulate robust optimization problems, cannot be applied to this problem. To reformulate (4.4), we first reformulate the original loss function (4.1) as the following MIP [97]:

### *4.2.4 MIP Reformulation for Original Loss Function*

$$\min_{\alpha, \beta, \gamma, \lambda} \frac{1}{N} \sum_{i=1}^{N} \alpha_i + C_0 \sum_{j=1}^{P} \beta_j + C_1 \sum_{j=1}^{P} \gamma_j \tag{4.5}$$

$$-M\alpha_i + \epsilon \leq y_i (x^i)^T \lambda \leq M(1 - \alpha_i) + \epsilon \quad i = 1, ..., N \tag{4.6}$$

$$-\Lambda \beta_j \leq \lambda_j \leq \Lambda \beta_j \quad j = 1, , , .P \tag{4.7}$$

$$-\gamma_j \leq \lambda_j \leq \gamma_j \quad j = 1, , , .P \tag{4.8}$$

$$\alpha \in \{0, 1\}^N, \beta \in \{0, 1\}^P, \gamma \in \mathbb{R}_+^P, \lambda \in \mathcal{X} \tag{4.9}$$

This new formulation (4.5)-(4.9) introduces new auxillary variables $\alpha, \beta$ and $\gamma$. With this new reformulation, $\alpha_i = \mathbb{1}\{y_i \neq \hat{y}_i\}, \beta_j = \mathbb{1}\{\lambda_j \neq 0\}$, and $\gamma_j = |\lambda_j|$. (4.5) is the average incorrect predictions plus the zero and one norm penalty terms. (4.6) assigns the auxiliary variable $\alpha_i$ to 0 or 1 depending on if the prediction is correct or incorrect, respectively. (4.7) assigns the value for the auxiliary variable $\beta$ to the $L_0$ norm. (4.8) assigns the value for the auxiliary variable $\gamma$ to the $L_1$ norm. Next, we introduce our novel formulation which combines the MIP (4.5)-(4.9) with the Rawlsian fairness objective (4.4).

#### *4.2.5 Rawlsian Reformulation*

$$\min_{\alpha,\beta\gamma,\lambda,\theta} \theta + C_0 \sum_{j=1}^{P} \beta_j + C_1 \sum_{j=1}^{P} \gamma_j \tag{4.10}$$

$$\theta \geq \frac{1}{N_k} \sum_{i=1}^{N_k} \alpha_{k(i)} \quad \forall k = 1, ...K, i = 1, ..., N_k \tag{4.11}$$

$$-M\alpha_i + \epsilon \leq y_{k(i)}(x^{k(i)})^T \lambda \leq M(1 - \alpha_{k(i)}) + \epsilon \quad \forall k = 1, ...K, i = 1, ..., N_k \tag{4.12}$$

$$-\Lambda\beta_j \leq \lambda_j \leq \Lambda\beta_j \quad j = 1, , , .P \tag{4.13}$$

$$-\gamma_j \leq \lambda_j \leq \gamma_j \quad j = 1, , , .P \tag{4.14}$$

$$\alpha \in \{0,1\}^N, \beta \in \{0,1\}^P, \gamma \in \mathbb{R}_+^P, \lambda \in \mathcal{X} \tag{4.15}$$

We include a new auxiliary decision variable, $\theta$, which we use as a measure of fairness. $\theta$ represents the classification error rate for the worst-off demographic group, which is facilitated by constraint (4.11). Leveraging this constraint allows us to eliminate the min-max objective in (4.4) and replace it with a single linear minimization objective. The resulting problem (4.10)-(4.15) is a MIP that can be solved by off-the-shelf solvers.

By using a MIP, we have significant modeling flexibility to customize the model based on the user's preference. Some examples of this customizability are as follows. We can achieve the optimal balance between sparsity and error as in [97] by tuning the values of $C_0$ and $C_1$. Furthermore, we can add additional hard constraints if the user wants to impose encourage restrictions more strictly. Additionally, $\lambda$ can be forced to be an integer vector, which would improve interpretability and prevent over-fitting. We discuss these proprieties in greater detail in the next section

### *4.3 Experimental Results*

#### *4.3.1 Experimental Setup*

We test our algorithm on two datasets and present several experiments to demonstrate our method's ability to achieve high classification accuracy without sacrificing the worst-case group's performance, as well as its flexibility and interpretability. We also compare our model's results to two different fair classification algorithms from the literature: [1] and [2], We conclude this section by summarizing our model's advantages in comparison to the other models. The following experiments use $\epsilon = 0.1$, $\Gamma = 1000000$, $C_0 = C_1 = 0$, $M = 100$, and $\mathcal{X} = \mathbb{R}^{P+1}$ unless stated otherwise. We use the Gurobi solver to solve the model on a 1.6 GHz Dual-Core Intel Core i5 processor.

Our first results are presented in Section 4.3.2, where we explore our model's performance on the Penguin Dataset. In section 4.3.3, we evaluate our model's performance on the larger Law Dataset, and present results that provide insight into the relationship between sparsity and model performance, In section 4.3.4, we compare our methodology to methods that define fairness differently as well as methods that define fairness similarly.

### *4.3.2 Penguin Dataset*

The first dataset we test our algorithm on is the Penguin Dataset from [107], which contains 344 instances of penguins, and the classification task is to determine if each individual's species is Adelie, Gentoo, or Chinstrap species. The data includes information on the penguins' bill length, bill depth, flipper length, body mass, and the year the data was collected. The demographic group we consider is the penguin's gender. We compare our Rawlsian MIP performance to the standard MIP approach used in [97] that doesn't consider fairness. For the training phase, we implement the MIP ((4.5)-(4.9) for the MIP with no fairness considerations and (4.10)-(4.15) for our Rawlsian MIP) on half of the dataset and obtain an optimal $\lambda^*$. Then, in the testing phase, we use this $\lambda^*$ to classify the second half of the data. Since our formulation only performs binary classification, we perform three separate experiments, where the classification task for Experiments 1, 2, and 3 is to determine whether a penguin species is Adelie or not-Adelie, Gentoo or not-Gentoo, and Chinstrap or not-Chinstrap, respectively. We present these results below in Table 4.1

| Exp. | MIP error (%) | Rawls error (%) | MIP fairness (%) | Rawls fairness (%) |
|---|---|---|---|---|
| 1 | 1.75 | 1.17 | 3.49 (M) | 2.53 (F) |
| 2 | 1.17 | 1.17 | 3.27 (F) | 3.27 (F) |
| 3 | 4.09 | 3.51 | 4.65 (M) | 3.80 (F) |

Table 4.1: Model comparison for Penguin Dataset

We compare the error and the fairness of our Rawlsian fairness method and the MIP method that does not consider fairness. Here, we define error as the percentage of incorrect classifications, and fairness is defined as the highest error across all demographic groups, which is a common way to evaluate fairness also used in [90], [91], [92], and [93]. Therefore, lower values of error and fairness are desirable. For example, if a dataset contained 2 black individuals and 8 white individuals for a

total of 10 individuals, and it incorrectly classified 1 black individual and 2 white individuals, the error would be $\frac{1+2}{10} = 0.4$ and the fairness would be $\max\{\frac{1}{2}, \frac{2}{8}\} = 0.5$. Furthermore, in these results, we indicate (M) or (F) to indicate whether the male or female group respectively, is the worst-off group for the particular instance. In all experiments, both methods had perfect accuracy during the training phase so we only present results from the testing phase in Table 4.1. We can see that our method achieves the same performance with respect to error and fairness in Experiment 2, but it improves in both metrics for Experiments 1 and 3. These results show that our methodology can improve accuracy for the worst-off group while additionally improving overall accuracy.

Furthermore, in these cases, the worst-off group in the MIP case was male, but it switched to female in the Rawlsian MIP case. Our method prioritizes helping the worst-off group, however, the worst-off group can differ for different instances. The worst-off group is an output of our model, so it determines who the worst-off group is during its execution. This means the user does not need to identify which group is the worst-off group before implementation. This is a useful property for cases where domain knowledge is limited and the user does not know which groups will be more vulnerable before running the model. Fortunately, our model can execute without this knowledge and it will produce a solution that minimizes the error for the worst-off group regardless of whether the worst-off group is male or female.

### *4.3.3 Law Student Dataset*

Next, we consider the Law Student Dataset [108], which is much larger than the Penguin Dataset. A detailed description of the dataset and access to the dataset itself is provided by [109]. It contains $20,798$ entries from 163 law schools in the United States in 1991. Because of the large size of the dataset, we are unable to solve the MIP to optimality given our available computational resources, so the results we present in this section represent the best solution obtained after one hour of runtime. Despite this, the results obtained still outperform other state-of-the-art fair classification algorithms. We provide a more detailed discussion of this comparison in Section 4.3.4.

The classification task is to predict whether a student will pass the bar exam or not. There are 9 features that we consider in the model that describe the student attributes such as GPA or LSAT score. Unlike the Penguin Dataset studied in the previous section, which had a nearly identical count of males and females, the Law Student Dataset is unbalanced. The majority race consists of 93% of the dataset and the minority race only consists of 7%. This presents a challenge because most classification algorithms that aim to maximize overall accuracy will ignore the minority class,

so these algorithms produce results where the minority class experiences a significantly higher error rate.

In the results presented in the remainder of Section 4.3.3, we first explore the relationship between model performance and sparsity. Particularly, we analyze the difference between incorporating sparsity via penalization or upper-bounding. We conclude Section 4.3.3 with a discussion on our algorithm's ability to address the unbalanced Law Student Dataset by prioritizing prediction accuracy for the worst-off group without sacrificing overall prediction accuracy.

*Sparsity via Penalization*

For our first experiment, we compare our model performance for different values of $C_0$ and $C_1$. We considered different values of $C_0$ ranging from 0 to 0.0175, and we considered two scenarios for $C_1$: $C_1 = 0$ and $C_1 = 0.001$. For the results presented below, we set $\mathcal{X} = \mathbb{Z}^{P+1}$. We evaluated the fairness produced by our model as the error for the worst-off demographic group, and we recorded this metric for the testing and training phases, as well as the overall performance, which is the average of the testing and training results.

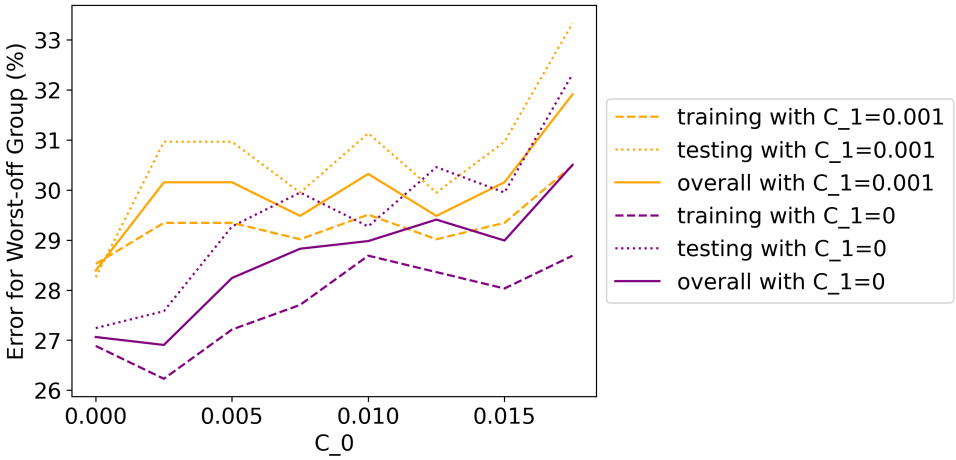



Figure 4.1: Fairness for different penalization values

As an overall trend, as $C_0$ increases, the error for the worst-off group increases. This indicates that higher $C_0$ values worsen our methodology's ability to enhance fairness. As $C_0$ increases, there appears to be a steeper increase in error for the worst-off group when $C_1 = 0$ compared to when $C_1 = 0.001$. Therefore, increasing $C_0$ has a more dramatic impact on the fairness when $C_1 = 0$ as opposed to when $C_1 = 0.001$. We can also observe that the overall fairness is worse when $C_1$ is

increased from 0 to 0.001 for all values of $C_0$.

$C_0$ and $C_1$ are both penalization terms in the objective, so increasing their values encourages the model to produce sparser solutions that have fewer non-zero coefficients, which ultimately improves interpretability. However, the results in Figure 4.2 indicate that increasing these penalization terms worsens the fairness of the model's output. Thus, these results demonstrate a trade-off between sparsity and fairness. However, we provide an alternative method to induce sparsity without sacrificing fairness, which we introduce next.

*Sparsity via Upper-bounding*

To further increase sparsity and interpretability we add a constraint that was not considered in [97]:

$$\sum_{i}^{N} \beta_i \leq b \tag{4.16}$$

This constraint limits the size of the decision variable $\beta$ by upper-bounding the sum of its components by a parameter $b$. This is ultimately a stronger way to reduce the size of the $L_0$ norm by introducing a hard constraint rather than using a penalization term in the objective function. We tested our model's performance for different values of $b$ ranging from 3 to 7, and we considered the case where $\lambda$ is restricted to real and integer values. Similarly to the results presented in Figure 4.2, we recorded the testing, training, and overall fairness, which we measure as the error for the worst-off group.

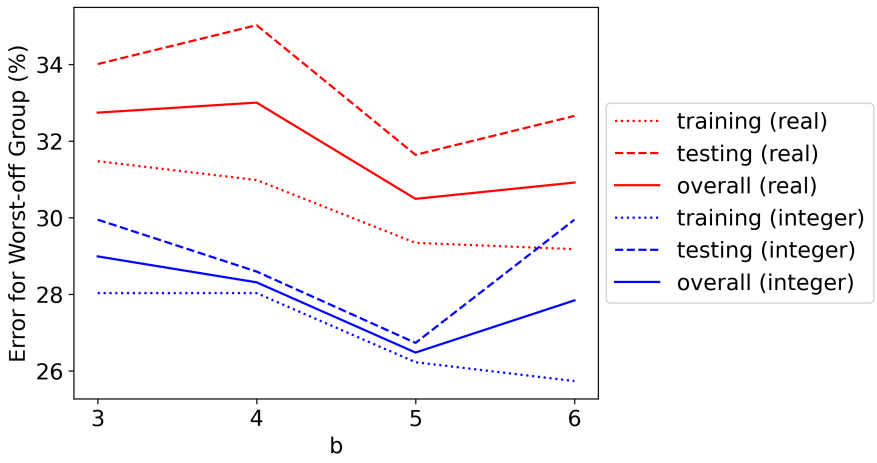



Figure 4.2: Fairness for different upper-bound parameter values

In both the real and integer cases, the best fairness was achieved at $b = 5$. In the integer case,

these results have a lower error for the worst-off group than any of the results shown in Figure 4.2, when the penalization terms were used. This indicates using constraint (4.16) does not suffer a decrease in fairness when sparsity is induced. When $b$ is increased to 6, we see the training performance improves because the constraint (4.16) is relaxed so the MIP can find a better solution in the training phase. However, this results in a worse overall performance, which indicates that the model is overfitting in this scenario. On the other hand, as $b$ is decreased to values of 3 or 4, the training performance also worsens because constraint (4.16) becomes tighter as $b$ decreases, and in these cases, the testing performance also suffers. Furthermore, we can also observe that the model performs better when $\lambda$ is restricted to be an integer across all values of $b$. Additionally, if $\lambda^*$ is an integer vector, this could produce a cleaner result that is easier for decision-makers to implement in practical settings, such as [104] [105]. We can compare the performance of our Rawlsian MIP method to a standard MIP formulation that does not consider fairness like the one used in [97].

| | training error(%) | training fairness(%) | testing error(%) | testing fairness(%) |
|---|---|---|---|---|
| MIP | 8.90 | 27.54 | 9.27 | 31.30 |
| Rawlsian | 9.30 | 26.23 | 9.07 | 26.73 |

Table 4.2: Error and fairness when $b = 5$ and $\lambda \in \mathbb{Z}^{P+1}$

During training, our Rawlsian MIP does less than 1% worse concerning error, but it does a little more than 1% better for fairness compared to the MIP. During testing, our Rawlsian MIP performs slightly better (less than 1% ) concerning error but is 5% better for fairness compared to the MIP. If we consider the overall performance (average of testing and training), our Rawlsian method improves fairness by more than 3%, while increasing prediction error by only 0.1% compared to the MIP.

In summary, our MIP gives us the modeling flexibility to apply a hard constraint in the form of (4.16) to provide an upper bound on the sparsity, which is typically difficult to implement due to the non-convexity of the $L_0$ norm. Unlike the results obtained when the penalization terms $C_0$ and $C_1$ were utilized, applying this constraint (4.16) allowed us to increase sparsity without sacrificing model performance, as shown in Table 4.2.

### *4.3.4 Comparisons*

In this section, we consider our results from Table 4.2, where we achieved a training, testing, and overall prediction error of 9.30%, 9.07%, and 9.19%, respectively, and training, testing, and overall prediction error for the worst-off group of 26.23%, 26.73%, and 26.48% respectively. We compare these results to the algorithms from [1] and [2]. These fairness methods use demographic parity and equalized odds to evaluate fairness, which differs from our Rawlsian approach. These methods aim to ensure that the model accuracy across demographic groups is comparable, however, these fairness definitions have several drawbacks. The first is that they do not prioritize accuracy for the worst-off group, so using these fairness definitions could lead to situations where the worst-off group has a low prediction accuracy. Another consequence of these fairness metrics is that they could result in low overall prediction accuracy because the algorithm could hurt the performance of groups that are not worst-off to ensure similar performance to the worst-off group.

In our comparison, we discuss the implications these alternative fairness metrics have on both the overall accuracy and the accuracy for the worst-off group. More details on the implementation of these methodologies are provided by [110].

First, we compare our results to the method proposed by [1]. Their method is a post-processing algorithm that is compatible with any classification algorithm that assigns a score value to each data point. After executing the classification algorithm, their methodology uses a different threshold value for each demographic group to determine whether each individual is classified as a 0 or 1. This threshold value is randomized, so its value could vary from individual to individual, and it is designed to satisfy either demographic parity or equalized odds. Demographic parity requires that a decision be independent of the demographic group, and equalized odds requires that the true and false positive rates are equal across demographic groups.

Our methodology implicitly finds the optimal threshold value as opposed to a randomized sub-optimal threshold value used by [1]. Another advantage is that by using only one singular threshold, we can access similar information across different demographic groups. By using a separate threshold value for each group, the method from [1] fails to capture shared characteristics of individuals across different demographic groups to aid in decision-making. Moreover, they use fairness metrics that enforce similar performance across different groups. This leads to outcomes where the worst-off group is not improved since hurting other groups to make their performance similar to the worst-off group could achieve fairness according to their definitions. This not only fails to help improve the

worst-off group's performance, but it unnecessarily hurts other groups besides the worst-off group to satisfy their fairness constraints. We verified these properties empirically and we tested the method from [1] with different combinations of utility functions and fairness constraints, and the results are shown below in Table 4.3.

| no. | obj. | constraint | train error (%) | train fairness (%) | test error (%) | test fairness (%) |
|---|---|---|---|---|---|---|
| 1 | AS | EO | 9.48 | 37.05 | 9.72 | 37.90 |
| 2 | AS | DP | 9.42 | 34.92 | 9.70 | 36.04 |
| 3 | BAS | DP | 22.92 | 27.38 | 24.25 | 29.10 |
| 4 | BAS | EO | 31.43 | 31.65 | 32.17 | 32.79 |
| 5 | SR | DP | 9.88 | 38.20 | 9.77 | 38.24 |
| 6 | TPR | DP | 9.88 | 38.20 | 9.77 | 38.24 |

Table 4.3: Results from [1].

We consider different combinations of classification objectives. The objectives AS, BAS, SR, TPR, and TNR represent accuracy score, balanced accuracy score, selection rate, true positive rate, and true negative rate, respectively. The fairness constraints are DP and EO, which represent demographic parity and equalized odds, respectively.

For experiments 1, 2, 5, and 6, the error is comparable to our results, but the fairness is worse. In these instances, their method achieves good overall classification accuracy, but they sacrifice the worst-off group's performance, which is at least 8% worse than our results. In experiments 3 and 4, they achieve similar fairness to our results, however, their overall error is much higher and worse than our results by at least 12%. Rather than improving the worst-off group to achieve demographic parity or equalized odds, they worsen the performance of the other groups (the non-worst-off groups), leading to an overall decrease in accuracy. In contrast, our method can improve the performance of the worst-off group without experiencing such a dramatic increase in overall error.

The method in [2] method is an in-processing approach, so it is more similar to our methodology. Their objective is to minimize the empirical error with respect to some random classifier subject to demographic parity (which is defined in the same way as [1]) constraints. They use Lagrangian reformulation to formulate a saddle point problem, then use grid search as a way of smartly searching

the feasible region. As a result of the grid search algorithm, their methodology can produce multiple solutions. The result of their grid-search algorithm is displayed below in Table 4.4.

| exp. no. | training error (%) | training fairness (%) | testing error (%) | testing fairness (%) |
|---|---|---|---|---|
| 1 | 9.18 | 29.27 | 9.18 | 29.27 |
| 2 | 8.82 | 27.38 | 9.26 | 29.61 |
| 3 | 8.89 | 27.54 | 9.21 | 29.95 |
| 4 | 9.06 | 30.16 | 9.14 | 29.61 |
| 5 | 9.15 | 30.82 | 9.26 | 31.81 |
| 6 | 9.31 | 32.13 | 9.33 | 33.33 |
| 7 | 9.50 | 34.59 | 9.43 | 34.52 |
| 8 | 9.66 | 36.07 | 9.57 | 35.87 |
| 9 | 9.69 | 36.39 | 9.62 | 36.55 |
| 10 | 9.79 | 36.89 | 9.66 | 37.06 |
| 11 | 9.83 | 36.89 | 9.67 | 37.06 |
| 12 | 9.89 | 37.70 | 9.69 | 37.56 |
| 13 | 9.90 | 37.87 | 9.72 | 37.90 |
| 14 | 9.92 | 38.20 | 9.73 | 38.07 |
| 15 | 9.93 | 38.36 | 9.75 | 38.41 |
| 16 | 9.93 | 38.36 | 9.75 | 38.41 |

Table 4.4: Results from [2]

The grid search approach allows the algorithm to produce multiple results which gives the user a variety of options to choose from. A few of the best results in terms of prediction error from Table 4.4 are comparable to our results from Table 4.2, but our model outperforms the majority of these results. In an effort to ensure that the worst-off group has comparable performance to the other groups, the algorithm in [2] unnecessarily hurts the performance of the other groups, thus decreasing the overall prediction accuracy.

Another difference in our results is that we achieve better performance for the worst-off group than all of the results produced from the grid search algorithm displayed in Table 4.4. The algo-

rithm in [2] does not prioritize the worst-off group's performance since they try to achieve similar performance across all groups. On the other hand, our method achieves a similar error overall, but it does so in a way that does not increase the error for the worst-off group. Therefore, our algorithm is better equipped to ensure that the minority group's performance is prioritized without sacrificing the overall classification accuracy.

Some works use the same means of measuring fairness as our Rawlsian fairness method, however their methods are incomparable to ours for numerous reasons. [91] proposes a way to minimize the error rate for the worst-off group, however, their solution methodology requires the second-order statistics of the underlying distribution to be known, whereas our method does not require this information to be available. [92] also aims to minimize their loss function for the worst-off group, however, they assume that the demographic group that each individual belongs to is unknown, which is different than the setting we consider. [90] also formulates a minimax objective like ours, but they consider a dynamic setting, where the data can change from one period to the next, whereas we consider a static case where the algorithm is only solved once. [93] also proposes a method to reduce classification error for the worst-off group, and they produce results that are comparable to ours on the same Law Student Dataset. However, they make assumptions on the minimum "partition density", which corresponds to the proportion a certain group makes up in the dataset. Our method, on the other hand, makes no such assumption on the structure of the data, which allows our methodology to handle any dataset, regardless of how unbalanced it is.

### *4.4 Conclusion*

We proposed a novel Rawlsian MIP formulation to address the fair classification problem. Our Rawlsian fairness method allowed our model to prioritize the worst-off demographic group's performance without unnecessarily harming the other groups. Combining this method with a MIP allowed us to leverage several advantages that MIPs offer. Our model directly optimizes accuracy and sparsity while providing model flexibility that enhances interpretability by allowing the user to adjust sparsity through penalization in the objective or with upper-bounding constraints. Our model also offers flexibility to obtain real or integer coefficients based on the user's domain-based preferences. Furthermore, our model outputs a single optimal threshold, which alleviates the user from selecting a threshold value before implementing the model. Consequently, our model provides deterministic classification results instead of probability values assigned to each individual.

We tested our methodology on two datasets, including one unbalanced dataset. Our experimen-

tal results displayed our Rawlsian MIP's ability to prioritize the performance of the worst-off group while still maintaining a high classification accuracy overall, even when performed on an unbalanced dataset. We also showcased the flexibility of our method in its ability to adjust the sparsity and interpretability of the results. Furthermore, we demonstrated the benefits of our method's ability to compute one single optimal threshold value compared to [1], which uses a different randomized threshold for each demographic group in a post-processing approach. Additionally, we showed how our Rawlsian approach outperforms the alternative fairness methods incorporated in [1] and [2].

# Chapter 5
# CONCLUSION

## *5.1 Summary*

We proposed a multi-stage distributionally robust optimization framework to support the real-time operation of a virtual power plant. Our framework is robust to distributional ambiguity, which provides the decision-maker with a risk-averse policy that is robust to situations when the behavior of randomness is unknown. The multi-stage aspect of our algorithm allows the decision-maker to adjust the dispatching strategy in real-time after observing the uncertainty in the previous period. Our experimental results showed our method's ability to outperform both a dynamic stochastic programming method and a static-DRO method.

We also proposed a target-oriented robust optimization method to evaluate the flexibility of grid operations and tested our methodology on a virtual power plant. Our metric evaluated the maximum deviation of non-renewable energy from its expected value that the system can tolerate. We proposed a primal generation algorithm that can solve the flexibility assessment problem more efficiently compared to a standard column and constraint generation approach.

Lastly, we proposed a classification methodology using a mixed-integer-program (MIP) formulation that incorporates a Rawlsian fairness objective. Our methodology shows the ability to provide flexible modeling capabilities, directly optimize sparsity, and handle unbalanced datasets. We also tested our method on different datasets and compared it to multiple state-of-the-art fair machine learning methods.

## *5.2 Extensions to Current Work*

An extension to the multi-stage distributionally robust methodology could include developing a model that considers continuous random scenarios instead of discrete ones. Another possible extension could be including unit commitment decisions for conventional generators, which would introduce integer variables into our framework.

We could extend our target-oriented robust optimization methodology by considering different flexibility metrics. Our current method optimizes total flexibility across all time periods. Instead of considering a hyperbox uncertainty set to describe the range of uncertainties that the system can

tolerate, we could use an ellipsoidal uncertainty set that captures the inter-temporal dependence of the system. We could also consider the use of distributionally robust optimization to consider the worst-case probability distribution of net load. This method would allow us to incorporate distributional information and would allow the decision maker to adjust the conservativeness level of the solution.

Lastly, extensions to our Rawlsian fairness framework could include further leveraging the flexibility that MIP formulations provide by including more complex fairness metrics. Examples include different linear fairness metrics mentioned in [111] or lexicographical fairness [112], which prioritizes performance for the worst-case group, but additionally aims to improve the overall accuracy as a secondary objective. We could also incorporate our Rawlsian framework with other classification methods, such as the support vector machine approach used in [113] to address optimal power flow flexibility.

### *5.3 Future Research Areas*

Fairness concerns in energy is an issue that is receiving more and more attention. From a policy standpoint, the federal government in 2021 made it a goal that 40 percent of the overall benefits of certain Federal investments flow to disadvantaged communities that are marginalized, underserved, and overburdened by pollution through the Justice40 Initiative [114]. Furthermore, several resources that track energy-related disparity data at the state and federal levels have emerged [115][116]. Energy-related fairness is complex and there are multiple perspectives to approach this issue. [117] defines two different definitions: energy justice and energy equity. They define energy justice as the goal of achieving equity in both the social and economic participation in the energy system, while also remediating social, economic, and health burdens on those disproportionately harmed by the energy system. Examples of energy justice include distributed, procedural, recognition, and restorative justice. They define energy equity as unlocking access to energy for everyone to realize the goal of energy justice. Energy equity-related issues include energy burden, poverty, insecurity, and vulnerability. More details on these definitions can be found in [117].

Gini coefficient is the most commonly used metric to address energy-related fairness. However, it can fail to address the disparity between the ratio of the income share held by the richest to the poorest individuals [118]. Generalized Entropy [119] and Atkinson Index [120] are popular metrics used to evaluate energy-related inequality, however, they are not commonly used to optimize fairness. Therefore, there is a need to develop new fairness metrics that can model inequalities

accurately and can also be optimized tractably. The Kolm-Pollak Equally Distributed Equivalent has been used to solve an equitable facility location problem in [121]. This metric has the potential to address the drawbacks of the previously mentioned metrics in energy-related fairness settings.

Fairness has been well studied regarding the reduction of carbon emissions and pollution, energy consumption, expenditure, and supply, as well as renewable energy accessibility. However, there have been limited approaches that address fairness in grid reliability. [122] studies this issue, however, they only provide methods to evaluate fairness, without providing a way to optimize it. A method that optimizes fairness with respect to grid reliability could play an important role in ensuring disadvantaged communities can safely receive energy in an environment with heightened uncertainty from increased extreme weather events and renewable energy usage. There are several challenges to incorporating fairness metrics in grid reliability. Firstly, with any grid reliability problem, there are physical constraints that define the system, which cannot be violated. Thus, from a modeling perspective, incorporating fairness objectives or constraints in addition to these physical constraints could be difficult. Secondly, it could be challenging to ensure fairness in an environment with a high degree of stochasticity from renewable energy resources. Thirdly, as mentioned above in [117], there are many different ways to define energy-related fairness. Grid reliability can impact different communities in various ways, so identifying the most appropriate fairness definition to use in evaluating and addressing inequalities is non-trivial. Finally, a fairness-efficiency trade-off is known to exist [123] [124], where increasing fairness could also increase system cost. Multi-objective optimization methods could be employed to address the challenge of finding a solution that ensures fairness without incurring significant costs.

# BIBLIOGRAPHY

[1] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018.

[3] Danny Pudjianto, Chin Kim Gan, Vladimir Stanojevic, Marko Aunedi, Predrag Djapic, and Goran Strbac. Value of integrating distributed energy resources in the uk electricity system. In *IEEE PES general meeting*, pages 1–6. IEEE, 2010.

[4] Qianchuan Zhao, Ying Shen, and Mingyang Li. Control and bidding strategy for virtual power plants with renewable generation and inelastic demand in electricity markets. *IEEE Transactions on Sustainable Energy*, 7(2):562–575, 2015.

[5] Morteza Rahimiyan and Luis Baringo. Strategic bidding for a virtual power plant in the day-ahead and real-time markets: A price-taker robust optimization approach. *IEEE Transactions on Power Systems*, 31(4):2676–2687, 2015.

[6] Evaggelos G Kardakos, Christos K Simoglou, and Anastasios G Bakirtzis. Optimal offering strategy of a virtual power plant: A stochastic bi-level approach. *IEEE Transactions on Smart Grid*, 7(2):794–806, 2015.

[7] Unlocking customer value: the virtual power plant, 2010.

[8] Zhongkai Yi, Yinliang Xu, Jianguo Zhou, Wenchuan Wu, and Hongbin Sun. Bi-level programming for optimal operation of an active distribution network with multiple virtual power plants. *IEEE Transactions on Sustainable Energy*, 11(4):2855–2869, 2020.

[9] Wen Chen, Jing Qiu, and Qingmian Chai. Customized critical peak rebate pricing mechanism for virtual power plants. *IEEE Transactions on Sustainable Energy*, 12(4):2169–2183, 2021.

[10] Virtual power plants offer a climate-forward response to hotter summers. https://www.greenbiz.com/article/virtual-power-plants-offer-climate-forward-response-hotter-summers, August 2023.

[11] Electricity in the u.s. https://www.eia.gov/energyexplained/electricity/electricity-in-the-us.php, July 2022.

[12] Rachel Ramirez. Power outages are on the rise, led by texas, michigan and california. here's what's to blame. https://www.cnn.com/2022/09/14/us/power-outages-rising-extreme-weather-climate/index.html, Sep 2022.

[13] California independent system operator: Today's outlook. https://www.caiso.com/TodaysOutlook/Pages/index.html.

[14] Manasa Ramakrishnan. Classification problems in machine learning: Examples, November 2023.

[15] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *Ethics of Data and Analytics: Concepts and Cases*, page 254, 2022.

[16] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[17] Hrvoje Pandžić, Juan M Morales, Antonio J Conejo, and Igor Kuzle. Offering model for a virtual power plant based on stochastic programming. *Applied Energy*, 105:282–292, 2013.

[18] Sadra Babaei, Chaoyue Zhao, and Lei Fan. A data-driven model of virtual power plants in day-ahead unit commitment. *IEEE Transactions on Power Systems*, 34(6):5125–5135, 2019.

[19] Diego Alvarado, Alexandre Moreira, Rodrigo Moreno, and Goran Strbac. Transmission network investment with distributed energy resources and distributionally robust security. *IEEE Transactions on Power Systems*, 34(6):5157–5168, 2018.

[20] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[21] Ali Ghahgharaee Zamani, Alireza Zakariazadeh, and Shahram Jadid. Day-ahead resource scheduling of a renewable energy based virtual power plant. *Applied Energy*, 169:324–340, 2016.

[22] Zhongkai Yi, Yinliang Xu, Wei Gu, and Wenchuan Wu. A multi-time-scale economic scheduling strategy for virtual power plant based on deferrable loads aggregation and disaggregation. *IEEE Transactions on Sustainable Energy*, 11(3):1332–1346, 2019.

[23] Shahrzad Hadayeghparast, Alireza SoltaniNejad Farsangi, and Heidarali Shayanfar. Day-ahead stochastic multi-objective economic/emission operational scheduling of a large scale virtual power plant. *Energy*, 172:630–646, 2019.

[24] Ali Ghahgharaee Zamani, Alireza Zakariazadeh, Shahram Jadid, and Ahad Kazemi. Stochastic operational scheduling of distributed energy resources in a large scale virtual power plant. *International Journal of Electrical Power & Energy Systems*, 82:608–620, 2016.

[25] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[26] Mohammad Hossein Abbasi, Mehrdad Taki, Amin Rajabi, Li Li, and Jiangfeng Zhang. Coordinated operation of electric vehicle charging and wind power generation as a virtual power plant: A multi-stage risk constrained approach. *Applied Energy*, 239:1294–1307, 2019.

[27] Ricardo M Lima, Antonio J Conejo, Sabique Langodan, Ibrahim Hoteit, and Omar M Knio. Risk-averse formulations and methods for a virtual power plant. *Computers & Operations Research*, 96:350–373, 2018.

[28] Mohammadreza Emarati, Farshid Keynia, and Masoud Rashidinejad. A two-stage stochastic programming framework for risk-based day-ahead operation of a virtual power plant. *International Transactions on Electrical Energy Systems*, 30(3):e12255, 2020.

[29] Mohammad Amin Tajeddini, Ashkan Rahimi-Kian, and Alireza Soroudi. Risk averse optimal operation of a virtual power plant using two stage stochastic programming. *Energy*, 73:958–967, 2014.

[30] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. Robust optimization. In *Robust Optimization*. Princeton university press, 2009.

[31] Liwei Ju, Rui Zhao, Qinliang Tan, Yan Lu, Qingkun Tan, and Wei Wang. A multi-objective robust scheduling model and solution algorithm for a novel virtual power plant connected with power-to-gas and gas storage tank considering uncertainty and demand response. *Applied Energy*, 250:1336–1355, 2019.

[32] Haotian Zhao, Bin Wang, Zhaoguang Pan, Hongbin Sun, Qinglai Guo, and Yixun Xue. Aggregating additional flexibility from quick-start devices for multi-energy virtual power plants. *IEEE Transactions on Sustainable Energy*, 12(1):646–658, 2020.

[33] Morteza Shabanzadeh, Mohammad-Kazem Sheikh-El-Eslami, and Mahmoud-Reza Haghifam. The design of a risk-hedging tool for virtual power plants via robust optimization approach. *Applied Energy*, 155:766–777, 2015.

[34] Huishi Liang and Jin Ma. Data-driven resource planning for virtual power plant integrating demand response customer selection and storage. *IEEE Transactions on Industrial Informatics*, 18(3):1833–1844, 2021.

[35] Fang Fang, Songyuan Yu, and Xiuli Xin. Data-driven-based stochastic robust optimization for a virtual power plant with multiple uncertainties. *IEEE Transactions on Power Systems*, 37(1):456–466, 2021.

[36] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[37] Tao Ding, Xiaosheng Zhang, Runzhao Lu, Ming Qu, Mohammad Shahidehpour, Yuankang He, and Tianen Chen. Multi-stage distributionally robust stochastic dual dynamic programming to multi-period economic dispatch with virtual energy storage. *IEEE Transactions on Sustainable Energy*, 13(1):146–158, 2021.

[38] Zhang Yuanyuan, Zhao Huiru, and Li Bingkang. Distributionally robust comprehensive declaration strategy of virtual power plant participating in the power market considering flexible ramping product and uncertainties. *Applied Energy*, 343:121133, 2023.

[39] Huichuan Liu, Jing Qiu, and Junhua Zhao. A data-driven scheduling model of virtual power plant using Wasserstein distributionally robust optimization. *International Journal of Electrical Power & Energy Systems*, 137:107801, 2022.

[40] Songyuan Yu, Fang Fang, and Jizhen Liu. Flexible operation of a chp-vpp considering the coordination of supply and demand based on a strengthened distributionally robust optimization. *IET Control Theory & Applications*, 2023.

[41] Huiru Zhao, Xuejie Wang, Zhuoya Siqin, Bingkang Li, and Yuwei Wang. Two-stage optimal dispatching of multi-energy virtual power plants based on chance constraints and data-driven distributionally robust optimization considering carbon trading. *Environmental Science and Pollution Research*, pages 1–21, 2023.

[42] Runzhao Lu, Tao Ding, Boyu Qin, Jin Ma, Xin Fang, and Zhaoyang Dong. Multi-stage stochastic programming to joint economic dispatch for energy and reserve with uncertain renewable energy. *IEEE Transactions on Sustainable Energy*, 11(3):1140–1151, 2019.

[43] Haoxiang Yang and Harsha Nagarajan. Optimal power flow in distribution networks under stochastic n- 1 disruptions. *Electric Power Systems Research*, 189:106689, 2020.

[44] Chixin Xiao, Danny Sutanto, Kashem M Muttaqi, and Minjie Zhang. Multi-period data driven control strategy for real-time management of energy storages in virtual power plants integrated with power grid. *International Journal of Electrical Power & Energy Systems*, 118:105747, 2020.

[45] Liwei Ju, Zhongfu Tan, Jinyun Yuan, Qingkun Tan, Huanhuan Li, and Fugui Dong. A bi-level stochastic scheduling optimization model for a virtual power plant connected to a wind–photovoltaic–energy storage system considering the uncertainty and demand response. *Applied energy*, 171:184–199, 2016.

[46] Fred Lambert. Tesla virtual power plant is rocketing up, reaches 50 mw, 2022.

[47] Kavya Balaraman. Sunrun, pg&e to roll out 30-mw virtual power plant to support california grid in the summer, 2023.

[48] Daniel Duque and David P Morton. Distributionally robust stochastic dual dynamic programming. *SIAM Journal on Optimization*, 30(4):2841–2865, 2020.

[49] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[50] Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[51] Victor M Panaretos and Yoav Zemel. *An invitation to statistics in Wasserstein space*. Springer Nature, 2020.

[52] Sheldon M Ross. *Introduction to stochastic dynamic programming*. Academic Press, 2014.

[53] Adam B Birchfield, Ti Xu, Kathleen M Gegner, Komal S Shetye, and Thomas J Overbye. Grid structural characteristics as validation criteria for synthetic networks. *IEEE Transactions on power systems*, 32(4):3258–3265, 2016.

[54] Xin Chen and Na Li. Leveraging two-stage adaptive robust optimization for power flexibility aggregation. *IEEE Transactions on Smart Grid*, 12(5):3954–3965, 2021.

[55] Eamonn Lannoye, Damian Flynn, and Mark O'Malley. Evaluation of power system flexibility. *IEEE Transactions on Power Systems*, 27(2):922–931, 2012.

[56] Juan Ma, Vera Silva, Régine Belhomme, Daniel S Kirschen, and Luis F Ochoa. Exploring the use of flexibility indices in low carbon power systems. In *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–5. IEEE, 2012.

[57] Anupam A Thatte and Le Xie. A metric and market construct of inter-temporal flexibility in time-coupled economic dispatch. *IEEE Transactions on Power Systems*, 31(5):3437–3446, 2015.

[58] Zheyu Guo, Yanan Zheng, and Gengyin Li. Power system flexibility quantitative evaluation based on improved universal generating function method: A case study of zhangjiakou. *Energy*, 205:117963, 2020.

[59] Felix Garcia-Torres, Carlos Bordons, Javier Tobajas, Rafael Real-Calvo, Isabel Santiago, and Stephane Grieu. Stochastic optimization of microgrids with hybrid energy storage systems for grid flexibility services considering energy forecast uncertainties. *IEEE Transactions on Power Systems*, 36(6):5537–5547, 2021.

[60] Kyriaki Antoniadou-Plytaria, David Steen, Ola Carlson, Baraa Mohandes, Mohammad Ali Fotouhi Ghazvini, et al. Scenario-based stochastic optimization for energy and flexibility dispatch of a microgrid. *IEEE Transactions on Smart Grid*, 13(5):3328–3341, 2022.

[61] Cheng Wang, Feng Liu, Jianhui Wang, Feng Qiu, Wei Wei, Shengwei Mei, and Shunbo Lei. Robust risk-constrained unit commitment with large-scale wind generation: An adjustable uncertainty set approach. *IEEE Transactions on Power Systems*, 32(1):723–733, 2016.

[62] IF Abdin, Y-P Fang, and Enrico Zio. A modeling and optimization framework for power systems design with operational flexibility and resilience against extreme heat waves and drought events. *Renewable and Sustainable Energy Reviews*, 112:706–719, 2019.

[63] Ruiwei Jiang, Jianhui Wang, and Yongpei Guan. Robust unit commitment with wind power and pumped storage hydro. *IEEE Transactions on Power Systems*, 27(2):800–810, 2011.

[64] Jinye Zhao, Tongxin Zheng, and Eugene Litvinov. A unified framework for defining and measuring flexibility in power system. *IEEE Transactions on power systems*, 31(1):339–347, 2015.

[65] Mohammad Ali Sadeghi, Mehdi Jafari Shahbazzadeh, Amir Abdollahi, Mahdiyeh Eslami, and MohammadIman Alizadeh. Robust flexibility driven security constrained unit commitment under wind uncertainty considering demand response and combined-cycle units. *International Journal of Electrical Power & Energy Systems*, 137:107814, 2022.

[66] Yun Fong Lim and Chen Wang. Inventory management based on target-oriented robust optimization. *Management Science*, 63(12):4409–4427, 2017.

[67] Lei Fan, Chaoyue Zhao, Guangyuan Zhang, and Qiuhua Huang. Flexibility management in economic dispatch with dynamic automatic generation control. *IEEE Transactions on Power Systems*, 37(2):876–886, 2021.

[68] Chengcheng Shao, Xifan Wang, Mohammad Shahidehpour, Xiuli Wang, and Biyang Wang. Security-constrained unit commitment with flexible uncertainty set for variable wind power. *IEEE Transactions on Sustainable Energy*, 8(3):1237–1246, 2017.

[69] Cheng Wang, Feng Liu, Jianhui Wang, Wei Wei, and Shengwei Mei. Risk-based admissibility assessment of wind generation integrated into a bulk power system. *IEEE Transactions on Sustainable Energy*, 7(1):325–336, 2015.

[70] Farzaneh Pourahmadi, Seyed Hamid Hosseini, and Mahmud Fotuhi-Firuzabad. Economically optimal uncertainty set characterization for power system operational flexibility. *IEEE transactions on industrial informatics*, 15(10):5456–5465, 2019.

[71] Bo Zeng and Long Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

[72] Chaoyue Zhao and Yongpei Guan. Unified stochastic and robust unit commitment. *IEEE Transactions on Power Systems*, 28(3):3353–3361, 2013.

[73] Farzaneh Pourahmadi, Seyed Hamid Hosseini, Payman Dehghanian, Ekundayo Shittu, and Mahmud Fotuhi-Firuzabad. Uncertainty cost of stochastic producers: Metrics and impacts on power grid flexibility. *IEEE Transactions on Engineering Management*, 69(3):708–719, 2020.

[74] Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the international workshop on software fairness*, pages 1–7, 2018.

[75] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, accountability and transparency*, pages 107–118. PMLR, 2018.

[76] John Rawls. *A theory of justice: Revised edition*. Harvard university press, 1999.

[77] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017.

[78] Nathanael Jo, Sina Aghaei, Jack Benson, Andres Gomez, and Phebe Vayanos. Learning optimal fair decision trees: Trade-offs between interpretability, fairness, and accuracy. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 181–192, 2023.

[79] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

[80] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Conference on Learning Theory*, pages 1920–1953. PMLR, 2017.

[81] Arvind Kumar, Shivani Goel, Nishant Sinha, and Arpit Bhardwaj. A review on unbalanced data classification. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2021*, pages 197–208. Springer, 2022.

[82] Dohyun Lee and Kyoungok Kim. An efficient method to determine sample size in oversampling based on classification complexity for imbalanced data. *Expert Systems with Applications*, 184:115442, 2021.

[83] Bin Liu and Grigorios Tsoumakas. Dealing with class imbalance in classifier chains via random undersampling. *Knowledge-Based Systems*, 192:105292, 2020.

[84] Arvind Kumar, Nishant Sinha, and Arpit Bhardwaj. A novel fitness function in genetic programming for medical data classification. *Journal of Biomedical Informatics*, 112:103623, 2020.

[85] Elisa Ferrari and Davide Bacciu. Addressing fairness, bias and class imbalance in machine learning: the fbi-loss. *arXiv preprint arXiv:2105.06345*, 2021.

[86] Urvesh Bhowan, Mark Johnston, and Mengjie Zhang. Developing new fitness functions in genetic programming for classification with unbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):406–421, 2011.

[87] Karmel S Shehadeh. Reducing disparities in transportation distance in a stochastic facility location problem. *Transportation Research Part C: Emerging Technologies*, 153:104199, 2023.

[88] Xinfang Wang. Health service design with conjoint optimization. *Journal of the Operational Research Society*, 70(7):1091–1101, 2019.

[89] Aurora del Carmen Munguía-López and José María Ponce-Ortega. Fair allocation of potential covid-19 vaccines using an optimization-based strategy. *Process Integration and Optimization for Sustainability*, 5:3–12, 2021.

[90] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

[91] Kulin Shah, Pooja Gupta, Amit Deshpande, and Chiranjib Bhattacharyya. Rawlsian fair adaptation of deep learning classifiers. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 936–945, 2021.

[92] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33:728–740, 2020.

[93] Natalia L Martinez, Martin A Bertran, Afroditi Papadaki, Miguel Rodrigues, and Guillermo Sapiro. Blind pareto fairness and subgroup robustness. In *International Conference on Machine Learning*, pages 7492–7501. PMLR, 2021.

[94] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

[95] Toon Calders and Szymon Jaroszewicz. Efficient auc optimization for classification. In *European conference on principles of data mining and knowledge discovery*, pages 42–53. Springer, 2007.

[96] Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass auc optimization. In *International conference on machine learning*, pages 906–914. PMLR, 2013.

[97] Berk Ustun, Stefano Traca, and Cynthia Rudin. Supersparse linear integer models for predictive scoring systems. *arXiv preprint arXiv:1306.5860*, 2013.

[98] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 180(3):689–722, 2017.

[99] Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102:349–391, 2016.

[100] Carlos Ramirez, Vladik Kreinovich, and Miguel Argaez. Why 1 is a good approximation to 0: A geometric explanation. 2013.

[101] Yuansi Chen, Armeen Taeb, and Peter Bühlmann. A look at robustness and stability of 1-versus 0-regularization: Discussion of papers by bertsimas et al. and hastie et al. 2020.

[102] Yufeng Liu and Yichao Wu. Variable selection via a combination of the l 0 and l 1 penalties. *Journal of Computational and Graphical Statistics*, 16(4):782–798, 2007.

[103] Berk Ustun and Cynthia Rudin. Methods and models for interpretable linear classification. *arXiv preprint arXiv:1405.4047*, 2014.

[104] Thomas R Guskey. The case against percentage grades. *Educational Leadership*, 71(1):68–72, 2013.

[105] William A Knaus, Douglas P Wagner, Elizabeth A Draper, Jack E Zimmerman, Marilyn Bergner, Paulo G Bastos, Carl A Sirio, Donald J Murphy, Ted Lotring, Anne Damiano, et al. The apache iii prognostic system: risk prediction of hospital mortality for critically iii hospitalized adults. *Chest*, 100(6):1619–1636, 1991.

[106] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR, 2019.

[107] Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*, 2020. R package version 0.1.0.

[108] Linda F Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

[109] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(3):e1452, 2022.

[110] Hilde Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. Fairlearn: Assessing and improving fairness of ai systems, 2023.

[111] Man Yiu Tsang and Karmel S Shehadeh. Convex fairness measures: Theory and optimization. *arXiv preprint arXiv:2211.13427*, 2022.

[112] John N Hooker and H Paul Williams. Combining equity and utilitarianism in a mathematical programming model. *Management Science*, 58(9):1682–1693, 2012.

[113] Wengeng Pan, Chaoyue Zhao, Lei Fan, and Shuai Huang. Efficient optimal power flow flexibility assessment: A machine learning approach. In *2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2023.

[114] The White House. Justice40. `https://www.whitehouse.gov/environmentaljustice/justice40/`, July 2021.

[115] Colorado Department of Health and Environment. Colorado enviroscreen. `https://cdphe.colorado.gov/enviroscreen`, June 2022.

[116] National Renewable Energy Laboratory. Slope: State and local planning for energy. `https://maps.nrel.gov/slope`, 2023.

[117] Weihang Ren, Yongpei Guan, Feng Qiu, Todd Levin, and Miguel Heleno. A literature review of energy justice. *arXiv preprint arXiv:2312.14983*, 2023.

[118] Thitithep Sitthiyot and Kanyarat Holasut. A simple method for measuring inequality. *Palgrave Communications*, 6(1):1–9, 2020.

[119] Nicholas Rohde. Lorenz curves and generalised entropy inequality measures. In *Modeling income distributions and Lorenz curves*, pages 271–283. Springer, 2008.

[120] Marta Pascual Sáez, David Cantarero-Prieto, and Jose R Pires Manso. Gross inland energy consumption inequality in europe: An empirical approach. *Revista de economía mundial*, (49), 2018.

[121] Drew Horton, Tom Logan, Joshua Murrell, Daphne Skipper, and Emily Speakman. A scalable approach to equitable facility location. *arXiv preprint arXiv:2401.15452*, 2024.

[122] Evelyn Heylen, Marten Ovaere, Stef Proost, Geert Deconinck, and Dirk Van Hertem. Fairness and inequality in power system reliability: Summarizing indices. *Electric Power Systems Research*, 168:313–323, 2019.

[123] Destenie Nock, Todd Levin, and Erin Baker. Changing the policy paradigm: A benefit maximization approach to electricity planning in developing countries. *Applied Energy*, 264:114583, 2020.

[124] Jan-Philipp Sasse and Evelina Trutnevyte. Distributional trade-offs between regionally equitable and cost-efficient allocation of renewable electricity generation. *Applied Energy*, 254:113724, 2019.