Pontifícia Universidade Católica
do Rio de Janeiro

Joaquim Masset Lacombe Dias Garcia

The Effectiveness of Bilevel Optimization in Large-Scale Power Systems Problems

A Bilevel Optimization Toolbox, a Framework for Application-Driven Learning, and a Market Simulator

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica, do Departamento de Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica.

Advisor : Prof. Alexandre Street de Aguiar
Co-advisor: Dr. Mario Veiga Ferraz Pereira

Rio de Janeiro
December 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Joaquim Masset Lacombe Dias Garcia

The Effectiveness of Bilevel Optimization in Large-Scale Power Systems Problems

A Bilevel Optimization Toolbox, a Framework for Application-Driven Learning, and a Market Simulator

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the Examination Committee:

Prof. Alexandre Street de Aguiar
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Dr. Mario Veiga Ferraz Pereira
Co-Advisor
PSR, Brazil

Prof. Anthony Papavasiliou
National Technical University of Athens, Greece

Prof. Bernardo Freitas Paulo da Costa
Universidade Federal do Rio de Janeiro, Brazil

Prof. Bruno Fânzeres dos Santos
Departamento de Engenharia Industrial – PUC-Rio, Brazil

Dr. David Pozo
European Commission, Joint Research Centre, Netherlands

Prof. Tito Homem-de-Mello
Universidad Adolfo Ibáñez, Chile

Rio de Janeiro, December the 9th, 2022

Joaquim Masset Lacombe Dias Garcia

Majored in Electrical Engineering with an emphasis in Telecommunications at the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) in 2016, with one year at the University of California at Santa Barbara. Majored in Mathematics at the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) in 2017. During undergraduate studies, Joaquim worked at LAMPS and LabOpto, both at PUC-Rio. Currently, he works at PSR, where Joaquim is a researcher and developer working with game theory, bilevel optimization and stochastic optimization. In PSR, Joaquim also works with high-performance computing and developing PSR analytical tools. Joaquim is also a core contributor to the JuMP modeling language.

Bibliographic data

Dias Garcia, Joaquim Masset Lacombe

The effectiveness of bilevel optimization in large-scale power systems problems - a bilevel optimization toolbox, a framework for application-driven learning, and a market simulator / Joaquim Masset Lacombe Dias Garcia; advisor: Alexandre Street de Aguiar; co-advisor: Mario Veiga Ferraz Pereira. – 2022.

150 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2022.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Otimização Estocástica. 3. Otimização Binível. 4. Linguagem Julia. 5. Economia dos Sistemas de Potência. 6. Aprendizado de Máquina. I. Street, Alexandre. II. Pereira, Mario. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Para B, JP, J, P, LG, R e LA

Acknowledgments

First and foremost, I would like to thank Beatriz for all the companionship and support of my friend, partner, girlfriend, and wife, Beatriz, who helped me with absolutely everything during my doctoral years. Your support, affection, attention, help, and lots of love made this journey much more peaceful and pleasant.

I would like to thank my family for their constant support, motivation, excitement, interest, and much love. My parents, Patricia and Joaquim, gave me every opportunity and always encouraged me. My grandparents are all examples that contributed a lot to my path. Helena always motivates my curiosity and willingness to explore everything. Maria Olivia always brings the whole family together to give more strength to everyone. Carlos José, the most creative person I know always excited about my academic developments. Luiz Antônio, is an incredible example of kindness and goodness, unfortunately, he could not see the end of this journey here on earth for a few months, but he has certainly seen it from wherever he is. My brother João Pedro who was always by my side at all times, always ready to do anything for me. Luiz Gabriel and Rafael who are my hope for the future of this world. Maria Teresa and Steve who have been so supportive over the years. My cousins Maria, Luiza and Carol for their company and support.

Thanks to my dear friends who have always been by my side on this journey. Especially when I needed to work on this thesis.

I would like to thank my advisor and great friend Alexandre Street. Thank you for your company and guidance on this long journey that began as a scientific initiation project in the electrical engineering graduation. Since then, we've worked hard, and I've learned how to write, present, research, converse, and much more. He was a sensational guide on this trip, and I could not have chosen someone better to develop this work with.

I thank Mario Veiga, my advisor, supervisor, friend, and one of my heroes. Mario shows every day how to do top-quality science. Since my first day of internship at PSR, Mario has taught me to be a better researcher, developer, professional, and person. Our daily discussions motivated the works of this thesis, among many others. Thank you for introducing me to JuMP, which was essential throughout the thesis.

Thanks to my incredible colleagues at PUC and LAMPS. Special thanks to Raphael Sampaio, Gustavo Amaral, Bruno Fânzeres, Alexandre Moreira, Guilherme Bodin and Andrew Rosemberg for all the years of work and research of the highest quality. Even more special thanks to Guilherme Bodin, who

was excited by the idea of developing BilevelJuMP and consequently was instrumental in chapter 2.

I thank my professors at PUC Rio. In particular, I thank my professor Carlos Tomei, who made me discover the beauty in mathematics, put me on the path of optimization, introduced me to Alexandre Street and Mario Veiga, and teaches me, until today, mathematics, research, and much more. I also thank Carlos Kubrusly, Marcus Poggi, Thibaut Vidal, José Mauro Fortes, and Guilherme Temporão for the fantastic classes and guidance that were essential steps in my training.

Thanks to my colleagues at PSR, who have been an integral part of this journey. A complete and qualified team is essential to make work like this possible. Discussions on mathematics and the electricity sector, computational infrastructure, and high-quality concrete information about systems make PSR a unique place of innovation capacity. Special thanks to Sergio Granville, who taught me the basics of bilevel optimization, and for the many discussions in chapter 4, to Gabriel Cunha and Erick Foti, who helped me to assemble the database, adjust the modeling and test the model in chapter 4, to Luiz Barroso who, in addition to being an example and academic guide, was essential in the discussions in chapter 4.

I am thankful to my co-authors of chapter 3, Tito and Francisco, with whom I and Alexandre Street had great discussions. I learned a lot of maths and power systems modeling with you.

I am thankful to all the members of the Jury, who carefully read the entire thesis e provided amazing and valuable feedback. Thanks to them, this thesis was greatly improved. Thanks Anthony Papavasiliou, Bernardo Freitas Paulo da Costa, Bruno Fânzeres dos Santos, David Pozo, Tito Homem-de-Mello.

I am thankful to my fellow JuMP developers, especially to Miles Lubin, Benoît Legat and Oscar Dowson and the amazing group that has been contributing to the JuMP ecosystem. Special thanks to Mathieu Besançon. Without JuMP, this thesis would not be possible.

Thanks to PUC-Rio and the Department of Electrical Engineering for supporting this work and for the encouraging atmosphere on the Gávea campus.

Thanks to the ANEEL Research and Development Program and Engie Brasil Energia S.A. for financing RD ANEEL PD-00403-0050/2020, which partially supported the development of chapter 4.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Dias Garcia, Joaquim; Street, Alexandre (Advisor); Pereira, Mario (Co-Advisor). **The Effectiveness of Bilevel Optimization in Large-Scale Power Systems Problems - A Bilevel Optimization Toolbox, a Framework for Application-Driven Learning, and a Market Simulator**. Rio de Janeiro, 2022. 150p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Bilevel Optimization is an extremely powerful tool for modeling realistic problems in multiple areas. On the other hand, Bilevel Optimization is known to frequently lead to complex or intractable problems. In this thesis, we present three works expanding the state of the art of bilevel optimization and its intersection with power systems. First, we present BilevelJuMP, a novel open-source package for bilevel optimization in the Julia language. The package is an extension of the JuMP mathematical programming modeling language, is very general, feature-complete, and presents unique functionality, such as the modeling of lower-level cone programs. The software enables users to model a variety of bilevel problems and solve them with advanced techniques. As a consequence, it makes bilevel optimization widely accessible to a much broader public. In the following two works, we develop specialized methods to handle much model complex and very large-scale bilevel programs arising from power systems applications. Second, we use bilevel programming as the foundation to develop Application-Driven Learning, a new closed-loop framework in which the processes of forecasting and decision-making are merged and co-optimized. We describe the model mathematically as a bilevel program, prove convergence results and describe exact and tailor-made heuristic solution methods to handle very large-scale systems. The method is applied to demand forecast and reserve allocation in power systems operation. Case studies show very promising results with good quality solutions on realistic systems with thousands of buses. Third, we propose a simulator to model long-term bid-based hydro-thermal power markets. A multi-stage stochastic

program is formulated to accommodate the dynamics inherent to hydropower systems. However, the subproblems of each stage are bilevel programs in order to model strategic agents. The simulator is scalable in terms of system data, agents, scenarios, and stages being considered. We conclude the third work with large-scale simulations with realistic data from the Brazilian power system with 3 price maker agents, 1000 scenarios, and 60 monthly stages. These three works show that although bilevel optimization is an extremely challenging class of NP-hard problems, it is possible to develop effective algorithms that lead to good-quality solutions.

Keywords

Stochastic Optimization; Bilevel Optimization; Julia Language; Power Systems Economics; Machine Learning.

Resumo

Dias Garcia, Joaquim; Street, Alexandre; Pereira, Mario. **A Eficácia da Otimização de Dois Níveis em Problemas de Sistemas de Potência de Grande Porte - Uma Ferramenta para Otimização de Dois Níveis, uma Metodologia para Aprendizado Dirigido pela Aplicação e um Simulador de Mercado**. Rio de Janeiro, 2022. 150p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

A otimização de binível é uma ferramenta extremamente poderosa para modelar problemas realistas em várias áreas. Por outro lado, sabe-se que a otimização de dois níveis frequentemente leva a problemas complexos ou intratáveis. Nesta tese, apresentamos três trabalhos que expandem o estado da arte da otimização de dois níveis e sua interseção com sistemas de potência. Primeiro, apresentamos BilevelJuMP, um novo pacote de código aberto para otimização de dois níveis na linguagem Julia. O pacote é uma extensão da linguagem de modelagem de programação matemática JuMP, é muito geral, completo e apresenta funcionalidades únicas, como a modelagem de programas cônicos no nível inferior. O software permite aos usuários modelar diversos problemas de dois níveis e resolvê-los com técnicas avançadas. Como consequência, torna a otimização de dois níveis amplamente acessível a um público muito mais amplo. Nos dois trabalhos seguintes, desenvolvemos métodos especializados para lidar com modelos complexos e programas de dois níveis de grande escala decorrentes de aplicações de sistemas de potência. Em segundo lugar, usamos a programação de dois níveis como base para desenvolver o Aprendizado Dirigido pela Aplicação, uma nova estrutura de ciclo fechado na qual os processos de previsão e tomada de decisão são mesclados e co-otimizados. Descrevemos o modelo matematicamente como um programa de dois níveis, provamos resultados de convergência e descrevemos métodos de solução heurísticos e exatos para lidar com sistemas de grande escala. O método é aplicado para previsão de demanda e alocação de reservas na operação de sistemas de potência. Estudos de caso mostram resultados muito promissores com soluções de boa qualidade

em sistemas realistas com milhares de barras. Em terceiro lugar, propomos um simulador para modelar mercados de energia hidrotérmica de longo prazo baseados em ofertas. Um problema de otimização estocástica multi-estágio é formulado para acomodar a dinâmica inerente aos sistemas hidrelétricos. No entanto, os subproblemas de cada etapa são programas de dois níveis para modelar agentes estratégicos. O simulador é escalável em termos de dados do sistema, agentes, cenários e estágios considerados. Concluímos o terceiro trabalho com simulações em grande porte com dados realistas do sistema elétrico brasileiro com 3 agentes formadores de preço, 1000 cenários e 60 estágios mensais. Esses três trabalhos mostram que, embora a otimização de dois níveis seja uma classe extremamente desafiadora de problemas NP-difíceis, é possível desenvolver algoritmos eficazes que levam a soluções de boa qualidade.

Palavras-chave

Otimização Estocástica; Otimização Binível; Linguagem Julia; Economia dos Sistemas de Potência; Aprendizado de Máquina.

Table of contents

List of figures

List of tables

List of Abreviations

API – Application Programming Interface

AML – Algebraic Modeling Language

AR – Auto Regressive

ARMA – Auto Regressive Moving Average

BFGS – Broyden–Fletcher–Goldfarb–Shanno algorithm

BO – Bilevel Optimization

BOP – Bilevel Optimization Problem

CCG – Column Constraint Generation

CP – Conic Programming

EMP – Extended Mathematical Programming

EPEC – Equilibrium Problem with Equilibrium Constraints

FCF – Future Cost Function

IEEE – Institute of Electrical and Electronics Engineers

ISO – Independent System Operator

KKT – Karush-Kuhn-Tucker Optimality Conditions

LEO – Learning Enabled Optimization

LS – Least Squares

LS-Ex – Least Squares Demand forecast with Exogenous Reserves

LS-Opt – Least Squares Demand forecast with Optimized Reserves

LP – Linear Programming

MILP – Mixed Integer Linear Programming

MIP – Mixed Integer Programming

MISOCP – Mixed Integer Second Order Cone Programming

ML – Machine Learning

MLE – Maximum Likelihood Estimation

MSO – Multistage Stochastic Optimization

MOI – MathOptInterface

MPEC – Mathematical Program with Equilibrium Constraints

NLP – Non-Linear Programming

NN – Neural Networks

NP – Non-Deterministic Polynomial Time

Opt-Ex – Optimization of Demands with Exogenous Reserves

Opt-Opt – Optimization of Demands and Reserves

QP – Quadratic Programming

SAA – Sample Average Approximation

SGD – Stochastic Gradient Descend

SDDP – Stochastic Dual Dynamic Programming

SDP – Stochastic Dynamic Programming

SL – Statistical Learning

SOS1 – Special Ordered Sets of Type I

SPO – Smart Predict and Optimize

SSDP – Sampling Stochastic Dynamic Programming

SVR – Support Vector Regression

UC – Unit Commitment

VI – Variational Inequality

I think that it is a relatively good approximation to ***truth — which is much too complicated to allow anything but approximations*** *— that mathematical ideas originate in empirics.*

John von Neumann, *The Mathematician, in Works of the Mind.*

1
Introduction

Bilevel Optimization (BO) is an extremely powerful class of optimization problems and a core subject of the three works in this thesis [1, 2, 3]. Such a framework can be used to describe unique mathematical models. Historically, BO is deeply related to game theory as it is equivalent to Stackelberg Games, in which agents interact strategically [4]. Moreover, it has proven to be useful for modeling parameter tunning problems in machine learning [5].

On the other hand, solving bilevel problems is very difficult. The first main reason is that bilevel optimization is inherently challenging as it is NP-Hard to solve general problems of this class [6]. Therefore, specialized algorithms are usually necessary to solve approximately large-scale instances of interest. Secondly, although some techniques are available in the literature to exactly solve BO problems, many (error-prone) steps are required on the user side to apply these methods because there is a limited number of modeling frameworks and solvers available.

The goal of this thesis is to demonstrate the effectiveness of bilevel optimization and present solution techniques that convert such complex models into practically tractable problems. By practically tractable, we mean a problem that can be NP-hard but can be solved within reasonable computational time, at least for meaningful instances that are relevant to real-world applications.

To achieve that goal, we present three works as follows. We start with an open-source software, or program, named BilevelJuMP:

[1] - Dias Garcia, J., Bodin, G., and Street, A. (2022). `BilevelJuMP.jl`: Modeling and Solving Bilevel Optimization in Julia. arXiv preprint arXiv:2205.02307.

A specialized library for bilevel optimization based on the JuMP library in the Julia language is detailed in this first work. The developed software allows users to easily formulate and solve bilevel optimization problems. We highlight that the software is very complete in terms of functionality, including features such as modeling conic optimization problems and representing the

dual variables in the model. Examples are presented to depict syntax and usage. This tool greatly reduces barriers between users and the BO technology, making it much more accessible to a broad range of users. Such users include both newcomers to the field of bilevel optimization willing to model their first problems and advanced users aiming to experiment and benchmark more sophisticated techniques.

In addition, we present a novel framework to find the best forecast for a given application:

[2] - Dias Garcia, J., Street, A., Homem-de-Mello, T., and Muñoz, F.D., 2021. Application-Driven Learning via Joint Prediction and Optimization of Demand and Reserves Requirement. arXiv preprint arXiv:2102.13273.

In which we describe a methodology to jointly predict demand and optimally allocate reserves based on the cost function of the actual application, that is, the re-scheduling of the power system in real-time operation. This is a closed-loop method in which a parametric forecast method is trained based on the final assessed cost from a real-time adaptation from a previous plan that was optimized based on the forecast. The mathematical model is a stochastic bilevel program that is solved exactly on small instances and sub-optimally on very large-scale instances, but in both cases, the results are superior than the benchmarks.

Finally, we develop a power system application that requires bilevel programming to model the strategic behavior of agents in a hydrothermal power market:

[3] - Dias Garcia, J., Street, A., and Pereira, M. (2022). Long-term Hydrothermal Bid-based Market Simulator with Case Studies in the Brazilian System. Unpublished.

This presents a simulator for a very challenging power market setting since the presence of hydropower plants adds complexity, such as time coupling and uncertainty, to the market optimization problems. We propose a methodology that is scalable in terms of the number of scenarios, stages and system representation. A case study with data from Brazil serves as background to demonstrate the capabilities of the tool.

The three works are fundamentally connected by having bilevel optimization at their core. The first work is mostly focused on bilevel optimization and describes software. Such software was used in the case studies of the second work and during the development of the third as a reference for testing approximations presented there. The second and third works contain power systems applications and are deeply connected to stochastic programming as well. Due to the large-scale nature of the problems in the two last works, special algorithms were developed to obtain good quality approximate solutions for practical problems.

In the following Chapters 2, 3 and 4, each work is presented in a completely self-contained fashion reproducing the corresponding publication. Conclusions and future research directions are summarized in Chapter 5. The references are unified.

The remainder of this introduction will briefly describe some fundamental concepts which are assumed as prerequisites by the three main works of this thesis. The goal of those sections is not to go into detail but to provide nomenclature and references for interested readers. In the last section of this chapter, we will present other works that were developed during the Ph.D. studies but are not part of this thesis.

1.1 Optimization

According to the Encyclopedia Britannica [7]:
"Optimization, also known as mathematical programming, collection of mathematical principles and methods used for solving quantitative problems in many disciplines, including physics, biology, engineering, economics, and business."

In a fairly general (and loose) setting, we consider the problem of finding the values of the vector of variables x for which the smallest possible value of an objective function f is attained, given constraints represented by the set X. The mathematical formulation of the previous sentence is:

$$\min_x f(x) \tag{1-1}$$

$$s.t.$$

$$x \in X \tag{1-2}$$

In the case where f is linear and X can be represented as affine inequalities,

we can write:

$$\min_x c^\top x \tag{1-3}$$

$$s.t.$$

$$Ax \leq b \tag{1-4}$$

which is known as Linear Programming (LP) and is one of the key workhorses of more general settings and applications. Linear programming dates back to the works of Dantzig [8, 9, 10] and Kantorovich [11], although Fourier had previously analysed the problem [8, 12]. Introductions to the subject can be found in [13] and [14]. Algorithms are very well developed for this class of problems, the most widely used being the Simplex Method and Interior Point Methods, both able to find globally optimal solutions very efficiently in practice.

A more complex case where f is still linear and X can be represented by an intersection of convex cones is known as Conic Programming (CP). [15], [16] and [17] are excellent introductions to the subject. Interior Point Methods generalized well to CP and are good theoretical tools to find globally optimal solutions, although there is more current research to solve larger scale problems.

Non-Linear Programming (NLP) is a third setting where f can be almost any function and X is represented by the intersection of nonlinear inequalities. When all the functions are convex, NLP is called convex programming, in which interior point methods are also a powerful tool to find globally optimal solutions. The general non-convex setting is much more complex and even small-scale instances might be very hard to solve. Introductions to NLP can be found at [18, 19].

In all the above settings, we could constrain some of the variables to be integer (or binary). Whenever this happens in the LP case, we call it Mixed Integer LP (MILP) or also Mixed Integer Programming (MIP) [20]. This class of problem can also be very challenging since there are no theoretically efficient methods. However, many problems of interest can be solved in a reasonable time which makes MIP extremely useful for practitioners. A great advance has been seen in the last two decades with improvements in the order of 1000 combining hardware and software developments [21].

In the next two sections, we will highlight two important extensions of Mathematical Programming and then we will discuss software for optimization.

1.1.1 Stochastic Optimization

Stochastic Programming is an extension of mathematical programming to consider random variables besides the previously presented standard structure with decision variables, objective function and constraints. Introductory expositions can be found in [22, 23]. Stochastic Programming is typically considered a sub-field of optimization under uncertainty.

One of the most common frameworks for stochastic programming is the two-stage stochastic program, which, in the LP case, can be formulated as follows:

$$\min_x c^\top x + \mathbb{E}_\xi[Q(x,\xi)] \tag{1-5}$$

$$s.t.$$

$$Ax \leq b \tag{1-6}$$

with

$$Q(x,\xi) = \min_y q(\xi)^\top y \tag{1-7}$$

$$s.t.$$

$$W(\xi)y \leq h(\xi) - T(\xi)x \tag{1-8}$$

Where (1-5)–(1-6) represents the first stage problem, ξ is a random variable, (1-7)–(1-8) is the recourse problem. We highlight that all data from the recourse problem might depend on ξ.

Another generalization on top of two-stage stochastic programs is the multi-stage stochastic optimization (MSO) framework. In MSO first-stage decisions, random variable realizations and recourse actions are chained so that the recourse of a stage t is the first stage decision of stage $t+1$. A mathematical formulation, with the notation from [24], goes as follows:

$$\min_{(x_1,y_1)\in F_1}\left\{ f_1(x_1,y_1) + \mathbb{E}_{\tilde{\xi}_{[1,T]}|\xi_{[1,1]}}\left[\min_{(x_2,y_2)\in F_2(x_1,\xi_2)}\left\{ f_2(x_2,y_2,\xi_2) + ... \right. \right. \right. \tag{1-9}$$

$$\left. \left. \left. + \mathbb{E}_{\tilde{\xi}_{[T,T]}|\xi_{[1,T-1]}}\left[\min_{(x_T,y_T)\in F_T(x_{T-1},\xi_T)}\left\{ f_T(x_T,y_T,\xi_T) \right\} \right] ... \right\} \right] \right\} \tag{1-10}$$

where $F_t(x_{t-1},\xi_t)$ represents the constraints at stage t given the previous stage solution x_{t-1} and the random variable realization ξ_t. $\mathbb{E}_{\tilde{\xi}_{[t,T]}|\xi_{[1,t-1]}}$ is the conditional expectation with respect to the distributions of $\tilde{\xi}_{[t,T]}$ given the previous realizations $\xi_{[1,t-1]}$. More information can be found in [25, 26].

We highlight that there are other frameworks for decision under uncer-

tainty, such as Robust Optimization [27] and Chance Constrained Optimization [28], which will not be explored in this thesis.

1.1.2 Bilevel Optimization

Bilevel Optimization is another extension of Mathematical Programming in which one (or more) constraint of the optimization problem is another optimization problem.

This framework is extremely flexible and is equivalent to Stackelberg Games, in which two agents interact in chronological order. First, the leader, represented by the outer (or upper) optimization problem, makes a decision, and then the follower, represented by the inner (or lower) optimization problem, reacts. The leader will then choose the best possible action based on the follower's reactions. Introductions to bilevel optimization can be found in classic textbooks [29, 30] and in recent surveys [31, 32].

A mathematical formulation of the linear case is given by:

$$\min_{x,y} c^\top x + d^\top y \tag{1-11}$$

$$s.t.$$

$$Ax + By \leq b \tag{1-12}$$

$$y \in \arg\min_{y} e^\top x + f^\top y \tag{1-13}$$

$$s.t.$$

$$Gx + Hy \leq h \tag{1-14}$$

where (1-11)–(1-12) is the upper-level problem and (1-13)–(1-14) is the lower-level problem . The variable x is the upper-level decision variable and y is also a decision variable in the upper problem, which is constrained to be a solution to the lower-level problem. Note that x is treated as a parameter in the lower level, while y is the only decision variable in the lower-level problem.

A more general version of the above is:

$$\min_{x,y,\pi} c^\top x + d^\top y + g^\top \pi \tag{1-15}$$

$$s.t.$$

$$Ax + By + C\pi \leq b \tag{1-16}$$

$$y, \pi \in \arg\min_{y} e^\top x + f^\top y \tag{1-17}$$

$$s.t.$$

$$Gx + Hy \leq h \quad : \pi \tag{1-18}$$

where the dual variable, π, of the lower-level problem can be used in the upper-level model.

Of course, it is also possible to combine bilevel and stochastic optimization as reviewed in [33] as we are going to do in the following chapters of this thesis.

1.1.3 Software for Optimization

The field of optimization includes theoretical works, but applied methods are of utmost importance. In applications, practitioners must declare their abstract mathematical models in a concrete form to computers and then apply numerical algorithms to solve these problems. Therefore, software for optimization comes in two main flavors, frequently interlaced: modeling frameworks and solvers.

Modeling frameworks are designed to bridge the mathematical models to concrete data structures that can be communicated to solvers. Modern frameworks frequently fall in the scope of Algebraic Modeling Languages (AML) that attempt to keep the computer code similar to handwritten (or LaTeX) models. These tools are not essential in the strict sense since users can communicate raw data to solvers. For instance, in the case of LP, matrices can be constructed and used to represent constraint data. However, AMLs can make the code simpler and consequently more robust and easy to extend. Well-know AMLs include AMPL[34], GAMS[35], Pyomo[36] and JuMP[37, 38, 39]. While the first two are almost computer languages on their own, the latter two packages are built on top of the general programming languages Python[40] and Julia[41].

JuMP was a fundamental cornerstone in this thesis since absolutely all models described in the following chapters were coded in JuMP. In fact, the work on Chapter 2 is even more tied to JuMP since it is a JuMP extension to

handle Bilevel Optimization.

The second fundamental piece of software for optimization, solvers, implements state-of-the-art or innovative algorithms to solve various classes of problems. Results in this thesis would not be possible without these solvers. Although we refer to many of them in the benchmarks of Chapter 2, we strongly used state-of-the-art MIP solvers Gurobi[42] and Xpress[43] in Chapters 3 and 4 respectively.

1.2 Power Systems

Power systems is a very large research area that includes many different subfields. For instance, reliability and stability analysis, forecasting, transmission and distribution networks, expansion planning, operation, economics and much more. All of these subareas are related and frequently it is hard to talk about one without mentioning another. Optimization is used in all the above-cited subareas, sometimes as the key tool and sometimes as a side tool. We will focus on operation planning, economics and their intersection with optimization frameworks.

1.2.1 Operation Planning

One of the central themes in operation planning of power systems is the economic dispatch problem [44, 45]. In this problem, we consider a load D that must be met by the generation g_i of a set of power plants, I, with capacity G_i and variable unit cost C_i. In mathematical terms:

$$\min_{g_i} \sum_{i \in I} C_i g_i \tag{1-19}$$

$$s.t.$$

$$\sum_{i \in I} g_i = D \tag{1-20}$$

$$0 \leq g_i \leq G_i, \quad \forall i \in I \tag{1-21}$$

This is one of the simplest optimization models in operation planning, and it is an LP.

Considering additional features in this problem will require more complex models, sometimes still LPs but other times MIPs, CPs and NLPs. Moreover, other models will require advanced frameworks such as Stochastic and Bilevel Optimization (or even both).

For instance, a simple model that considers fixed operation costs F will require binary variables x that will represent whether the generator is active, 1, or inactive, 0:

$$\min_{g_i, x_i} \sum_{i \in I} C_i g_i + \sum_{i \in I} F_i x_i \tag{1-22}$$

$$s.t.$$

$$\sum_{i \in I} g_i = D \tag{1-23}$$

$$0 \leq g_i \leq G_i x_i, \quad \forall i \in I \tag{1-24}$$

$$x_i \in \{0, 1\}, \quad \forall i \in I \tag{1-25}$$

which is a MIP that can be extended to consider additional constraints such as multiple time steps, ramp, uptime, and downtime leading to a problem known as Unit Commitment (UC) [46, 47].

Note that the loads considered in the above problems are forecasts since they are not perfectly known in advance. As forecasts, they are generated by a statistical or a machine learning method that is obtained by solving specialized optimization problems [48].

Other features can be added on top of the above-described problems, such as the physical rules of power flows that can either be linearized or considered in more detail as non-linear constraints, which would make the problem an NLP [49]. A third alternative is to consider conic convex approximations of the non-linear power flow constraints so that global optimality can be proved efficiently. This would turn the problem into a CP [50, 51].

If we consider uncertainties in the economic dispatch problem, for instance, by replacing point forecasts with probabilistic forecasts, we obtain stochastic programs which can be two-stage or even multistage if multiple time steps are considered [52]. However, it might be hard to solve the stochastic program or even to formulate it if very little is known about the random variables. A common strategy is to consider reserves, which are generation slots allocated to generators in an economic dispatch but are not deployed initially. Instead, those are only deployed in real time to adapt the system operation to the realized uncertainties [53, 54].

Another reason to consider MSO is explicitly modeling hydro plants and reservoirs in the economic dispatch [55, 52]. Reservoirs act like large batteries, therefore, time coupling is strong and multiple stages must be considered. Moreover, incoming water is uncertain and, hence, treated as a random variable.

On the other hand, the simple economic dispatch might be made complex

by considering the strategic behavior of different players [56, 57]. For instance, in a setting where the generators can offer their costs and capacities at will, a generator might choose to optimally reduce the available capacity or increase it. This leads to the following bilevel problem:

$$\min_{q,g,\pi} \pi g_0 \tag{1-26}$$

$$s.t.$$

$$0 \leq q \leq G_0 \tag{1-27}$$

$$g, \pi \in \arg\min_{g,\pi} C_0 g_0 + \sum_{i \in I} C_i g_i \tag{1-28}$$

$$s.t.$$

$$g_0 + \sum_{i \in I} g_i = D \quad : \pi \tag{1-29}$$

$$0 \leq g_0 \leq q \tag{1-30}$$

$$0 \leq g_i \leq G_i, \quad \forall i \in I \tag{1-31}$$

where a strategic player indexed by 0 optimizes its quantity offer q trying to optimize its spot revenue. π is the spot price, the dual variable of the load balance constraint (1-29) in the economic dispatch of the inner (follower) subproblem (1-28)–(1-31). We follow the most common approach in the power systems literature, where games are in terms of prices [56] and also highlight that the Stackelberg game where a player chooses quantities first and the other follows is different com Cournot games, where players play simultaneously [58].

Many other power systems models have used bilevel optimization for various tasks. Another special case of bilevel problems in power systems is their relation with forecasting. In particular, hyperparameter tuning and other bilevel models can be used to train statistical and machine learning models with non-standard cost functions such as [59] and [5]. This is particularly important to this thesis as it is deeply related to Chapter 3.

1.3 Papers not in this thesis

Finally, we conclude the introduction with a list of other works developed during the Ph.D. program that are not direct parts of this thesis. Nevertheless, many works led to tools and software developments that were used in this thesis.

[39] - Lubin, M., Dowson, O., Garcia, J. D., Huchette, J., Legat, B., and Vielma, J. P. (2022). JuMP 1.0. arXiv preprint arXiv:2206.03866.

[60] - Sharma, A., Besançon, M., Garcia, J. D., and Legat, B. (2022). Flexible Differentiable Optimization via Model Transformations. arXiv preprint arXiv:2206.06135.

[61] - Souto, M., Garcia, J.D. and Veiga, Á., 2022. Exploiting low-rank structure in semidefinite programming by approximate operator splitting. Optimization, 71(1), pp.117-144.

[62] - Soares, A., Street, A., Andrade, T. and Garcia, J.D., 2022. An Integrated Progressive Hedging and Benders Decomposition with Multiple Master Method to Solve the Brazilian Generation Expansion Problem. IEEE Transactions on Power Systems.

[63] - Legat, B., Dowson, O., Garcia, J.D. and Lubin, M., 2021. MathOptInterface: a data structure for mathematical optimization problems. INFORMS Journal on Computing.

[64] - Rosemberg, A.W., Street, A., Garcia, J.D., Valladão, D.M., Silva, T. and Dowson, O., 2021. Assessing the Cost of Network Simplifications in Long-Term Hydrothermal Dispatch Planning Models. IEEE Transactions on Sustainable Energy, 13(1), pp.196-206.

[65] - da Costa, L.C., Thomé, F.S., Garcia, J.D. and Pereira, M.V., 2020. Reliability-constrained power system expansion planning: A stochastic risk-averse optimization approach. IEEE Transactions on Power Systems, 36(1), pp.97-106.

[66] - Rosemberg, A.W., Street, A., Garcia, J.D., Silva, T., Valladão, D.M. and Dowson, O., 2020, July. HydroPowerModels.jl: A Julia/JuMP package for hydrothermal economic dispatch optimization. In Proceedings of the JuliaCon Conferences (Vol. 1, No. 1, p. 35).

2
`BilevelJuMP.jl`: Modeling and Solving Bilevel Optimization in Julia

In this chapter, we present BilevelJuMP, a new Julia package to support bilevel optimization within the JuMP framework. The package is a Julia library that enables the user to describe both upper and lower-level optimization problems using the JuMP algebraic syntax. Due to the generality and flexibility our library inherits from JuMP's syntax, our package allows users to model bilevel optimization problems with conic constraints in the lower level and all JuMP-supported constraints in the upper level (Conic, Quadratic, Non-Linear, Integer, etc.). Moreover, the user-defined problem can be subsequently solved by various techniques relying on mathematical programs with equilibrium constraints (MPEC) reformulations. Manipulations on the original problem data are possible due to MathOptInterface.jl's structures and Dualization.jl features. Hence, the proposed package allows quickly modeling, deployment, and thereby experimenting with bilevel models based on off-the-shelf mixed integer linear programming and nonlinear solvers.

2.1 Introduction

Bilevel optimization has been a widely used modeling tool in mathematical programming, operations research, and economics since its first introduction by [67] in game theory. The broad range of applications includes hyperparameter optimization in machine learning [5], toll setting in transportation networks [68], multiple problems in energy and power systems [69], defense applications [70], facility location [71] only to list a few areas.

Complete introductions to bilevel optimization can be found in books covering theoretical background and analysis, taxonomy, solutions algorithms for special classes, and selected applications [29, 30, 72]. In addition to books, many reviews on the subject have been published in the last decades [73, 74, 75, 31]. A very long list of publications related to bilevel optimization can be found in [32].

It is well known that general bilevel optimization problems fall in the NP-hard class [6]. Hence, there is no hope of finding efficient algorithms for generic

problems. On the other hand, modeling bilevel problems is a theoretically more straightforward task, albeit, in practice, the modeling step can be the difference between finding a tractable model for which there is a reasonable solution approach in realistic cases or not. Because bilevel models are very complex and constitute a broad class of mathematical programming problems, many modeling languages lack the proper functionality to handle these problems.

In the following subsections, we present the main literature regarding available techniques and software to place and justify the contribution of our work properly. Notwithstanding, it is out of the scope of this section to provide a comprehensive review on the subject, for which we refer to previously reported books and reviews.

2.1.1 Solving bilevel optimization

Many strategies have been proposed to solve Bilevel Optimization problems. Some of the most widely known techniques are based on classical algorithms, such as the simplex method, the branch and bound method, and the interior-point methods. Due to the inherent combinatorial nature of many Bilevel optimization problems, some of the first developed techniques are intrinsically combinatorial. Among a large set of enumeration algorithms, which can be seen as modifications of the simplex method for linear programming, we highlight the K^{th}-best algorithm [76].

The fundamental technique that will be explored in this work is converting the bilevel problem into a single-level problem by adding the lower-level KKT conditions to the upper-level problem. The resulting optimization problem is known as Mathematical Programming with Equilibrium Constraints, MPEC [77]. This group of techniques has also been labeled enumeration-based because handling the complementarity constraints is a combinatorial problem. Thus, a classic solution is a specially tailored Branch and Bound [78, 79]. Instead of writing a branch and bound method from scratch, one could reformulate the single-level problem into an amenable form for an off-the-shelf algorithm like mixed integer programming (MIP) solvers. This method was first presented relying upon big-M formulations in [80]. More recently, formulations based on special ordered set of type 1, SOS1, were developed in [81].

Interestingly, non-linear programming (NLP), with additional regularization terms, can also be used to solve MPECs [82, 83, 84]. In this case, it would not make as much sense to call the MPEC reformulations an enumeration-based method. On the other hand, the latter might be called local search methods, as this strategy leads to local solutions, in contrast with the global

ones provided by MIP-based methods. Some solvers were specially tailored to tackle MPEC's: KNITRO [85], filterMPEC [86, 82], NLPEC [87]. A combination of NLP and MIP-based methods was proposed by [88].

Other strategies to solve bilevel problems include: bundle type algorithms [89], semi-definite relaxations [90], penalty function based methods [91, 92, 93], Benders decomposition [94]. Cutting-plane approaches [95, 96, 97] have received attention recently because they can handle lower-level problems with integer variables — solvers were developed by [98, 99]. Descent methods were proposed by [100] and [101] to obtain quick local solutions. Heuristic methods were also developed to obtain practical solutions, for instance, the bi-objective-based method of [102]. Finally, we refer to [103] for evolutionary approaches.

2.1.2
Modeling bilevel optimization

Algebraic modeling languages (AML) play a central role in unlocking the huge potential of optimization models through a friendly environment that integrates solvers and models in a very practical manner where problems of all disciplines can be efficiently modeled and solved. Bilevel Optimization interfaces, which mostly automate reformulations and pass to specific solver types, have also been proposed.

GAMS has an interface described in its Extended Mathematical Programming, [104]. Variables and constraints are created as usual, and then they are annotated to specify the level they belong to in an external file. The annotated problem is reformulated by GAMS EMP routines using the KKT reformulations [105]. Finally, the problem is optimized by the available MPEC solvers, namely, the above-cited KNITRO and NLPEC. The follower subproblems can be linearly constrained with quadratic objectives (QP) or Variational Inequalities (VI). The upper-level problem is constrained by the selected solver capability.

The YALMIP MATLAB package for optimization modeling [106] also provides a Bilevel Optimization interface [107]. The variables and constraints are defined by the standard methods, and then they are passed as lists to a "solvebilevel" function. The lower level problem can be a QP, and the upper level can be anything supported by the YALMIP interface. The available solution methods are based on the MPEC reformulation where the handling of complementarity constraints is forwarded to external MIP, NLP or MPEC solvers or to an internal branch and bound that allows different solvers for upper and lower-level problems.

It is also possible to model Bilevel optimization in the Pyomo Adversarial

Optimization, PAO [108], a Python package extending Pyomo [36]. A (lower) submodel is created as an object attached to the main (upper) model, then variables and constraints are created directly in their owner models, the former can be shared in objectives and constraints. As in the previous two AMLs, the problem is automatically reformulated, and in Pyomo's case can be passed either to an NLP solver [84] or a MIP solver [80]. Additionally, PAO has an interface to the MibS solver [98] and implements a Column Constraint Generation, CCG [109].

The Julia [41] package BilevelOptimization.jl [110] provides a very simple interface for Bilevel modeling in JuMP [37, 111]. Still, the interface remains very basic. For instance, although the upper level can represent arbitrary JuMP problems (as long as the selected solver supports them), the lower level is constrained to QP. The package supports two MIP-based methods: big-M [80] and the SOS1-based reformulation. The critical issue is that the lower level is not represented by a JuMP-based syntax, preventing Julia users from fully enjoying the modeling power provided by JuMP. Instead, lower-level problems must be described by matrices, which can be easily manipulated to write KKT reformulations. This is one of the salient features and the primary goal of our proposed package, BilevelJuMP, namely, to allow representing the lower level problem within the JuMP syntax in a single and integrated new bilevel JuMP model.

It is important to highlight that, just like [112] and [34], JuMP also has native support for complementarity constraints, thereby being capable of handling MPEC models. However, none of them can model Bilevel Optimization models directly.

2.1.3 Objective and Contributions

The main objective of this work is to provide a complete open-source interface for Bilevel Optimization fully integrated into the JuMP modeling language named BilevelJuMP.jl and available online at:

`https://github.com/joaquimg/BilevelJuMP.jl`

It is also readily available at the Julia Package manager, at the time of this thesis, in version 0.5.1. Julia users can run `add BilevelJuMP` and have full access to all features of the library.

Regarding similar works, [110] provided great motivation and a nice first step to tackle bilevel models in Julia. However, it is incomplete as a modeling framework due to the strong limitations associated with the modeling

of second-level problems, as explained before. The consideration of a generic JuMP-based second-level model within a JuMP-integrated interface significantly increases functionality parity with other AML's. Notwithstanding, and more importantly, it paves the way for new developments and computational applications based on bilevel optimization. Because the proposed package provides a simple and integrated interface, fully embedded into the JuMP language, new and expert users can easily prototype and test bilevel models enjoying all open-source and commercial linear, nonlinear, and MILP solvers integrated into JuMP, depending on the model characteristics. Hence, the newly proposed BilevelJuMP.jl can be used in teaching environments to introduce practical aspects of bilevel modeling as well as in practical applications inheriting many of the functionalities and advances directed to JuMP, one of the main packages of Julia.

Similar to other interfaces, BilevelJuMP.jl is also capable of reformulating bilevel problems and exporting the model to existing external solvers. In particular, many reformulations were implemented to allow practitioners to test each method for their particular problems. Some experiments are presented in this work to provide a first glance at the differences between existing methods.

Researchers of the Bilevel Optimization community are similarly benefited. The functionality exposed by BilevelJuMP.jl can be used as a benchmark in terms of performance and solution quality for new algorithms. These new algorithms might even be implemented in Julia with the data structures already defined in the package. Therefore, we also provide a software contribution with pieces of code that can be directly used in future implementations.

BilevelJuMP.jl was designed to be extensible, and the various implemented methods vouch for it. As we shall discuss, the key ingredient to developing all the solution methods is to rely on MathOptInterface.jl's API.

Advanced functionality is also part of the contributions. BilevelJuMP.jl can represent a Conic Program, CP, in the lower-level problem and can deal with upper-level constraints, including dual variables of lower-level problems (see Section 2.8.1). Finally, the composability inherent in many Julia packages allows performing reformulations that enable the user to solve even more complex problem classes.

2.2
Conic Bilevel standard form

We will only consider optimistic bilevel problems [29], in short, the solution of the lower level will be the one that optimizes the upper level in case

of degeneracy. We start by describing the main notation that will be used in the remainder of the work. z and x are vectors of decision variables, respectively, from the upper and lower-level problems. While x is n^L dimensional, z has n^U entries. $[x, z]$ is a $(n^L + n^U)$–vector with the elements of x and z stacked. Q^j, a_i^j, d_i^j, b_i^j, A_i^j, D_i^j, for $j \in \{U, L\}$ are matrices (upper case) and vectors (lower case) of constants. $\mathcal{C}_i^j$, for $j \in \{U, L\}$, are sets of arbitrary finite dimensions, which most commonly will be convex cones. m^U and m^L are the numbers of vector constraints in the upper and lower problems. As in traditional bilevel programming, z is decided in the upper level and passed to the lower level as a parameter and x might be seen as an upper-level variable constrained to be an optimal solution of the lower level. Hence, the optimistic bilevel problem follows:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^{n^L}, z \in \mathbb{R}^{n^U}} \quad & \frac{1}{2}[x, z]^\top Q^U [x, z] + {a_0^U}^\top x + {d_0^U}^\top z + b_0^U \\
\text{s. t.} \quad & A_i^U x + D_i^U z + b_i^U \in \mathcal{C}_i^U, i = 1 \dots m^U \\
& x(z) \in \underset{x \in \mathbb{R}^{n^L}}{\arg\min} \quad \frac{1}{2}[x, z]^\top Q^L [x, z] + {a_0^L}^\top x + {d_0^L}^\top z + b_0^L \\
& \qquad \text{s. t.} \quad A_i^L x + D_i^L z + b_i^L \in \mathcal{C}_i^L, i = 1 \dots m^L
\end{aligned}
$$

As detailed by [113], describing constraints as function-set pairs is very flexible. For simplicity, we limited ourselves to affine functions contained in sets in the constraints of the above model. If all sets are all convex cones, we have a standard conic form for bilevel programs.

Keeping the lower level problem as a convex conic program is especially useful for writing KKT conditions when converting the problem into MPEC form. Although lower-level integer variables could be tackled by specialized solvers [97], the same goes for non-linear constraints like the ones in [103]. The upper-level problem can be more complex, including non-linear constraints and integer variables, because they are not affected in MPEC reformulations. For simplicity of presentation, we do not include lower level duals in the upper level problem, the generalization is simple and can be done as in [114].

2.2.1
KKT Reformulation of bilevel programs

Given a conic bilevel program in the standard form that we described in the previous section, we can formulate an equivalent MPEC applying the KKT reformulation to convert the lower level optimization problem into a set of linear and non-linear equations.

Let us focus on the following parametric convex quadratic conic problem

that is equivalent to the lower level problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & +\frac{1}{2}x^\top Q_1 x + x^\top Q_2 z + \frac{1}{2}z^\top Q_3 z + {a_0}^\top x + b_0 + d_0^\top z \\ \text{s.t.} \quad & A_i x + b_i + D_i z \in \mathcal{C}_i \quad i = 1 \dots m \end{aligned} \tag{2-1}$$

Note that the L superscripts were dropped for simplicity, and we split the Q matrix in Q_1, Q_2, and Q_3. Because z are parameters, only Q_1 is required to be a positive semi-definite matrix. In the following, we will denote the dual cone of $\mathcal{C}_i$ as $\mathcal{C}_i^*$.

Following [15], we can write the KKT conditions as:

– Primal Feasibility:

$$A_i x + b_i + D_i z \in \mathcal{C}_i, \quad i = 1 \dots m \tag{2-2}$$

– Dual Feasibility:

$$y_i \in \mathcal{C}_i^*, \quad i = 1 \dots m \tag{2-3}$$

– Stationarity:

$$Q_1 x + Q_2 z + a_0 - \sum_{i=1}^{m} A_i^\top y_i = 0 \tag{2-4}$$

– Complementary slackness:

$$y_i^\top (A_i x + b_i + D_i z) = 0, \quad i = 1 \dots m \tag{2-5}$$

Combining all the above we arrive at the MPEC form of the bilevel conic program:

$$\begin{aligned} \min_{x, z, y_1, \dots, y_{m^L}} \quad & \frac{1}{2}[x, z]^\top Q^U [x, z] + {a_0^U}^\top x + {d_0^U}^\top z + b_0^U \\ \text{s. t.} \quad & A_i^U x + D_i^U z + b_i^U \in \mathcal{C}_i^U, \quad i = 1 \dots m^U \\ & A_i^L x + b_i^L + D_i^L z \in \mathcal{C}_i^L, \quad i = 1 \dots m^L \\ & y_i \in \mathcal{C}_i^{L^*}, \quad i = 1 \dots m^L \\ & Q_1^L x + Q_2^L z + a_0^L - \sum_{i=1}^{m} {A_i^L}^\top y_i = 0 \\ & y_i^\top (A_i^L x + b_i^L + D_i^L z) = 0, \quad i = 1 \dots m^L \end{aligned}$$

Such a form is particularly useful for solving bilevel optimization problems, as described in the previous sections. The main challenge being (2-5) constraints which are non-linear and non-convex. Dealing with the latter requires special solvers, tailor-made algorithms, or extra reformulation steps to reach the standard form of some NLP or MIP solvers.

2.3 BilevelJuMP.jl

BilevelJuMP.jl is an extension of the JuMP modeling language [111, 37] for optimization problems in the Julia language [41]. Other packages successfully extending JuMP and MOI include SDDP.jl [115], SumOfSquares.jl [116], InfiniteOpt.jl [117]. BilevelJuMP.jl has two main functionalities: modeling and solving bilevel optimization problems.

This open-source package heavily relies on MathOptInterface.jl, also referred to as MOI, [113], another Julia package that was written to be the new backend of JuMP which led to a complete rewrite of the latter. MathOptInterface.jl is an intermediary layer between JuMP's user-friendly AML interface and the diverse and typically matrix-oriented format of solvers. In BilevelJuMP.jl, MOI is used to store problem data from an extended JuMP interface and reformulate bilevel optimization problems into the MPEC form and then into a solver-compatible formulation of MPEC.

2.3.1 A Modeling Interface for Bilevel Optimization

The basic modeling interface of BilevelJuMP.jl relies on JuMP's extensible methods and macros to write and combine two optimization problems. Not surprisingly, other methods had to be created to accommodate the needs of bilevel optimization interfaces.

The main data structure in this software is the `BilevelModel`, which is a subtype of JuMP's `AbstractModel`. `BilevelModel` holds two other JuMP Models to represent the upper and lower optimization problems. Also, additional information is held to link the two problems and store additional JuMP data and attributes used in reformulations.

Just like a regular JuMP `Model`, the `BilevelModel` will need a solver constructor to solve an optimization problem. On the other hand, the BilevelModel will require a solution `mode` which will select the technique used in the solution process. The final pieces of the basic interface are the `Lower` and `Upper` methods that direct JuMP macros to the proper bilevel optimization levels.

We will exemplify the basic interface by modeling the following simple bilevel optimization problem from [29], Chapter 3.2, Page 25:

$$
\begin{aligned}
\min_{y \in \mathbb{R}} \quad & 3x + y \\
\text{s.t.} \quad & x \leq 5 \\
& y \leq 8 \\
& y \geq 0 \\
& x(y) \in \underset{x \in \mathbb{R}}{\arg\min} \quad -x \\
& \qquad\qquad \text{s.t.} \quad\quad x + y \leq 8 \\
& \qquad\qquad\qquad\quad\;\; 4x + y \geq 8 \\
& \qquad\qquad\qquad\quad\;\; 2x + y \leq 13 \\
& \qquad\qquad\qquad\quad\;\; 2x - 7y \leq 0
\end{aligned}
$$

The code to model, solve and query solutions are presented in Figure 2.1.


```
using JuMP, BilevelJuMP, SCIP
model = BilevelModel(SCIP.Optimizer, mode = BilevelJuMP.SOS1Mode())
@variable(Upper(model), y)
@variable(Lower(model), x)
@objective(Upper(model), Min, 3x + y)
@constraints(Upper(model), begin
    x <= 5
    y <= 8
    y >= 0
end)
@objective(Lower(model), Min, -x)
@constraints(Lower(model), begin
     x +  y <= 8
    4x +  y >= 8
    2x +  y <= 13
    2x - 7y <= 0
end)
optimize!(model)
objective_value(model) # = 3 * (3.5 * 8/15) + 8/15
value(x) # = 3.5 * 8/15
value(y) # = 8/15
```

Figure 2.1: Code to solve the example of Figure 2.1

We can follow the general workflow: include packages; initialize the model jointly with a solver, `SCIP` [118] in this case, and the solution mode `SOS1Mode` (modes will be discussed in the following sections); add variables to the proper levels so that they can be used by all constraints and objectives; add constraints and objectives to the proper levels (which can be done in any order); optimize the model; and query solutions.

2.3.2
Solving Bilevel Optimization with Reformulations

MathOptInterface defines a unique and well-posed interface that makes it possible to perform reformulations in problem instances to convert from one format into others. We start from an arbitrary user formulation in JuMP stored as an MOI model, then we can rewrite this model in an alternate form, which will lead to an MOI model, and consequently, we can pass the model to a solver wrapper that implements the MOI API. The solver optimizes the model and returns the solutions, which flow back to JuMP by applying the necessary mappings and transformations. The simplest and most used of these transformations are bridges [113], which are applied to individual variables, constraints, and objectives. The bridge system automatically converts a problem into the specific form expected by the solver.

However, some transformations require looking at the model as a whole and not only at its pieces (variables, objectives, and constraints). The first implementation of a whole model transformation was Dualization.jl [119]. Dualization.jl's main function receives an MOI model and writes its dual in a second MOI model. Clearly, to perform such modification, the complete model must be known in advance. This feature is especially useful because some solvers only accept specific forms of mathematical programs. Hence, we can convert between primal and dual forms and solve the converted form without relying on the bridge system, which might increase the problem size to reach the form required by the solver.

Dualization.jl also plays a key role in BilevelJuMP.jl's reformulations. Given a primal model like (2-1), where x are variables and z are parameters, we can obtain the dual form following [120]:

$$\begin{aligned}
\max_{y_1,\dots,y_m,w} \quad & -\frac{1}{2}w^\top Q_1 w + \frac{1}{2}z^\top Q_3 z - \sum_{i=1}^{m}(b_i + D_i z)^\top y_i + d_0^\top z + b_0 \\
\text{s.t.} \quad & a_0 + Q_2 z - \sum_{i=1}^{m} A_i^\top y_i + Q_1 w = 0 \qquad (2\text{-}6) \\
& y_i \in \mathcal{C}_i^*,\ i = 1 \dots m \qquad (2\text{-}7)
\end{aligned}$$

We observe that the Dual feasibility constraints (2-6)–(2-7) are structurally very similar to the Dual Feasibility (2-3) and Stationarity (2-4) constraint sets from the KKT conditions. The only difference is that, in (2-6), Q_1 multiplies an additional variable w, and in (2-4) Q_1 multiplies a variable x.

BilevelJuMP.jl performs a more complicated model transformation. The two JuMP models that described each level of the bilevel program must be

combined in a particular way to create the corresponding MPEC. Even though each variable belongs to one level, they are created in both but tagged with additional data to mark their level and their corresponding variable in the other level. However, from the user perspective, they should be accessed from the level they were created.

The first part of the transformation is to copy the upper level into a new model to append the other pieces of the MPEC. The second step is to add the KKT conditions of the second level. The Primal Feasibility constraints of the lower level are added as new constraints to the model (using the variable map between the two original JuMP models). Then the lower level model is dualized, considering upper-level variables as constants, and its constraints are passed to the new model to represent Stationarity and Dual Feasibility constraints. The additional variables, w, created in the dual problem are mapped into the upper variables x. At this point, we only need to add complementarity constraints.

2.4 KKT Formulations

In this section, we describe the reformulation of the conic MPEC to obtain mathematical programs that can be passed to existing solvers.

2.4.1 Complementary slackness reformulations

Starting from a convex problem, all the KKT conditions lead to convex constraints except the complementary slackness constraints. The main challenge in KKT reformulations is dealing with such non-linearity. Now we present some possible formulations which were already implemented and tested in BilevelJuMP.

We will assume that y_i and $A_i x + b_i + D_i z$ are scalars since almost all formulations rely on this assumption. The following formulations are restricted to: $\mathcal{C}_i \in \{\mathbb{R}_+, \mathbb{R}_-, \{0\}\}$. Without loss of generality, we will assume $\mathcal{C}_i = \mathbb{R}_+$.

2.4.1.1 Special Ordered Sets of type 1

One SOS1 reformulation was presented in [121] and in [81]. In BilevelJuMP.jl this formulation consists in replacing the complementarity con-

straints with the following:

$$f_i = A_i x + b_i + D_i z \tag{2-8}$$
$$[y_i; f_i] \in SOS1 \tag{2-9}$$

Considering this is the classic SOS1 set from [122], the SOS1 constraint implies that a solution is feasible only if at most one of the variables in the SOS1 set is different from zero. It is equivalent to the original formula because one of the two scalars will have to be zero to have the product equal to zero.

Many solvers can handle this kind of constraint, e.g., Cbc, CPLEX, Gurobi, SCIP, Xpress, which makes this formulation particularly useful for practitioners.

2.4.1.2 Indicator constraints

Indicator constraints [123] and SOS1 are deeply related. A typical indicator constraint is defined by:

$$x = 0 \implies Ay \leq b \tag{2-10}$$

This means that the constraint $Ay \leq b$ is only considered if $x = 0$, where x is a binary variable. Analogously, another indicator constraint could depend on $x = 1$. Hence, one possible formulation for the complementarity slackness with indicator constraints is:

$$f = 0 \implies A_i x + b_i + D_i z = 0 \tag{2-11}$$
$$f = 1 \implies y_i = 0 \tag{2-12}$$
$$f \in \{0, 1\} \tag{2-13}$$

Many solvers are also capable of handling this kind of constraint which also makes this formulation very useful. As a final note on this formulation, we note that a solver might not support Indicator Constraints for both $f = 0$ and $f = 1$, in this case, we simply need one additional variable g and the constraint: $f + g = 1$.

2.4.1.3 Fortuny-Amat and McCarl

This formulation is commonly known by the name of the authors of [80] and is frequently used in practice. In very few words, it is an application of the big-M method:

$$A_i x + b_i + D_i z \leq M_p f \tag{2-14}$$

$$y_i \leq M_d(1 - f) \tag{2-15}$$

$$f \in \{0, 1\} \tag{2-16}$$

In such formulation, M_p and M_d are large numbers. We have assumed that both $A_i x + b_i + D_i z$ and y_i are positive, thus, for each value of f one of the elements in the complementarity pair is forced to zero.

The main drawback of this method is that the values of M_p and M_d must be large enough so that the optimal solution of the problem is not excluded. One can usually develop bounds on primal variables because the variable might be bounded due to the problem definition. However, finding reasonable bounds for dual variables might be much harder on specific applications. The work by [124] shows that commonly used heuristics to select the big-Ms can fail. [125] go further and demonstrate that verifying big-Ms is at least as hard as solving the Bilevel Problem itself.

When good bounds are available, the Fourtuny-Amat and McCarl formulation is very efficient in practice [121]. Moreover, no extra constraints are required from solvers. Therefore, less complete MIP solvers like GLPK can be used to solve this kind of reformulation. On the other hand, the difficulty of computing bounds makes the SOS1 and Indicator formulations very useful for experimentation.

2.4.1.4 Products

This is not a reformulation because, in this case, the actual complementarity constraint in its product form is added to the optimization problem:

$$y_i^\top (A_i x + b_i + D_i z) = 0 \tag{2-17}$$

NLP solvers frequently use this form to seek local optimal solutions. Although no guarantees of global optimality are provided when using this method, it is useful the get initial solutions to be used as bounds or even for cases where MIP solving is not practical. An additional weakness of this method is that (2-17) does not satisfy constraint qualification [83, 84, 82] and is regularised

as:

$$y_i^\top (A_i x + b_i + D_i z) \leq t \tag{2-18}$$

where t is a small number.

In theory, one could reformulate all the products with binary expansion techniques such as the one in [126] and use MIP solvers jointly with NLP solvers to reach solutions close to globally optimal solutions. In practice, binary expansions also require bounds on the variables that are multiplied. This adds complications to the solution method because these cannot be added as regular constraints at the lower level; otherwise, they would be dualized, leading to more unbounded variables on both sides.

The binary expansion technique was implemented in QuadraticToBinary.jl [127], which can be used as an intermediary layer between BilevelJuMP.jl (or JuMP) and the selected solver—allowing any MIP solver with an MOI interface to solve approximations of quadratically constrained problems.

This formulation easily extends to vector sets. Hence, conic bilevel problems will require this formulation in BilevelJuMP.

2.4.1.5 Complements

Some solvers are able to handle explicit complement constraints like Knitro [85], filterMPEC [86, 82], NLPEC [87]. These solvers receive the constraints as special structures: pairs of variables or variable-expression pairs. Internally, the solver will employ their own reformulations.

$$y_i \perp A_i x + b_i + D_i z \tag{2-19}$$

2.4.1.6 Mixed mode

Usually, practitioners select one single formulation and apply it to all complementarity constraints in the problem, but this is not a technical requirement. Consequently, one could combine formulations and select which formulation will be used for each pair. For instance, if one has good bounds for a specific pair, then just use Fortuny-Amat and McCarl for that constraint, while the other constraints would be reformulated with SOS1, for instance.

Alternatively, even a conic bilevel with multiple linear constraints could be reformulated with SOS1 for all linear constraints and product mode (and binary expansions) for the conic constraints. We present an application of this method in Section 2.8.2.

2.4.2
Primal Dual Equality reformulation

This formulation takes advantage of the fact that, under strong duality, the complementarity constraints are equivalent to enforcing that the primal and dual objective values are the same for a solution that is both primal and dual feasible. Therefore, the complementarity constraints are replaced by:

$$\frac{1}{2}x^\top P_1 x + x^\top P_2 z + a_0^\top x = -\frac{1}{2}w^\top P_1 w - \sum_{i=1}^{m}(b_i + D_i z)^\top y_i \tag{2-20}$$

Where the identical terms were already eliminated. This is also a non-convex quadratic constraint, even if the problem is linear due to z and y products.

One exciting feature of this formulation is that all the quadratic terms are concentrated in a single constraint, and the number of variable products might be much smaller than the number of complementarity constraints. Consequently, binary expansions were shown to be helpful in replacing the quadratic terms and achieving approximate global optimal solutions in [128, 129].

2.4.3
Comparison of methods

We present a brief comparison between the solution methods. Table 2.1 presents the method name in BilevelJuMP, the section in which it is described, solver requirements and additional comments.

Method Name	Sec.	Solver requirement	Comments
`SOS1Mode`	2.4.1.1	MIP solver with SOS of type 1	No additional information. Only linear constraints.
`IndicatorMode`	2.4.1.2	MIP solver with Indicator Constraints	No additional information. Only linear constraints.
`FortunyAmatMcCarlMode`	2.4.1.3	MIP solver	Require non-trivial big-M. Only linear constraints.
`ProductMode`	2.4.1.4	Non-convex quadratic constraints	Works with conic constraints. Require regularization.
`ComplementMode`	2.4.1.5	Complementarity constraint	Few solvers supporting such constraints.
`MixedMode`	2.4.1.6	Requirements of selected methods	Pros and cons from selected methods.
`StrongDualityMode`	2.4.2	Non-convex quadratic constraints	Works with conic constraints.

Table 2.1: Reformulation methods

2.5
Example

In this section, we describe a slightly more interesting example of a bilevel program. The main goal is to start from a non-trivial problem, model it in

BilevelJuMP, and solve it with multiple methods to have a glimpse of the multitude of applications of the package.

As previously mentioned, hyperparameter tuning with bilevel optimization is a recent trend in the intersection of the Machine Learning and Optimization communities [5, 130, 131]. Although most hyperparameter tuning methods based on bilevel optimization are usually heuristic with special considerations to the problem in question, this is a good case to describe the functionality of the package due to the simplicity of the model and because small enough instances can be solved by standard methods implemented in this package.

We have selected hyperparameter tuning in support vector regressions (SVR). The example will follow the one from [132], though with some simplifications. Given a set of features J and two data sets O and I, with out-of-sample and in-sample data, represented by the points labeled by i, $(y_i, \{x_{ij}\}_{j\in J})$, we define the following model:

$$
\begin{aligned}
\min_{C\geq 0,\varepsilon\geq 0,\xi^U\geq 0,w} \quad & \sum_{i\in O} \xi_i^U \\
\text{s. t.} \quad & \xi_i^U \geq +y_i - \sum_j w_j x_{ij}, \quad i \in O \\
& \xi_i^U \geq -y_i + \sum_j w_j x_{ij}, \quad i \in O \\
& w(C,\varepsilon) \in \underset{\xi^L\geq 0,w}{\arg\min} \quad ||w||_2^2 + C\sum_{i\in I} \xi_i^L \\
& \qquad\qquad \text{s. t.} \quad \xi_i^L + \varepsilon \geq +y_i - \sum_{j\in J} w_j x_{ij}, \quad i \in I \\
& \qquad\qquad\qquad\quad \xi_i^L + \varepsilon \geq -y_i + \sum_{j\in J} w_j x_{ij}, \quad i \in I
\end{aligned}
$$

The lower model is the SVR problem formulation. Therefore it is responsible for obtaining the best possible support vectors w given the problem data and the hyperparameters C and ε. The hyperparameters are variables selected by the upper level so that the w optimized by the lower level has the smallest out-of-sample error. The variables ξ^U and ξ^L denote the absolute value loss in the upper and lower models, respectively. The upper level is a linear program, while the lower level is quadratic. In Figure 2.2 we present BilevelJuMP.jl code to model the hyperparameter tuning of SVR described above. Thanks to the JuMP syntax, the code greatly resembles the abstract model, simplifying the writing and documenting of the code.

That same code was used to perform a series of comparisons between solvers. We started by creating instances with a different number of features and observations (dataset size). We randomly created the matrix x with a

```
using JuMP, BilevelJuMP
# sample data
Features = 2
Samples = 10
J = 1:Features
I = 1:div(Samples, 2)
O = (div(Samples, 2)+1):Samples
x = 2 * (rand(Samples, Features) .- 0.5)
w_real = ones(Features)
y = x * w_real .+ 0.1 * 2 * (rand(Samples) .- 0.5)
# model building
model = BilevelModel()
@variable(Upper(model), C >= 0)
@variable(Upper(model), eps >= 0)
@variable(Upper(model), xi_U[i=O] >= 0)
@variable(Lower(model), w[j=J])
@variable(Lower(model), xi_L[i=I] >= 0)
@objective(Upper(model),
    Min, sum(xi_U[i] for i in O))
@constraints(Upper(model), begin
    [i in O], xi_U[i] >= + y[i] - sum(w[j]*x[i,j] for j in J)
    [i in O], xi_U[i] >= - y[i] + sum(w[j]*x[i,j] for j in J)
end)
@objective(Lower(model),
    Min, sum(w[j]^2 for j in J) + C * sum(xi_L[i] for i in I))
@constraints(Lower(model), begin
    [i in I], xi_L[i] + eps >= + y[i] - sum(w[j]*x[i,j] for j in J)
    [i in I], xi_L[i] + eps >= - y[i] + sum(w[j]*x[i,j] for j in J)
end)
```

Figure 2.2: Code to model SVR hyper-parameter tuning

uniform distribution in $[-1, +1]$, then we created the *real* w as a vector of ones with appropriate dimension. Next, we defined $y = xw + \epsilon$, where ϵ follows a uniform distribution in $[-0.1, +0.1]$. Half of the dataset was considered in-sample data, while the other half was considered out-of-sample data. It is not our intention to be fully realistic here, our goal is to provide a didactic example.

We created instances with 10, 100 and 1000 samples. For all these sample sizes we created samples with 1, 2 and 5 features. For the datasets with 100 samples, we also created datasets with 10, 20 and 50 features.

Finally, we optimized the bilevel problem for each data set with multiple reformulations and with multiple solvers. The only solver attribute we set was a time limit of 600 seconds (10 minutes) and left all other attributes as default, which might differ considerably from one solver to the other. Again, our primary goal is not a detailed and rigorous comparison of solvers but to show the software's functionality in a usage example, as practitioners and researchers might want to solve the same problem with multiple methods and select the one that best fits their needs.

We present the results in the following tables. We used Julia 1.6.2, CPLEX 22.1 [133], Gurobi 9.5 [42], HiGHS 1.2 [134], Ipopt 3.14 [135], Knitro 13.0 [85], SCIP 8.0 [118], Xpress 8.13 [43]. All the required code is in the

benchmarks folder of the git repository, including exact package versions (see the *manifest.toml* file).

All tables have a similar format. The first column describes the instance, the first number being the sample size and the second the number of features. Then we have three columns for each solver, the first, *Obj*, presents the upper-level objective value returned by the solver (typically the best incumbent solution), the second contains the *Gap* in percent (%), Ipopt and KNITRO will not have gaps as they are NLP solvers, if no gap was reported the entry will be blank (with a "−"), the third is *Time* in seconds, if the time reaches 600 the entry will be blank (with a "−").

Table 2.2 presents results for `SOS1Mode` and `IndicatorMode`. Table 2.3 presents results for `FortunyAmatMcCarlMode`, with big-Ms set to 100, `StrongDualityMode` and `ProductMode`, the latter two with binary expansions, so the resulting problem is a MIP, where the variable bounds were set to $+/-100$. Finally, Table 2.4 presents the solutions of both `ProductMode` and `StrongDualityMode` for Non-Linear Programming solvers and Gurobi with its *NonConvex* mode activated.

We can draw some conclusions from the tables. We note that `SOS1Mode` and `IndicatorMode` perform well in smaller instances, with a slight advantage for `SOS1Mode`. Interestingly, CPLEX's solution for 1000/01 with `IndicatorMode` slightly disagrees with the solution from all solvers using the `SOS1Mode`. `FortunyAmatMcCarlMode` and `StrongDualityMode` seem very amenable to MIP solvers, with Gurobi closing the gap within the given 10 minutes for all but one instance in the latter mode. However, we must be careful since we selected arbitrary bounds for those methods and `StrongDualityMode` also relies on binary expansion approximations, which led solvers to a solution that disagrees with the other methods on the 10/05 instance. On the other hand, `ProductMode` is the worst strategy for MIP solvers in these instances. For NLP solvers, both `ProductMode` and `StrongDualityMode` return objective values that are close to the ones found by MIP solvers, but in this case there is a slight advantage for `ProductMode`. Finally, Gurobi NonConvex seems to work much better with `StrongDualityMode`, claiming very good results in the instances with 1000 samples that agree with some of the other presented objective values.

The results are particular to a toy problem. However, the tables demonstrate that the software can interface with multiple solvers and consider multiple methods. Moreover, there is value in experimenting with multiple solvers and methods implemented in BilevelJuMP.

		CPLEX			Gurobi			SCIP			Xpress		
	Inst	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
SOS1Mode	10/01	0.30	0	0	0.30	0	0	0.30	0	0	0.30	0	0
	10/02	0.22	0	0	0.22	0	0	0.22	0	0	0.22	0	0
	10/05	0.09	0	0	0.09	0	0	0.09	0	0	0.09	0	0
	100/01	2.42	0	0	2.42	0	0	2.42	0	0	2.42	0	0
	100/02	2.40	4	-	2.40	4	-	2.40	4	-	2.40	4	-
	100/05	2.30	6	-	2.30	6	-	2.31	6	-	54.87	-	-
	100/10	8.54	392	-	79.59	-	-	79.59	-	-	79.59	-	-
	100/20	102.79	-	-	8.21	457	-	102.79	-	-	96.89	-	-
	100/50	23.35	307	-	23.35	299	-	23.35	-	-	23.35	350	-
	1000/01	25.02	0	-	28.63	14	-	25.02	0	-	25.02	0	-
	1000/02	323.30	-	-	323.30	-	-	323.30	-	-	323.30	-	-
	1000/05	533.37	-	-	533.37	-	-	533.37	-	-	533.37	-	-
IndicatorMode	10/01	0.30	0	0	0.30	0	0	0.30	0	0	0.30	0	0
	10/02	0.22	0	0	0.22	0	0	0.22	0	0	0.22	0	0
	10/05	0.09	0	0	0.09	0	0	0.09	0	0	0.09	0	0
	100/01	2.42	0	0	2.42	0	0	2.42	0	2	2.42	0	2
	100/02	2.40	4	-	2.40	4	-	9.08	294	-	2.42	5	-
	100/05	2.30	6	-	2.30	6	-	39.08	-	-	2.31	6	-
	100/10	79.59	-	-	79.59	-	-	79.59	-	-	79.59	-	-
	100/20	102.79	-	-	-	-	-	102.79	-	-	102.79	-	-
	100/50	23.35	-	-	23.35	-	-	23.35	-	-	23.35	576	-
	1000/01	25.06	0	-	-	-	-	77.45	209	-	195.11	680	-
	1000/02	323.30	-	-	-	-	-	323.30	-	-	-	-	-
	1000/05	533.37	-	-	533.37	-	-	533.37	-	-	-	-	-

Table 2.2: Mixed Integer Programming solvers with `SOS1Mode` and `IndicatorMode`, Time in seconds (s), Gap in percent (%).

		CPLEX			Gurobi			HiGHS			SCIP			Xpress		
	Inst	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
FortunyAmatMcCarlMode	10/01	0.30	0	0	0.30	0	0	0.30	0	0	0.30	0	0	0.30	0	0
	10/02	0.22	0	0	0.22	0	0	0.22	0	0	0.22	0	0	0.22	0	0
	10/05	0.09	0	0	0.09	0	0	0.09	0	0	0.09	0	0	0.09	0	0
	100/01	2.42	0	0	2.42	0	0	2.42	0	0	2.42	0	1	2.42	0	0
	100/02	2.40	4	-	2.40	4	-	2.40	4	-	2.43	5	-	2.43	5	-
	100/05	2.30	6	-	2.29	5	-	54.87	-	-	39.08	-	-	2.31	6	-
	100/10	79.59	-	-	2.33	34	-	79.59	-	-	79.59	-	-	79.59	-	-
	100/20	22.11	-	-	22.17	-	-	102.79	-	-	102.79	-	-	-	-	-
	100/50	23.35	355	-	23.35	330	-	23.35	-	-	23.35	952	-	23.35	736	-
	1000/01	25.02	0	-	25.02	0	-	25.02	0	-	70.13	180	-	25.02	0	-
	1000/02	24.46	3	-	23.74	0	12	323.30	-	-	323.30	-	-	23.75	0	-
	1000/05	533.37	-	-	533.37	-	-	533.37	-	-	533.37	-	-	533.37	-	-
ProductMode	10/01	0.30	0	344	0.30	0	17	0.30	0	269	0.30	0	83	0.30	0	447
	10/02	1.82	739	-	0.22	0	-	3.12	-	-	0.33	0	495	0.30	38	-
	10/05	0.53	-	-	7.22	-	-	8.72	-	-	0.67	-	-	0.60	-	-
	100/01	18.08	647	-	14.37	494	-	23.28	0	52	21.87	803	-	20.74	0	55
	100/02	27.06	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	100/05	-	-	-	48.56	-	-	-	-	-	-	-	-	-	-	-
	100/10	-	-	-	75.06	-	-	-	-	-	-	-	-	78.62	-	-
	100/20	-	-	-	99.82	-	-	-	-	-	-	-	-	101.51	-	-
	100/50	247784.27	-	-	183.62	-	-	-	-	-	-	-	-	-	-	-
	1000/01	45.15	80	-	-	-	-	-	-	-	58.77	135	-	147.68	490	-
	1000/02	-	-	-	-	-	-	-	-	-	-	-	-	165.04	595	-
	1000/05	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
StrongDualityMode	10/01	0.30	0	512	0.30	0	-	0.30	0	-	0.30	0	-	0.30	0	-
	10/02	0.22	0	0	0.22	0	0	0.22	0	7	0.22	0	36	0.22	0	52
	10/05	0.00	0	2	0.00	0	3	0.00	0	165	0.00	0	37	0.00	-	4
	100/01	2.42	0	2	2.42	0	1	2.42	0	2	-	-	-	2.42	0	0
	100/02	2.30	0	7	2.30	0	10	2.30	0	63	2.36	2	-	-	-	-
	100/05	2.16	0	139	2.16	0	66	41.77	-	-	-	-	-	2.18	0	-
	100/10	1.73	0	269	1.73	0	34	75.84	-	-	78.61	-	-	-	-	-
	100/20	90.35	-	-	1.45	0	269	-	-	-	-	-	-	1.57	8	-
	100/50	176.21	-	-	160.70	-	-	-	-	-	-	-	-	-	-	-
	1000/01	25.01	0	153	25.01	0	22	25.01	0	32	25.01	0	20	25.01	0	10
	1000/02	23.74	0	39	23.74	0	23	23.75	0	-	-	-	-	23.74	0	20
	1000/05	529.71	-	-	24.43	0	543	408.46	-	-	-	-	-	24.43	0	482

Table 2.3: Mixed Integer Programming solvers with `ProductMode`, `FortunyAmatMcCarlMode`, and `StrongDualityMode`, Time in seconds (s), Gap in percent (%).

		Gurobi NonConvex			Ipopt			Knitro		
	Inst	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
`ProductMode`	10/01	0.31	2	-	0.32	-	1	0.32	-	0
	10/02	0.22	3	-	0.22	-	1	0.22	-	0
	10/05	0.67	-	-	0.09	-	0	0.09	-	0
	100/01	2.42	0	6	2.43	-	18	2.46	-	0
	100/02	2.71	17	-	2.41	-	12	2.43	-	0
	100/05	54.87	-	-	2.47	-	24	2.54	-	0
	100/10	79.59	-	-	2.64	-	12	2.35	-	0
	100/20	102.79	-	-	3.43	-	11	3.51	-	0
	100/50	185.45	-	-	19.46	-	22	19.46	-	0
	1000/01	25.21	0	-	25.13	-	272	25.10	-	1
	1000/02	323.30	-	-	24.08	-	148	23.84	-	2
	1000/05	533.37	-	-	25.70	-	121	24.89	-	4
`StrongDualityMode`	10/01	0.30	1	-	0.32	-	0	0.33	-	0
	10/02	0.25	17	-	0.22	-	0	0.22	-	0
	10/05	0.13	-	-	0.09	-	0	0.09	-	0
	100/01	2.42	0	0	2.44	-	2	2.43	-	0
	100/02	2.41	4	-	2.44	-	2	2.53	-	0
	100/05	2.33	7	-	2.50	-	0	2.32	-	0
	100/10	2.41	39	-	2.21	-	1	2.08	-	0
	100/20	3.55	145	-	3.43	-	8	2.90	-	0
	100/50	64.20	-	-	23.34	-	1	185.45	-	0
	1000/01	25.03	0	-	54.89	-	3	25.08	-	3
	1000/02	23.80	0	-	71.32	-	9	23.78	-	6
	1000/05	24.86	1	-	29.52	-	175	24.75	-	4

Table 2.4: Gurobi NonConvex and NLP solvers with `ProductMode` and `StrongDualityMode`, Time in seconds (s), Gap in percent (%).

2.6 Package Comparison

A comparison between BilevelJuMP.jl and four other bilevel optimization modeling interfaces that include solution methods is presented in Table 2.5. We include BilevelOptimization.jl as it was the key motivation for BilevelJuMP.jl; PAO, as the new bilevel interface of pyomo; GAMS that relies on EMP; and YALMIP which motivated the development of Dualization.jl.

In each table line, we briefly depict the answer to each of the following questions:

1. Which programming language does a user have to write the models?
2. What is the licensing scheme? (MIT is the most permissive among the ones shown).
3. Does the modeling interface support MIP solver-based methods, like SOS1 and big-M?
4. Does the modeling interface support NLP solver-based methods, like products?
5. Does the modeling interface support MPEC solvers that accept explicit complementarity constraints?
6. Can the user access dual variables of the lower-level problem and use them explicitly while modeling the upper-level problem?
7. Which problem classes are accepted in the lower-level problem?
8. Which problem classes are accepted in the upper-level problem?

The two latter questions used the following code: CP is Conic Programming, QP is linear programming with optional quadratic objective, NLP stands for Non-Linear Programming, VI represents Variational Inequalities, and Int is Integer Programming. Although one can model problems of given classes, specific solvers will be required to handle the resulting formulations. Finally, we note that all classes might not be supported simultaneously by all the interfaces, BilevelJuMP.jl supports all the described classes in the same model. Finally, it is worth mentioning that the possibility of considering bilevel models in which lower-level primal and dual variables are present in the first-level problem significantly enlarges the spectrum of practical applications that can be covered with the package. We marked YALMIP as ready to handle lower level duals because this can be achieved by explicitly calling their "kkt" function

on the lower level data and appending to the primal problem. For instance, strategic bidding as well as market-power assessments in electricity markets highly depend on bilevel models with such dependencies [136].

Name	BilevelJuMP.jl	BilevelOptimization.jl	PAO/Pyomo	GAMS	YALMIP
Language	Julia	Julia	Python	GAMS	MATLAB
License	MIT	MIT	BSD	Commercial	YALMIP
MIP solvers	Yes	Yes	Yes	No	Yes
NLP solvers	Yes	No	Yes	Yes	Yes
MPEC solvers	Yes	No	No	Yes	Yes
DualVar	Yes	No	No	Yes	Yes
Lower Level	CP/QP	QP (matrix form)	QP/Int	QP/NLP/VI	QP
Upper Level	CP/QP/NLP/Int	CP/QP/NLP/Int	QP/Int	QP/NLP/Int	CP/QP/NLP/Int

Table 2.5: Modeling interfaces for bilevel optimization

2.7 Conclusion

We presented BilevelJuMP, an open-source Julia library for Bilevel Optimization that allows the user to model a wide range of bilevel optimization problems very easily. Moreover, the user has access to multiple reformulation techniques that can be considered to handle different problems better. More specifically, BilevelJuMP.jl allows modeling very general problems at the upper level (all JuMP supported formulations, such as non-linear, conic, mixed integer constraints) and conic problems in the lower level. Additionally, it implements multiple MPEC-based reformulation techniques with MIP or NLP as solution algorithms. This broad and flexible infrastructure of models and methods all built in a single open-source package for JuMP allows practitioners to use BilevelJuMP.jl to quickly deploy and run experiments using state-of-the-art solvers and methods. It can be used by students introduced to the realm of bilevel optimization due to its easy-to-use and flexible structures, researchers and developers that can quickly test (or benchmark) new methods and models, and also develop new applications and packages, as well as by industry practitioners, who may not be familiar with bilevel solution strategies, but can rely on the package to address specific bilevel problems composing parts of real-world applications.

BilevelJuMP.jl joins a group of JuMP and MOI extensions that were made possible thanks to the good design of the latter two.

Just like JuMP and MOI, BilevelJuMP.jl is under active development, and more features are planned to be included. The library has gotten great interest from other contributors that are currently working towards new features, including support for integer variables in the lower level with the solver MibS [97]. Further developments include: implementing other techniques such as valid inequalities [137], column-constraint generation based techniques [109];

developing a file format for bilevel optimization based on MathOptFormat; integrating other solvers such as the one from [99].

2.8 Additional formulations and code

2.8.1 Lower-level duals

This modeling feature enables the implementation of workflows where one (or more) of the upper-level variables is the dual of a lower-level constraint. In particular, in the energy sector, it is common to model energy prices as the dual variable associated with the energy demand equilibrium constraint. One example of an application that uses this feature is [136], which focuses on strategic bidding in auction-based energy markets. A small and simplified example of the modeled problem would be the model:

$$\max_{\lambda, q_S} \quad \lambda \cdot g_S \tag{2-21}$$

$$\text{s.t.} \quad 0 \leq q_S \leq 100 \tag{2-22}$$

$$(g_S, \lambda) \in \underset{g_S, g_1, g_2, g_D}{\arg\min}\ 50g_{R1} + 100g_{R2} + 1000g_D \tag{2-23}$$

$$\text{s.t.} \quad g_S \leq q_S \tag{2-24}$$

$$0 \leq g_S \leq 100 \tag{2-25}$$

$$0 \leq g_1 \leq 40 \tag{2-26}$$

$$0 \leq g_2 \leq 40 \tag{2-27}$$

$$0 \leq g_D \leq 100 \tag{2-28}$$

$$g_S + g_1 + g_2 + g_D = 100 \quad : \quad \lambda \tag{2-29}$$

Where λ is the dual of the load balance constraint (2-29), g_S, g_1, g_2 represent the generation of the strategic bidder and from two other (non-strategic) plants. g_D represents the deficit in generation. Finally, q_S is the quantity bid optimized by the strategic generator.

BilevelJuMP.jl allows users to implement similar models using the function `DualOf` that binds a new variable in the upper level to an existing constraint in the lower level. The model can be written as:

```
model = BilevelModel()
@variable(Upper(model), 0 <= qS <= 100)
@variable(Lower(model), 0 <= gS <= 100)
@variable(Lower(model), 0 <= gR1 <= 40)
@variable(Lower(model), 0 <= gR2 <= 40)
@variable(Lower(model), 0 <= gD <= 100)
@objective(Lower(model), Min, 50gR1 + 100gR2 + 1000gD)
```

```
@constraint(Lower(model), gS <= qS)
@constraint(Lower(model), demand_equilibrium, gS + gR1 + gR2 + gD == 100)
@variable(Upper(model), lambda, DualOf(demand_equilibrium))
@objective(Upper(model), Max, lambda*gS)
```

2.8.1.1 NLP solution

This model can be solved by selecting a reformulation and a solver. Here we select Strong-Duality reformulation, the Ipopt solver and call optimize to perform the reformulation and solve it.

```
BilevelJuMP.set_mode(model, BilevelJuMP.StrongDualityMode())
set_optimizer(model, Ipopt.Optimizer)
optimize!(model)
```

2.8.1.2 MIP solution

It is also possible to solve such a problem by using a MIP formulation. The main issue is the product of variables in the upper level objective. However, this can be easily handled by using the aforementioned `QuadraticToBinary` package for automatic binary expansions. Because binary expansions require bounds on variables, we change the following line:

```
@variable(Upper(model), 0 <= lambda <= 1000, DualOf(demand_equilibrium))
```

Then, as before, we set a solver (now SCIP with the `QuadraticToBinary` wrapper) and a solution method (now Fortuny-Amat and McCarl):

```
set_optimizer(model,
    ()->QuadraticToBinary.Optimizer{Float64}(SCIP.Optimizer()))
BilevelJuMP.set_mode(model,
    BilevelJuMP.FortunyAmatMcCarlMode(dual_big_M = 100))
optimize!(model)
```

2.8.2
Conic Bilevel and Mixed Mode

Here we present a simple bilevel program with a conic lower level model described in example 3.3 from [138].

$$\max_{x \in \mathbb{R}} \quad x + 3y_1 \tag{2-30}$$

$$\text{s.t.} \quad 2 \leq x \leq 6 \tag{2-31}$$

$$y(x) \in \arg\min_{y \in \mathbb{R}^3} -y_1 \tag{2-32}$$

$$\text{s.t.} \quad x + y_1 \leq 8 \tag{2-33}$$

$$x + 4y_1 \geq 8 \tag{2-34}$$

$$x + 2y_1 \leq 12 \tag{2-35}$$

$$y \in SOC_3 \tag{2-36}$$

In this problem, most of the constraints are regular linear constraints, while the last one, (2-36), is a second order cone constraint. Such constraint ensures that the vector y belongs to a second order cone of dimension 3, that is: $y_1 \geq \sqrt{y_2^2 + y_3^2}$. This problem can be encoded using regular JuMP syntax for conic programs:

```
model = BilevelModel()
@variable(Upper(model), x)
@variable(Lower(model), y[i=1:3])
@objective(Upper(model), Min, x + 3y[1])
@constraint(Upper(model), x >= 2)
@constraint(Upper(model), x <= 6)
@objective(Lower(model), Min, - y[1])
@constraint(Lower(model), con1, x +  y[1] <=  8)
@constraint(Lower(model), con2, x + 4y[1] >=  8)
@constraint(Lower(model), con3, x + 2y[1] <= 12)
@constraint(Lower(model), con4, y in SecondOrderCone())
```

2.8.2.1
NLP solution and start values

We can set, for instance, the product reformulation and select Ipopt as a solver. As Ipopt does not have native support for second order cones, we use the non-default MOI bridge `SOCtoNonConvexQuad` to convert second order cones into quadratic constraints.

```
BilevelJuMP.set_mode(model,BilevelJuMP.ProductMode(1e-5))
set_optimizer(model,
    ()->MOI.Bridges.Constraint.SOCtoNonConvexQuad{Float64}(Ipopt.Optimizer()))
optimize!(model)
```

This problem is very simple, but more complex models might require more information, such as starting points that can be passed at the variable creation with standard JuMP syntax, for instance:

```
@variable(Upper(model), x, start = 6)
```

The user could also use the alternative JuMP syntax:

```
set_start_value(x, 6)
set_dual_start_value(con2, 0)
```


2.8.2.2 MIP solution and mixed mode

Alternatively, we could have used a Mixed Integer Second Order Cone Program (MISOCP) solver together with binary expansions. Complementarity of conic constraints is more difficult to handle because they require a sum of products that cannot be reformulated with other methods. Therefore, we rely on product reformulation for conic constraints. However, we can use other reformulations like indicator constraints for the non-conic constraints. Mixing the two of them can be done with Mixed Mode from Section 2.4.1.6.

The following code describes how to solve the problem with a MISOCP based solver.

```
set_optimizer(model,
    ()->QuadraticToBinary.Optimizer{Float64}(Xpress.Optimizer(),lb=-10,ub=10))
BilevelJuMP.set_mode(model,
    BilevelJuMP.MixedMode(default = BilevelJuMP.IndicatorMode()))
BilevelJuMP.set_mode(con4, BilevelJuMP.ProductMode(1e-5))
optimize!(model)
```

We set the reformulation method as Mixed Mode and select Indicator constraints to be the default for the case in which we do not explicitly specify the reformulation. Then we set the product mode for the second order cone reformulation.

As described in Section 2.8.1, binary expansions require bounded variables, hence the `QuadraticToBinary` meta-solver accepts fallback to upper and lower bounds (`ub` and `lb`), used for variables with no explicit bounds.

3
Application-Driven Learning: A Closed-Loop Prediction and Optimization Approach Applied to Dynamic Reserves and Demand Forecasting

Forecasting and decision-making are generally modeled as two sequential steps with no feedback, following an open-loop approach. In this paper, we present application-driven learning, a new closed-loop framework in which the processes of forecasting and decision-making are merged and co-optimized through a bilevel optimization problem. We present our methodology in a general format and prove that the solution converges to the best estimator in terms of the expected cost of the selected application. Then, we propose two solution methods: an exact method based on the KKT conditions of the second-level problem and a scalable heuristic approach suitable for decomposition methods. The proposed methodology is applied to the relevant problem of defining dynamic reserve requirements and conditional load forecasts, offering an alternative approach to current *ad hoc* procedures implemented in industry practices. We benchmark our methodology against the standard sequential least-squares forecast and dispatch planning process. We apply the proposed methodology to an illustrative system and to a wide range of instances, from dozens of buses to large-scale realistic systems with thousands of buses. Our results show that the proposed methodology is scalable and yields consistently better performance than the standard open-loop approach.

3.1 Introduction

The most common approach to make decisions under uncertainty involves three steps. In the first step, one develops a forecast for all uncertainties that affect the decision-making problem based on all information available. In the second step, an action based on the forecast is selected. Finally, in the third step, one implements corrective actions after uncertainties are realized. This three-step procedure constitutes an open-loop forecast-decision process in which the outcomes of the decisions are not considered in the forecasting framework.

In the electricity sector, it is common for system operators to use an open-

loop forecast-decision approach. First, loads are forecast based on standard statistical techniques, such as least squares (LS), and reserve requirements are defined by simple rules, based on quantiles, extreme values and standard deviation of forecast errors according to specified reliability standards [139]. Then, a decision is made to allocate generation resources following an energy and reserve scheduling program [53, 54]. In real-time, reserves are deployed to ensure that power is balanced at every node, compensating for forecast errors.

From the academic perspective, it has been demonstrated that stochastic programming models yield better results than deterministic ones when making decisions under uncertainty because the former takes distributions into consideration. These models provide better results in terms of cost, reliability, and market efficiency compared to deterministic approaches [140]. Nevertheless, in practical applications, two issues arise: proper modeling of distributions is challenging and tractability imposes small sample sizes for techniques like sample average approximations (SAA). A consequence of this tractability issue is that SAA solutions become sample dependent [141, 142], thereby compromising market transparency and preventing stakeholders' acceptance [143]. Therefore, most system operators worldwide still rely on deterministic short-term scheduling (economic dispatch or unit commitment, UC) models with exogenous forecasts for loads and reserve requirements [53, 144]. Within this context, one alternative to improve the performance of deterministic scheduling tools is to forecast load and reserve requirements with the goal of minimizing energy and reserve scheduling costs.

There is empirical evidence that system operators rely on *ad hoc* or out-of-market actions—and not just on reserves— to deal with uncertainty in operations. According to the 2019 Annual Report on Market Issues and Performance of the California ISO [145], *"...operators regularly take significant out-of-market actions to address the net load uncertainty over a longer multi-hour time horizon (e.g., 2 or 3 hours). These actions include routine upward biasing of the hour-ahead and 15-minute load forecast, and exceptional dispatches to commit and begin to ramp up additional gas-fired units in advance of the evening ramping hours."* Additionally, reserve requirements are, in practice, empirically defined according to further *ad hoc* off-line rules based on off-line analysis [139, 144]. These *ad hoc* procedures lack technical formalism and transparency to minimize operating and reliability costs, which prevents agents from internalizing the forecasts in their bids. On the other hand, a scientifically grounded method as the one described here would allow incorporating biased forecasts in bids. Consequently, this challenging real-world application requires further research observing the practical issues that need to be ad-

dressed to improve the current state-of-the-art of industry practices.

For years, decision-making and forecasting have been treated as two completely separate processes [146]. Many communities, such as Statistics and Operations Research, have studied these problems and developed multiple tools combining probability and optimization. The machine learning community, which combines many ideas from optimization and probability, has also been tackling such tasks and has proposed methods to treat them jointly [59].

Classical forecasting methods do not take the underlying application of the forecast into account. Consequently, hypotheses such as prediction error symmetry in least squares (LS) might not be the best fit for problems with asymmetric outcomes. By acknowledging the asymmetry in particular problems, researchers have attempted to capture it empirically; however, such an approach does not take the application into account directly. Some existing methods do capture asymmetry, such as Quantile Regression [147]. The interest in exploring asymmetric loss functions is not new. For instance, [148] and [149] acknowledge that biased estimators can perform even better than those that make accurate predictions of statistical properties of the stochastic variables. The author exemplifies that an overestimation is not as bad as an underestimation for the case of dam construction and attributes a second example about the asymmetry on real estate assessment to [150].

Within this context, two possible avenues of research are opened to achieve better results: i) a focus on improving the decision-making model (prescriptive framework), which assumes we can change it to consider embedded co-optimized forecasts [146]; or ii) a focus on improving the forecasting model (predictive framework), which assumes we can not change the decision-making process (in our application, defined by system operators' dispatch models), but we can change the forecasts to incorporate, in a closed-loop manner, a given application cost function [59, 151, 152]. Therefore, in this paper, we focus on the latter avenue. In Section 3.2, we provide a literature review on this subject.

3.1.1 Objective and contribution

The objective of this paper is to present a new closed-loop *application-driven learning* framework to be used in point forecast applications. In the proposed method, the application is characterized through an optimization model, which is then used as part of the estimation problem. In this new framework, both *ex-ante* (planning) and *ex-post* (implementation/assessment) cost-minimization structures of the decision-maker, i.e., the *application schema*, are considered in the prediction process. Therefore, our framework replaces the

traditional statistical error minimization objective with a cost-minimization structure of a specific application.

To achieve the objectives described before, we derive the following general technical contributions:

– A new and flexible application-driven learning framework based on a bilevel optimization model. To the best of the authors' knowledge, for the first time in the literature, the relevant case of applications based on linear programming models affected by right-hand-side uncertainty is addressed with specialized algorithms. Two solution approaches are presented. The first approach is an exact method based on the KKT conditions of the second-level problem. The second is a scalable heuristic approach suitable for decomposition methods and parallel computing. Although not limited to linear bilevel programs, we show how to design efficient methods tailored for off-the-shelf linear optimization solvers. Additionally, our scalable heuristic method ensures optimal second-level solutions. This is a salient feature of our method in contrast with other methods that rely on surrogates of the second level [151] or solve approximations of the KKT conditions with non-linear solvers [152]. In this context, the proposed framework is general and suitable for a wide range of applications relying on the standard structure of the forecast-decision process.

– We provide new asymptotic convergence proofs for both the objective function value and estimated parameters of the proposed application-driven learning method. The convergence proof is completely novel and highlights that the method is asymptotically the best that can be done for a specific application (described by planning and implementation processes) given a forecast functional form (see Corollary 1). It is important to emphasize that Lemma 1 extends the state-of-the-art SAA results from [153] to stationary-ergodic time series, which is also a relevant contribution to the subject. Under the hypothesis of our methodology and based on the aforementioned convergence results, we show that the solution of our method converges to the best estimator in terms of the expected cost of the selected application.

Notwithstanding, our paper also provides relevant contributions to the applied field of power systems operation. In particular, the application of this paper focuses on the problem of demand and reserve requirements forecasting for power system operators. This is a critical problem of the power systems industry, which has not been addressed by previous works on the subject and

is currently being tackled by system operators through *ad hoc* out-of-market actions to address the net load uncertainty [145]. Therefore, as a contribution to this applied field, we propose a new methodology to forecast the load and define the reserve requirements for the power-system dispatch planning application. The method can be used to either jointly optimize the load forecast and reserve requirements or to optimize only the reserve requirements given an exogenous forecast for the load. In both cases, the optimal solution defines the optimal policy to dynamically allocate reserve requirements so that the expected cost is minimized in the long run. In both cases, agents are provided with a scientifically grounded and comprehensively described methodology that can be used to reduce the number of *ad hoc* procedures currently implemented in practice.

In our specific application, the bilevel formulation can be summarized as follows. The first level accounts for both the predictive model specification (parameter selection) and the cost evaluation metric based on the actual operation of the system for many data points. It is relevant to mention that the methodology is flexible to internalize and address relevant practical issues, such as reserve requirement constraints imposed by regulatory rules, reliability standards [154, 139], and risk-aversion metrics [153]. In the second level, the *ex-ante* energy and reserve scheduling process of the system operator is accounted for based on 1) a conditional demand forecast and 2) the definition of adjusted (dynamic) reserve requirements, both defined in the first level as a function of previous data for each point of the training set. Thus, in our bilevel model, we have multiple parallel lower-level problems, each of which represents the one-step-ahead deterministic two-stage scheduling process performed by the system operator for each point of the training dataset. In this context, the second level ensures closed-loop feedback characterizing joint scheduling decisions of energy and reserve allocations without perfect information on the target period data. Such a closed-loop formulation applied to the definition of *conditional load forecasts* and *dynamic reserve requirements* is a salient and original contribution to the subject of power systems operation.

Finally, to empirically corroborate the relevance of our contributions and to demonstrate the applicability, performance, and scalability of our methodology, we benchmark the proposed method with the traditional sequential least squares forecast and energy and reserve scheduling approach. To do that, we analyze the proposed methodology in several case studies using multiple test systems. First, we present studies with an illustrative single-bus system to explore multiple properties of the methods. Second, we apply the method to various instances based on the IEEE 24-, 118-, and 300-bus test systems to

show that the results are of high quality. Third, we consider 10 data sets ranging from 600 to 6000 buses created artificially by combining instances of the 300 bus system to analyze the scalability of the method. Fourth, we consider realistic, very large-scale systems, ranging from 6,468 to 13,659 buses, with infrastructure and conditions that are very close to those of a systems operator performing hour-ahead planning to demonstrate that the methodology can be seriously considered for deployment by operators. Results show that the two proposed application-driven learning approaches (demand and reserves, and only reserves) yield consistently better performance on out-of-sample tests than the benchmark where forecasts and decisions are sequentially carried out. For large and very large systems, where the exact method fails to find solutions within reasonable computational times, the heuristic method exhibits high-quality performance compared to the benchmark for all test systems.

3.2 Literature Review

We review the literature on 1) forecast models jointly optimized for a given application, hereinafter referred to as application-driven forecast, and 2) uncertainty forecasting and reserve sizing.

3.2.1 Application-driven forecast models

The ingenious idea of integrating the process of forecasting and optimizing a downstream problem was first proposed in the seminal paper by [59]. More than twenty years ago, the author emphasized the importance of estimating parameters with the correct goals in mind. In that work, a Neural Network (NN) is trained to forecast stocks with an objective function that describes the portfolio revenue given an allocation based on stocks forecast. Still, in the finance sector, [155] proposed new neural network structures and presented extremely promising benchmarks. An attempt to lower the burden of the method was proposed by [156]; the idea is to train multiple prediction models with standard regressions, but choose the best one in the out-of-sample analysis considering the proper application-driven objective function. The work by [157] presents another intermediary methodology. The model for estimating forecasts includes both the application objective function and the fitness measure similar to maximum likelihood estimation (MLE). A bi-objective problem is solved with scalarization; the authors look for a good balance between MLE and application value.

Following the key idea of [59] closely, the work by [158] presents a generic

algorithm to deal with parameter optimization of forecasting models embedded in stochastic programming problems, that is, parameters estimated considering the loss function of the actual problem. The algorithm is based on the stochastic gradient descent (SGD) method and employs tools for automatically differentiating strongly convex quadratic optimization problems. The method is applied to small prototypical quadratic programming problems and the local solutions obtained are shown to be promising. The idea of extending to combinatorial problems via convexification and regularization (to make it a QP) schemes is presented in [159].

More recently, the work Smart "Predict and Optimize" (SPO) [151], recognizes the importance of closed-loop estimation. The authors develop an algorithm for the linear programming case that is based on relaxation and convexification of the nonlinear loss function before applying a tailored SGD approach, instead of looking for local solutions with nonlinear methods. To develop the algorithm, the authors limit themselves to linear dependency on features and restrict uncertainty to the objective function. This contrasts with our work that preserves the original planning and implementation functions (detailed in Section 3.3) and focuses on non-linear methods to obtain global or local solutions. Theoretical improvements were made to the SPO method in [160], which presents new bounds for the generalization capability of the method. Also following SPO, [161] presents an adaptation of SPO for the case of combinatorial problems. SPO is similar to the method proposed by [162] to estimate uncertain objective coefficients without considering features in a different context.

While working on this paper, the authors became aware of the work by [152] (developed simultaneously with ours, by a completely different team) that guards many conceptual similarities to the general version of our proposed model. The work of [152] also focuses on the idea of finding the best forecast for a given application (or context) through a bilevel framework. The framework proposed in [152] is applied to estimate a parameter of the inverse demand curve of a Cournot strategic producer bidding in forward markets. The scalability of their model relies on a nonlinear relaxation of the right-hand side of the complementarity constraints. In our method, we adopt a different approach to overcome the issue of suboptimal lower-level solutions and to tackle very large-scale problems, as will be explained in Section 3.8.4. Moreover, we prove convergence of our method, whereas [152] only analyze the convergence in a particular example and point to a preprint of our work when discussing the requirement of lower level uniqueness in their bilevel program.

[163] describes a framework named Decision Based Model Selection,

which fits in the general grand scheme of properly combining forecasting and optimizing decisions. The authors propose a two-step procedure. First, they estimate a forecasting model with available data and generate more samples with this model; then, with a new (artificial) data set in hand, the model (forecasting + optimization model + algorithm) with the best decisions is chosen from a discrete list. The framework is general enough to allow forecasts that depend on decisions. Related work in power systems is [164] that proposed to generate multiple forecasting models and then select the best one according to a min-max regret based on the application cost.

[165] and [166] describe the so-called Learning Enabled Optimization (LEO). LEO is a framework to combine Statistical Learning (SL), Machine Learning (ML) and (stochastic) Optimization. The idea is to compare a set of predefined SL/ML models with the cost value of the actual application in mind. The main difference here is that the ML/SL are still estimated based on classical methods. More recently, LEO has been extended to Coupled LEO to accommodate decision-dependent uncertainty [167] solved by derivative-free methods.

Also related is the field of Optimal Learning [168, 169]. Optimal Learning is very broad, encompasses many techniques and can be applied to an enormous range of problems, as detailed by the previous references. The idea of estimating unknowns through experimentation and testing guesses in simulation models is a possible framework for choosing forecast models based on applications. Deeply related to Optimal Learning is Bayesian Optimization [170, 171], which is a set of methods used to optimize continuous functions, typically low dimensional (possibly noisy) functions that are expensive to evaluate, which might be the case of an application as a cost function.

Not surprisingly, some of the above works formulate the problems as bilevel optimization problems, which is mostly done implicitly and rarely explicitly [152]. For more information on bilevel optimization, the reader is directed to [30]. The work by [32] lists hundreds of references related to bilevel optimization, including papers related to parameter optimization. Parameter optimization is frequently modeled as bilevel optimization and has been drawing the attention of many fields, such as ML, control, energy systems, and game theory. This can be thought of as a version of the closed-loop paradigm since these works target the best parameters for algorithms and applications. We refer to [5] for applications in hyper-parameter tuning. Under a broader ML umbrella, our methodology can be seen as an ML forecast approach where the *application schema* is embedded into the method through its explicit mathematical programming formulation.

3.2.2 Uncertainty forecasting and reserve sizing

The operation of power systems has been profoundly related to uncertainty handling. The electric load has been among the main challenges for forecasters in power systems for many years. Researchers around the world have proposed the most varied methodologies, ranging from standard linear regressions to Neural Networks (NN); these techniques, along with many others, are reviewed in [48] and [172]. The forecast of variable renewable energy sources is probably one of the current big challenges power systems. Although many techniques are already available, forecasting renewable generation such as from wind farms has proven to be significantly harder than load [172, 173]. As shown in reviews [174, 172], wind and solar forecasting are divided into two main trends: i) physical-based methods that rely on topographic models and Numerical Weather Predictions; ii) statistical methods, including Kalman filters, ARMA models, and NN.

Although forecast methods have consistently improved in the last years, the systems must be ready to withstand deviations from predicted values. The widely used approach is to allocate reserves in addition to the power scheduled for each generator to meet demand forecasts. The additional power is scheduled as reserve margins to give the system operator flexibility to handle real-time operations. Many methods have been proposed to account for the variations in loads, contingencies, and variable renewable energy [175]. Furthermore, different reserve sizing rules are applied by different ISOs all over the world [139]. These rules vary from deterministic *ad hoc* procedures to more statistical-oriented guidelines.

In [139], examples of real-life *ad hoc* procedures to allocate reserves are presented, most of them relying on static approaches. Although time-varying reserves have been studied in the past, they have re-emerged as dynamic probabilistic reserves [54]. In the context of a more sophisticated dynamic probabilistic reserve approach, probabilistic forecasts are frequently used to account for forecast errors. These probabilistic reserves can be sized following a variety of methods with different complexity based on: forecast error standard deviations [176, 175], non-parametric estimation of the forecast error distribution [177], or even machine learning [54]. These are all considered stochastic methods and are simple alternatives to capture and incorporate fairly complex dynamics that are challenging for bottom-up approaches.

A prominent alternative to the use of reserves in power systems is the Stochastic Unit commitment. In such applications, many types of reserves can be defined endogenously, targeting cheaper operations on average. However,

as described in the review on Unit Commitment by [178], there are at least three main barriers toward the wide acceptance of stochastic unit commitment: i) uncertainty modeling, ii) computational performance, and iii) market design. Uncertainty modeling is jointly tackled by statistical modeling of the uncertainty concerning scenario generation and forecasting and by a decision-making framework like risk-averse stochastic optimization, robust optimization, and so on. Computational performance is the focus of many works like the Lagrangian decomposition [179], improved formulations [180], progressive hedging [181]. Notwithstanding the relevant recent advances in this area, the computational burden and the consequential instability of solutions under small sample sizes still preclude the acceptance of stochastic UC models by ISOs. Finally, the least studied challenge is market design. It requires experimenting and developing rules that are both feasible to be implemented and accepted by stakeholders [182, 143]. Based on previously reported industry practices and since ISOs currently follow the alternate route and tackle the uncertainty of UC with reserves [143], we will also follow this approach to propose a readily practical method.

3.3 Application-Driven Learning and Forecasting

In this section, we contrast the standard sequential framework, referred to as *open-loop*, and the joint prediction and optimization model, referred to as *closed-loop*. The presentation is in general form to facilitate the description of the solution algorithm, to set notation for the convergence results and to highlight that the method has applications beyond load forecasting and reserve sizing in power systems. We will specialize the bilevel optimization problem for closed-loop load forecasting and reserve sizing in Section 3.7.

We consider a dataset of historical data $\{y_t, x_t\}_{t\in\mathbb{T}}$, where $\mathbb{T} = \{1, \ldots, T\}$. Here y_t are observations of a variable of interest that we want to forecast, while x_t are observations of external variables (covariates or features) that can be used to explain the former. Furthermore, the latter might include lags of y_t as in autoregressive time series models. Additionally, it is worth mentioning that both y_t and x_t can be vector-valued.

The classic forecast-decision approach works as follows. The practitioner *trains* a parametric *forecast* model seeking for the best vector of parameters, θ, such that a loss function, $l(\cdot,\cdot)$, between the conditional forecast for sample t, $\hat{y}_t(\theta, x_t)$, and the actual data, y_t, is minimized, i.e., solving $\min_\theta \frac{1}{T}\sum_t l(\hat{y}_t(\theta, x_t), y_t)$. This is frequently done by solving LS optimization problems and finding $\theta^{LS} \in arg\min_\theta \frac{1}{T}\sum_t \|\hat{y}_t(\theta, x_t) - y_t\|^2$. In the *planning*

step, a decision is made by an optimized policy based on the previously obtained forecast, i.e., with $\hat{y}_t^{LS} = \hat{y}_t(\theta^{LS}, x_t)$. This results in a vector $z^*(\hat{y}_t^{LS})$, which in our application comprises the schedule of energy and reserves through generating units. Finally, the actual data y_t is observed, and the decision-maker must adapt to it, for instance, the system operator responds with a balancing re-dispatch, and a *cost*, $G_a(z^*(\hat{y}_t^{LS}), y_t)$, is measured. There is no feedback of the final cost into the forecasting and decision policy, hence, the name *open-loop*.

3.3.1
The proposed closed-loop application-driven framework

The core of the proposed predictive framework is to explore a feedback structure between the estimated predictive model and the application cost assessment. The general idea is depicted in Figure 3.1, which also stresses the difference from the open-loop model.

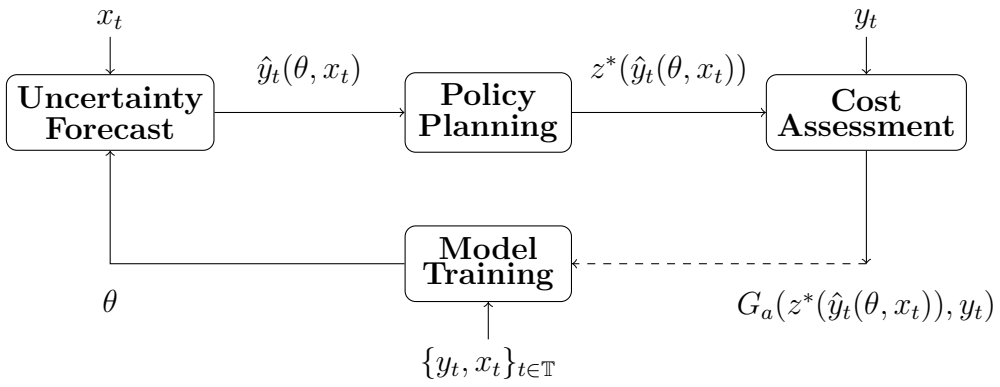


Figure 3.1: Learning models: considering the dashed line we have the *closed-loop* model, otherwise it represents the *open-loop* model.

The estimation method can be mathematically described through the following bilevel optimization problem (BOP):

$$\theta_T \in \underset{\theta\in\Theta,\hat{y}_t,z_t^*}{\arg\min} \quad \frac{1}{T}\sum_{t\in\mathbb{T}} G_a(z_t^*, y_t) \tag{3-1}$$

$$s.t. \quad \hat{y}_t = \Psi(\theta, x_t)\ \forall t \in \mathbb{T} \tag{3-2}$$

$$z_t^* \in \underset{z\in Z}{\arg\min}\, G_p(z, \hat{y}_t)\ \forall t \in \mathbb{T}, \tag{3-3}$$

where, for $i \in \{a, p\}$,

$$G_i(z, y) = c_i^\top z + Q_i(z, y) \tag{3-4}$$

$$Q_i(z, y) = \min_u \{q_i^\top u \mid W_i u \geq b_i - H_i z + F_i y\} \tag{3-5}$$

Note that the functions in (3-4) and (3-5) resemble the formulation of two-stage stochastic programs, in the sense that given a decision z and an observation y, one determines the best corrective action u. In that context, $c_i,\ q_i,\ W_i,\ b_i,\ H_i$ and F_i $(i \in \{a, p\})$ are parameters defined according to the problem of interest. The a subscript refers to *assessment*, while the p subscript refers to *planning*. Note also that the uncertainty y appears only on the right-hand side of the problems defining Q_a and Q_p; this will be important for our convergence analysis and solution methods.

In model (3-1)–(3-5), $\Psi(\theta, x_t)$ represents a *forecasting* model that depends on both the vector of parameters, θ, and the features vector, x_t, possibly including lags of y_t. The vector $\hat{y}_t$ is the forecast generated for sample (or period) t (comprising load and reserve requirements) conditioned to the vector of features, x_t, as defined in (3-2). For each t, the forecast $\hat{y}_t$ is used as input in a second-level problem and a decision planning policy, z_t^*, is obtained as a function of $\hat{y}_t$, i.e., $z_t^*(\hat{y}_t)$. This is done by optimizing the decision-maker *planning* cost function, $G_p(z, \hat{y}_t)$ in (3-3). Then, the optimized policy z_t^* is evaluated in the first level against the actual realization, y_t, for each t. The evaluation is made under the decision-maker's *assessment* (or *implementation*) cost function, $G_a(z_t^*(\hat{y}_t), y_t)$. Hence, the application is embedded into the estimation process in both the *ex-ante* planning policy and *ex-post* implementation objective (3-1)–(3-5). It is worth noticing that the proposed formulation can be interpreted as an optimization over θ in a back-test, in which for a given θ, the assessment of the forecast performance is completely determined by $G_a(z_t^*(\hat{y}_t), y_t)$. Within this context, the upper level identifies the parameters with the best back-test performance. Furthermore, note that for a fixed θ, there is no coupling between two samples, thereby the model can be decomposed per t. This will be used in one of the proposed

solution methods. Finally, there is a slight abuse of notation in (3-1) because the argmin only retrieves θ_T, a solution for θ with T samples, thus disregarding the rest of the first level decision vectors, $\hat{y}_t$ and z_t^*. Note that θ_T is one among the multiple options in the set of all possible solutions S_T.

One key difference from previous works [158, 151, 152] is that G_a and G_p can be different functions. This is extremely useful in the context of power systems operations where planning models might differ from real-time ones. Although model (3-1)–(3-3) is fairly general, we specialize to the case of linear programs and right-hand-side uncertainty, (3-4)–(3-5), and we will assume polyhedral structure for the set Z in (3-3). This can be contrasted with previous works that considered strongly quadratic programs [158] and objective uncertainty [151]. As mentioned earlier, this specialization will be important for developing our asymptotic convergence results and our solution methods.

3.4 A Motivating Example

In this interlude, we present a small and illustrative example to showcase how asymmetries can affect open-loop and closed-loop models.

Consider the scheduling process for the next hour of a power system containing a single power plant with a capacity of 4 MW (hence, capable of generating 4 MWh in an hour) and a generation cost of 10 \$/MWh. Consider a penalty of 100 \$/MWh for scheduling the plant below the realized demand and 0 for scheduling above the realized demand. Now, consider a demand for a given hour with the following mass distribution: 0 MWh with a probability of 0.5 and 2 MWh with a probability of 0.5. We consider there are no reserves for the sake of simplicity. Also, suppose there are no external variables, and consider the forecast model: $\Psi(\theta) = \theta$ to obtain a demand forecast $\hat{D}$, i.e., $\hat{D} = \theta$. The LS solution for the demand forecast is $\hat{D} = 1$ as it is the minimizer of: $\min_\theta 0.5(0-\theta)^2 + 0.5(2-\theta)^2$.

Consider the following optimization of the planning function, G_p:

$$g^* \in \arg\min_{g,\delta^{LS},\delta^{SP}} \; 10g + 100\delta^{LS} + 0\delta^{SP} \tag{3-6}$$

$$s.t.\; g + \delta^{LS} - \delta^{SP} = \hat{D} \tag{3-7}$$

$$0 \le g \le 4, \quad 0 \le \delta^{LS}, \quad 0 \le \delta^{SP} \tag{3-8}$$

where g is the generation being scheduled for the related hour, g^* is the schedule decision, δ^{LS} is the value of load shed (missing energy), and δ^{SP} is the load being spilled (excess energy). The optimization of the assessment

(or implementation) function, G_a, is given by:

$$\min_{\delta^{LS},\delta^{SP}} \; 10g^* + 100\delta^{LS} + 0\delta^{SP} \tag{3-9}$$

$$s.t.\ \delta^{LS} - \delta^{SP} = D - g^* \tag{3-10}$$

$$0 \leq \delta^{LS}, \quad 0 \leq \delta^{SP} \tag{3-11}$$

where g^* is fixed and D is the actual demand that is realized during the implementation.

Considering the LS forecast, the operator will run its dispatch *plannning* model and say that the generator should be set to $g^* = 1$ (MWh), at the cost of $1 \times 10 = 10$ \$. Later, in the assessment or implementation step, 50% of the time, the total cost will be just 10 \$, as the system was over-prepared, and 50% of the time, the cost will be $1 \times 10 + 1 \times 100 = 110$ \$, thereby leading to an average cost of $0.5 \times 10 + 0.5 \times 110 = 60$ \$.

Now consider a biased forecast of $\hat{D} = 1.1$ (MWh). The generator should be set to $g^* = 1.1$, at cost of $1.1 \times 10 = 11$ \$. Hence, 50% of the time, the total cost will be just 11, as the system was over-prepared, and 50% of the time, the cost will be $1.1 \times 10 + 0.9 \times 100 = 101$ \$, thus leading to an average cost of $0.5 \times 11 + 0.5 \times 101 = 56$ \$. Hence, on average, a biased forecast did perform better than the LS forecast for the problem in question. Notwithstanding, based on our methodology, we can go even further and obtain the best forecast for this application. For this simple case, we can analytically compute the application-driven cost function given a forecast rule, $\hat{D} = \theta$. For simplicity, we only consider $\hat{D} \in [0, 2]$, which contains the extremes of the support set. Thus, in this case, $g^* = \hat{D}$. So, if $\theta \leq 2$, then the cost is given by $10\theta + 0.5 \times 100(2 - \theta)$. Else, if $\theta \geq 2$, then the average cost is 10θ and the minimizer is clearly $\hat{D} = \theta = 2$, which is clearly different from the one obtained with least squares ($\hat{D} = 1$).

One of the main reasons for the phenomenon described above is that the operator is constrained to a specific decision rule (planning method) that has to consider a given forecast in a predetermined way. While the above example is indeed very stylized to allow for a simple exposition, it carries the fundamental idea that is applied in most power systems: 1) planning models are predetermined and must consider demand forecasts; 2) costs are asymmetric as the cost of not delivering energy is usually more expensive than over-scheduling; 3) some operators have noticed the relevance of the asymmetry and, as described in the introduction, have been implementing *ad hoc* procedures to introduce out-of-market bias on forecasts.

3.5 Convergence Results

In this section, we discuss some conditions for the convergence of estimators obtained with application-driven joint prediction and optimization. Again, our goal is to obtain the best possible forecast $\hat{y}_t$, but this is completely defined by the parameters θ since x_t is known. Let S_T be the set of optimal solutions of (3-1)-(3-5), so that $\theta_T \in S_T$. We will show that any sequence of θ_T, each in the set S_T, converges to the solution set of the actual expected value formulation of the problem (as opposed to the previously presented sampled version). We will start by describing some assumptions, then we will state and prove the main theorem.

Assumption 1 *There is a unique solution z_t of optimization problem* (3-3) *for all possible values of $\hat{y}_t$.*

In other words, the problem is always feasible and the solution set is a singleton. This is not as restrictive as it seems. The feasibility requirement is similar to the classical assumption of complete recourse in stochastic programming. The above uniqueness requirement is equivalent to the absence of dual degeneracy in a linear program [183]. In this case, the problem in question is dual-degenerate, but it is possible to eliminate this degeneracy by perturbing the objective function—in our case, the vectors c_p and q_p—with small numbers that do not depend on the right-hand-side (RHS) of the problem. Thus, the same perturbation is valid for all possible $\hat{y}_t$ [184]. Another possibility would be resorting to some lexicographic simplex method [185].In this setting, we can define the set-valued function:

$$\zeta(y) := \text{argmin}_{z \in Z}\ G_p(z, y) \tag{3-12}$$

From [186] we know that if $\zeta(y)$ is a compact set for all y then it is a continuous set-valued function. Moreover, since $\zeta(y)$ is a singleton for all possible values of y, then we treat it as a vector-valued function that is continuous and piece-wise affine [183].

Assumption 2 *The feasibility set Z that appears in* (3-3) *is a non-empty and bounded polyhedron.*

Assumption 2 is reasonable since this is the set of implementable solutions of the decision-maker, typically representing physical quantities.

Assumption 3 *The feasibility set of the dual of the problem that defines $Q_a(z, y)$ in* (3-5) *is non-empty and bounded.*

Note that the set from Assumption 3 does not depend on z and y, since they appear in the RHS of the primal problem. Again, this assumption is akin to a relatively complete recourse assumption applied to the problem defining the outer-level function.

We state now our main convergence result.

Theorem 1 *Consider the process given by* (3-1)–(3-5) *and any possible output* $\theta_T \in S_T$*, for each* T*. Suppose that (i) Assumptions 1, 2 and 3 hold, (ii) the forecasting function* $\Psi(\cdot,\cdot)$ *is continuous in both arguments, (iii) the data process* $(X_1, Y_1), \ldots, (X_T, Y_T)$ *is independent and identically distributed (with* (X, Y) *denoting a generic element), (iv) the random variable* Y *is integrable, and (v) the set* Θ *is compact. Then, with probability 1,*

$$\lim_{T\to\infty} d(\theta_T, S^*) = 0, \tag{3-13}$$

where d *is the Euclidean distance from a point to a set and* S^* *is defined as*

$$S^* = \operatorname{argmin}_{\theta\in\Theta} \left[G_a\Big(\zeta(\Psi(\theta, X)), Y\Big)\right], \tag{3-14}$$

with $\zeta(\cdot)$ *defined in* (3-12).

Proof: First, notice that $G_i(z, Y)$, $i \in \{a, p\}$, is continuous with respect to its arguments as it is a sum of a linear function and the optimal value of a parametric program [187]. Recall that ζ is a continuous vector-valued function because of Assumption 1. Hence, $G_a(\zeta(\Psi(\theta, X)), Y)$ is a real-valued continuous function. Next, we show that $G_a(\zeta(\Psi(\theta, X)), Y)$ is integrable. Indeed, since Z is bounded (Assumption 2), it follows that $\zeta(y)$ is bounded for all x by a constant, say K_1, so that $\|\zeta(y)\| \leq K_1$. By duality, $Q_i(z, y) = \max_{\pi}\{(b_i - H_i z + F_i y)^\top \pi \mid W_i^\top \pi = q_i, \pi \geq 0\}$, but by Assumption 3 the dual variables of $Q_a(z, y)$ are bounded by a constant, say K_2, so $\|\pi\| \leq K_2$. Thus, by a sequence of applications of Cauchy-Schwarz and triangle inequalities, we have that

$$\begin{aligned}
\Big|G_a(\zeta(\Psi(\theta, X)), Y)\Big| &\leq \Big|c_a^\top \zeta(\Psi(\theta, X)) + Q_a(\zeta(\Psi(\theta, X)), Y)\Big| \\
&\leq \Big\|c_a\Big\|\Big\|\zeta(\Psi(\theta, X))\Big\| + \\
&\qquad \Big\|b_a - H_a\zeta(\Psi(\theta, X)) + F_a Y\Big\| \max_{W_i^\top \pi = q_a, \pi \geq 0} \Big\|\pi\Big\| \\
&\leq K_1\Big\|c_a\Big\| + K_2\left(\Big\|b_a\Big\| + \Big\|H_a\zeta(\Psi(\theta, X))\Big\| + \Big\|F_a Y\Big\|\right) \\
&\leq K_1\Big\|c_a\Big\| + K_2\left(\Big\|b_a\Big\| + \Big\|H_a\Big\|K_1 + \Big\|F_a\Big\|\Big\|Y\Big\|\right).
\end{aligned}$$

Hence, since Y is integrable (condition (iv) of the Theorem), we have that $G_a(\zeta(\Psi(\theta, X)), Y)$ is integrable.

It follows that the conditions of Theorem 7.53 in [153] are satisfied and we conclude that: (i) the function $\varphi(\theta) := \mathbb{E}[G_a(\zeta(\Psi(\theta, X)), Y)]$ is finite valued and continuous in θ, (ii) by the Strong Law of Large Numbers, for any $\theta \in \Theta$ we have

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} G_a\Big(\zeta(\Psi(\theta, X_t)), Y_t\Big) = \Big[G_a\Big(\zeta(\Psi(\theta, X)), Y\Big)\Big] \quad \text{w.p.1,} \tag{3-15}$$

and (iii) the convergence in (3-15) is *uniform* in θ. Thus, by Theorem 5.3 in [153], since the set Θ is compact we have that the minimizers (over Θ) of the expression inside the limit on the left-hand side of (3-15)—i.e., θ_T—converge to the minimizers of the expression on the right-hand side in the sense of (3-13)–(3-14). *Q.E.D.*

Remark 1 *Assumption 3 can be replaced by assuming a compact support of Y; in this case, $G_a(z, y)$ is a continuous function, where both arguments are defined on compact sets, hence it attains a maximum and is trivially integrable.*

While Theorem 1 provides the desired convergence result, condition (iii) of the theorem clearly precludes modeling the situation where the features x_t include (functions of) previous observations $y_{t-1}, \ldots, y_{t-k}$. We now extend that result to the case where the features x_t include only lagged observations of $\{y_t\}$. To do so, suppose that the data process generating $\{Y_t\}_{t=1}^{\infty}$ is a *stationary ergodic* time series. Stationarity means that the joint distribution of $(Y_1, Y_2, \ldots, Y_k)$ is the same as the joint distribution of $(Y_{t+1}, Y_{t+2}, \ldots, Y_{t+k})$ for all positive integers t and k. It is a standard assumption in the analysis of time series; see, e.g., [188] (note that [188] actually call this notion *strict stationarity*, but elsewhere in the literature it is called just stationarity; see, e.g., [189] or [190]). On the other hand, an ergodic time series is—roughly speaking—one that exhibits a form of "average asymptotic independence"; a precise definition can be found, for instance, in [189]. Statistical tests for stationarity and ergodicity of Markovian processes have been developed by [191].

For reference, we state now a lemma that provides a more general result than Theorem 7.53 in [153]:

Lemma 1 *Theorem 7.53 in [153] is still valid if the i.i.d. assumption is replaced with a weaker assumption that the samples form a stationary ergodic process.*

Proof: Any measurable function of a stationary ergodic process is also a stationary ergodic process [190]. Moreover, if a process $\{W_t\}_{t=1}^{\infty}$ is stationary and ergodic, then the classical ergodic theorem (see, e.g., [190]) ensures that

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} W_t \;=\; [W_1] \quad \text{w.p.1.}$$

A closer look at the proof of Theorem 7.53 in [153] shows that the i.i.d. assumption is used only to invoke the Strong Law of Large Numbers, which as shown above, can be replaced by the ergodic theorem in the more general case. *Q.E.D.*

We can now state a more general version of Theorem 1:

Theorem 2 *Theorem 1 is still valid if the assumption that the data process* $(X_1, Y_1), \ldots, (X_T, Y_T)$ *is independent and identically distributed is replaced with the following assumption:* X_t *is defined as a (measurable) function of* $Y_1, \ldots, Y_t$*, and the data process generating* $\{Y_t\}_{t=1}^{\infty}$ *is a stationary ergodic time series.*

Proof: Fix $\theta \in \Theta$. Consider the function Φ defined as

$$\Phi(Y_1, \ldots, Y_t) \;:=\; G_a\Big(\zeta(\Psi(\theta, X_t)), Y_t\Big)$$

and process $\{W_t\}_{t=1}^{\infty}$ defined as $W_t := \Phi(Y_1, \ldots, Y_t)$. Under the assumption that $\{Y_t\}_{t=1}^{\infty}$ is a stationary ergodic time series, it follows that $\{W_t\}_{t=1}^{\infty}$ is stationary and ergodic, since Φ is measurable. Lemma 1 shows that (3-15) holds in this case and, therefore, the proof follows the same steps as those of the proof of Theorem 1. *Q.E.D.*

Corollary 1 *Theorems 1 and 2 imply that if* G_a *is an assessment function describing the ultimate goal of a given practitioner—e.g., the expected cost incurred when using the forecast function* $\Psi(\theta, X)$ *within a given application—then, under the conditions of Theorem 1 or Theorem 2, any convergent subsequence of the process* $\{\Theta_T\}_{T=1}^{\infty}$ *generated by the estimation process* (3-1)–(3-5) *converges to the (not necessarily unique) best forecast model,* $\Psi(\theta^*, X)$*, in terms of the related application. That is,* $\Big[G_a\Big(\zeta(\Psi(\theta^*, X)), Y\Big)\Big] \leq \Big[G_a\Big(\zeta(\Psi(\theta, X)), Y\Big)\Big]$ $\forall \theta \in \Theta$*. This ensures that our model performs asymptotically better than the classical offline approaches.*

Proof: Because of Theorem 1 or Theorem 2, we know that the distance between θ_T and the set S^* of minimizers of the underlying problem (given by (3-14))

converges to zero as $T \to \infty$. Thus, any convergent subsequence of the process $\{\Theta_T\}_{T=1}^{\infty}$ converges to some minimizer $\theta^* \in S^*$. Moreover, since $\{\Theta_T\}_{T=1}^{\infty} \subseteq \Theta$ and Θ is assumed to be compact, it follows that there exists at least one convergent subsequence. Meanwhile, any other possible choice of θ, obtained by any other estimation method, is merely a feasible solution for the underlying problem, hence, cannot be strictly better than θ^*. Q.E.D.

Corollary 1 highlights that application-driven learning is the best one can do for a fixed triplet of assessment, planning and forecasting functions when the ultimate goal is only minimizing the assessment cost. Although it is possible that other methods lead to the same optimal objective cost, they cannot be better. For other goals, such as minimizing the squared error of forecasts, other methods, such as least squares, will be better since they are inherently aligned with such other goals.

3.6 Solution Methodology

In this section, we describe solution methods to estimate the forecasting model within the proposed application-driven closed-loop framework described in (3-1)–(3-5). First, we present an exact method based on an equivalent single-level mixed integer linear programming (MILP) reformulation of the bilevel optimization problem (3-1)–(3-5). This method uses MILP-based linearization techniques to address the Karush Kuhn Tucker (KKT) optimality conditions of the second level and thereby guarantee the global optimality of the solution in exchange for limited scalability. Second, we describe how to use zero-order methods [192] that do not require gradients to develop an efficient and scalable heuristic method to achieve high-quality solutions to larger instances. These methods will leverage existing optimization solvers, their current implementations and features.

3.6.1 MILP-based exact method

Our first approach consists of solving the bilevel problem (3-1)–(3-3) with standard techniques based on the KKT conditions of the second-level problem [80]. Thus, the resulting single-level nonlinear equivalent formulation can be reformulated as a MILP and solved by standard commercial solvers. The conversion between the KKT form to the MIP form can be done by numerous techniques [81, 128, 80], all of which have pros and cons. These techniques are implemented and automatically selected by the open-source package BilevelJuMP.jl [1]. This new package was conceived to allow users to

formulate bilevel problems in JuMP [38] and solve them with multiple off-the-shelf optimization solvers.

For the sake of completeness, we write the single-level nonlinear reformulation of the bilevel problem (3-1)–(3-5) in (3-16)–(3-20). For simplicity, in this model, we assume that $Z = \{z|Ax \geq h\}$ and that Θ is polyhedral.

$$\min_{\theta\in\Theta,\hat{y}_t,z_t^*,u_t,\pi_t} \quad \frac{1}{T}\sum_{t\in\mathbb{T}}\left[c_a^\top z_t^* + Q_a(z_t^*, y_t)\right] \tag{3-16}$$

$$s.t. \quad \forall t \in \mathbb{T}:$$

$$\hat{y}_t = \Psi(\theta, x_t) \tag{3-17}$$

$$W_p y_t + H_p z_t^* \geq b_p + F_p \hat{y}_t \;;\quad A z_t^* \geq h \tag{3-18}$$

$$W_p{}^\top \pi_t = q_p \;;\quad H_p^\top \pi_t + A^\top \mu_t = c_p \;;\quad \pi_t, \mu_t \geq 0 \tag{3-19}$$

$$\pi_t \perp W_p u_t + H_p z_t^* - b_p - F_p \hat{y}_t \;;\quad \mu_t \perp A z_t^* - h \tag{3-20}$$

Equations (3-16) and (3-17) are the same as (3-1) and (3-2). (3-3) was replaced by (3-18)–(3-20). (3-18) are the primal feasibility constraint, (3-19) are the dual feasibility constraints, and (3-20) represents the complementarity constraints.

3.6.2 Scalable heuristic method

The proposed class of methods will make extensive use of the way of thinking described in Figure 3.1. In other words, the core algorithm decomposes the problem as follows: We call this method a pseudo-algorithm because a few

Algorithm 1: Pseudo algorithm

```
Result: Optimized θ
Initialize θ ;
while Not converged do
    Update θ;
    for t ∈ 𝕋 do
        Forecast: ŷ_t ← Ψ(θ, x_t);
        Plan Policy: z_t^* ← arg min_{z∈Z} G_p(z, ŷ_t);
        Cost Assessment: cost_t ← G_a(z_t^*, y_t)
    end
    Compute cost: cost(θ) ← Σ_{t∈𝕋}(cost_t)
end
```

steps are not well specified, namely *Initialization*, *Update*, and *Convergence* check, allowing for a wide range of possible specifications. *Initialization* can be as simple as θ receiving a vector of zeros, which might not be good if the actual algorithm is a local search. One alternative that will be applied in the case study section is the usage of traditional models as starting points, for

instance, the ordinary least squares. In the case study, we will initialize the algorithm with the LS estimate, this guarantees that the algorithm will return at most the same cost as the open-loop framework in the training sample. There are many possibilities for the *convergence* test. For instance, iteration limit, time, the variation of the objective function value, and other algorithm-specific tests. Finally, the *update* step depends on the selected concrete algorithm that is ultimately minimizing the non-trivial $cost(\theta)$ function.

We will focus on a derivative-free local search algorithm named Nelder-Mead [192]. Notwithstanding, it is relevant to highlight the generality of the proposed pseudo-algorithm. For instance, gradient-based algorithms could also be developed based on numerical differentiation and automatic differentiation [185]. In this context, gradient calculation would enable the usage of Gradient Descent and BFGS-like algorithms [185].

The main features of the above-proposed pseudo-algorithm are: 1) it is suitable for parallel computing (the loop in the sample $\mathbb{T}$ is intrinsically decoupled); 2) each step is based on a deterministic LP defining the second-level variables in (3-3), suitable for off-the-shelf commercial solvers that find globally optimal solutions in polynomial time; 3) each inner step can significantly benefit from warm-start processes developed in linear programming solvers (e.g., the dual simplex warm-start is extremely powerful, and many times only a handful of iterations will be needed in comparison to possibly thousands of iterations if there were no warm-start, cf. [185]). It is worth emphasizing that the aforementioned feature 2) allows for an exact (always optimal) description of the second-level problem. In our approach, we keep the second level exact and face the challenge of optimizing a nonlinear problem on the upper level. In contrast, [152] choose to relax the complementarity constraints and deal with an NLP-based reformulation lacking the benefits of the above-mentioned features 1), 2) and 3) of the pseudo-algorithm. As will be illustrated in the case study, our choice is supported by empirical evidence about the shape of the nonlinear function faced in the objective function. Additionally, it is usual in more complex estimation processes (like maximum likelihood-based methods) to rely on nonlinear optimization methods to select the best parameters [193]. Moreover, although not convex, as explored in our case study, the objective function seems to be quasi-convex (from the graph inspection) which facilitates the search within the parameters domain. Finally, note that this heuristic approach allows for a wider set of forecast models, such as NN and other machine learning models, as it only requires that a forecast can be pointwise obtained and its performance evaluated by the cost function for a given trial solution (parameters).

One caveat is that variations on θ can lead to possibly infeasible results for the *Policy Planning* and *Cost Assessment* optimization problems. Consequently, we require complete recourse for such problems. In cases where this property does not hold, it is always possible to add artificial (slack) variables with high penalty costs in the objective function to keep the problem feasible. In the energy and reserve dispatch problem, this requirement is addressed by imbalance variables (load and renewable curtailment decisions).

3.7 Application-Driven Load Forecasting and Reserve Sizing

In this work, we focus on the energy and reserve scheduling problem of power systems [53, 194]. In this problem, we aim to obtain the best joint conditional point-forecast for the vector of nodal demands, $\hat{D}_t$, and vectors of up and down zonal or nodal reserve requirements, $\hat{R}_t^{(up)}$ and $\hat{R}_t^{(dn)}$. While the forecast vector of nodal loads represents, e.g., the next hour operating point target that system operators and agents should comply with, up- and down-reserve requirements represent a forecast of the system's resource availability (or security margins), defined per zone or node, allowing the system to withstand load deviations. Note that we can think of loads as a general net load that corresponds to load minus non-dispatchable (e.g., renewable) generation.

The inputs of the problem are: vectors of historical data of dependent and explanatory variables, $\{y_t, x_t\}_{t\in\mathbb{T}}$, including lags of demand, $D_{t-1}, ..., D_{t-k}$, and possibly other exogenous covariates such as climate and weather indices (or forecasts), dummy variables, and non-linear machine-learning-based forecasts for the dependent variables; vectors of data associated with generating units; maximum generation capacity, G, dispatch costs or offers, c; maximum up- and down-reserve capacity, $\bar{r}^{(up)}$ and $\bar{r}^{(dn)}$; up- and down-reserves costs, $p^{(up)}$ and $p^{(dn)}$; load-shed and spillage penalty costs, λ^{LS} and λ^{SP}; network data comprising the vector of transmission line capacities F; and network sensitivity matrix, B, describing the network topology and physical laws of electric circuits. Additionally, it is important to mention that the input data describing the system characteristics can be provided under two perspectives: 1) under the perspective of the actual *ex-post* (or assessed/implemented) operation, i.e., based on the observed demand data and most accurate system's description for optimizing the function G_a defined in (3-1); and 2) under the *ex-ante* planning perspective, G_p, which is accounted for in (3-3) based on observed features, such as previous information, and system operator's description of the system considered in the dispatch tool. While the former has already been listed at the beginning of this paragraph, the latter uses the same symbols but with a tilde

above, i.e., $\tilde{c}, \tilde{p}^{(up)}, \tilde{p}^{(dn)}, \tilde{G}, \tilde{B}$, etc. For a simple matrix representation of the problem, we define e to be a vector with one in all entries and an appropriate dimension. M is an incidence matrix with buses in rows and generators in columns that is one when the generator lies in that bus and zero otherwise. Similarly, N is an incidence matrix with generators in columns and reserve zones in rows, which is one if the generator lies in that area. Thus, we study the following particularization of the closed-loop framework proposed in (3-1)–(3-5):

$$\min_{\substack{\theta_D,\theta_{R^{up}},\theta_{R^{dn}},\\ \hat{D}_t,\hat{R}_t^{(up)},\hat{R}_t^{(dn)},g_t^*,\delta_t^{LS},\delta_t^{SP}\\ g_t^*,r_t^{(up)*},r_t^{(dn)*}}} \frac{1}{T}\sum_{t\in\mathbb{T}}\Big[c^\top g_t + p^{(up)\top}\hat{r}_t^{(up)*} + p^{(dn)\top}\hat{r}_t^{(dn)*} + \lambda^{LS}\delta_t^{LS} + \lambda^{SP}\delta_t^{SP}\Big] \tag{3-21}$$

$$s.t. \quad \forall t \in \mathbb{T}:$$

$$\hat{D}_t = \Psi_D(\theta_D, x_t) \tag{3-22}$$

$$\hat{R}_t^{(up)} = \Psi_{R^{(up)}}(\theta_{R^{(up)}}, x_t) \tag{3-23}$$

$$\hat{R}_t^{(dn)} = \Psi_{R^{(dn)}}(\theta_{R^{(dn)}}, x_t) \tag{3-24}$$

$$e^\top(Mg_t - \delta_t^{SP}) = e^\top(D_t - \delta_t^{LS}) \tag{3-25}$$

$$-F \le B(Mg_t + \delta_t^{LS} - D_t - \delta_t^{SP}) \le F \tag{3-26}$$

$$g_t^* - r_t^{(dn)*} \le g_t \le g_t^* + r_t^{(up)*} \tag{3-27}$$

$$\delta_t^{LS}, \delta_t^{SP}, \hat{R}_t^{(up)}, \hat{R}_t^{(dn)}, g_t \ge 0 \tag{3-28}$$

$$\Big(g_t^*, r_t^{(up)*}, r_t^{(dn)*}\Big) \in \underset{\substack{\hat{g}_t,\hat{\delta}_t^{LS},\hat{\delta}_t^{SP},\\ \hat{r_t}^{(up)},\hat{r_t}^{(dn)}}}{\arg\min} \Big[\tilde{c}^\top\hat{g}_t + \tilde{p}^{(up)\top}\hat{r_t}^{(up)} + \tilde{p}^{(dn)\top}\hat{r_t}^{(dn)} + \tilde{\lambda}^{LS}\hat{\delta}_t^{LS} + \tilde{\lambda}^{SP}\hat{\delta}_t^{SP}\Big] \tag{3-29}$$

$$s.t. \quad e^\top(M\hat{g}_t - \hat{\delta}_t^{SP}) = e^\top(\hat{D}_t - \hat{\delta}_t^{LS}) \tag{3-30}$$

$$-\tilde{F} \le \tilde{B}(M\hat{g}_t + \hat{\delta}_t^{LS} - \hat{D}_t - \hat{\delta}_t^{SP}) \le \tilde{F} \tag{3-31}$$

$$N\hat{r_t}^{(up)} = \hat{R}_t^{(up)} \tag{3-32}$$

$$N\hat{r_t}^{(dn)} = \hat{R}_t^{(dn)} \tag{3-33}$$

$$\hat{g}_t + \hat{r_t}^{(up)} \le \tilde{G} \tag{3-34}$$

$$\hat{g}_t - \hat{r_t}^{(dn)} \ge 0 \tag{3-35}$$

$$\hat{r_t}^{(up)} \le \bar{r}^{(up)} \tag{3-36}$$

$$\hat{r_t}^{(dn)} \le \bar{r}^{(dn)} \tag{3-37}$$

$$\hat{g}_t, \hat{r_t}^{(up)}, \hat{r_t}^{(dn)}, \hat{\delta}_t^{LS}, \hat{\delta}_t^{SP} \ge 0. \tag{3-38}$$

In (3-21)–(3-38), the objective function of the upper level problem (3-21)

comprises the sum of the actual operating cost, the cost of scheduled reserves, and the implemented load-shed and renewable spillage costs for all periods within the dataset, i.e., for $t \in \mathbb{T}$. In the upper level, constraints (3-22)–(3-24) define the forecast model. Note that all periods are coupled by the vector of parameters $\theta = [\theta_D^\top, \theta_{R^{(up)}}^\top, \theta_{R^{(dn)}}^\top]^\top$, which do not depend on t. These parameters define the forecast model that will be applied to each t for demand, as per $\hat{D}_t$ in (3-22), for up reserve requirements, as per $\hat{R}_t^{(up)}$ in (3-23), and for down reserve requirements, as per $\hat{R}_t^{(dn)}$ in (3-24). The forecast models are defined by functions Ψ_D, $\Psi_{R^{(up)}}$, and $\Psi_{R^{(dn)}}$ that transform parameters and the historical data on load and reserve requirement forecasts. For the sake of simplicity and didactic purposes, in this work, we assume affine regression models. The reserves are parts of the forecast vector $\hat{y}_t$ because the method optimizes a model for them. However, reserves' historical data need not be in y_t, since, typically, the choice of reserves in each period is not based on past values of reserves.

Constraints (3-25)–(3-28) together with the objective function (3-21) particularize G_a from (3-1). They assess the *ex-post* operating cost (first term, $c_t^T g_t$, and the last two terms, $\lambda_t^{LS}\delta_t^{LS} + \lambda_t^{SP}\delta_t^{SP}$) of the actual dispatch given the *ex-ante* planned generation, g_t^*, and allocated up and down reserves, $r_t^{(up)*}$ and $r_t^{(dn)*}$, defined by the second level (3-29)–(3-38). Constraint (3-25) accounts for the *ex-post* energy balance constraint, where total generation meets total observed load data. The left-hand side of the constraint is the sum of generated energy in all buses, with Mg_t resulting in the nodal generation injection vector (total generation per bus). δ_t^{SP} represents the nodal generation spilled per bus (positive load imbalance decision). The right-hand side of the constraint accounts for the net-nodal load vector (observed net-demand vector D_t). δ_t^{LS} represents the decision vector of nodal load-shed (negative load imbalance decision). Finally, constraint (3-26) limits the flow of energy through each transmission line to pre-defined bounds and (3-27) limits the *ex-post* generation to respect the operation range defined by the *ex-ante* planned generation, g_t^*, and allocated up and down reserves, $r_t^{(up)*}, r_t^{(dn)*}$. Constraint (3-28) ensures the positiveness of slack, generation, and reserve requirement variables.

The planning policy defines variables g_t^*, $r_t^{(up)*}$, and $r_t^{(dn)*}$ under the conditional information available in vector x_t. These variables should respect the optimality of the market's or system operator's *ex-ante* scheduling (or planning policy), as per (3-29), based on the vector of load forecasts, $\hat{D}_t$, and vectors of reserve requirements, $\hat{R}_t^{(up)}$ and $\hat{R}_t^{(dn)}$, for the next period (e.g., hour). Again, it is relevant to note that this planning policy may differ from the actual *ex-post* implemented policy (based on the observed load data D_t),

resulting in the actual operating cost considered in the objective function of the first level (3-21). Within this context, constraints (3-29)–(3-38) detail the second-level problem, which represents the optimization that takes place in the planning phase at each period to define a generation and reserve schedule for the next period. Thus, these constraints are particularizing the general model, G_p in (3-3). In the proposed closed-loop framework, (3-29) is key. It allows us to define, within a problem that seeks the best forecast model aiming to minimize the *ex-post* operation cost, the objective of an *ex-ante* scheduling problem minimizing energy and reserve costs for a conditioned load forecast ($\hat{D}_t$) and reserve requirements ($\hat{R}_t^{(up)}$ and $\hat{R}_t^{(dn)}$). Constraints (3-30) and (3-31) are similar to (3-25) and (3-26). Expressions (3-32) and (3-33) ensure that the total reserve requirements, $\hat{R}_t^{(up)}$ and $\hat{R}_t^{(dn)}$, which are considered as parameters for the lower-level problem, must be allocated among generators in the form of up and down reserves, $\hat{r}_t^{(up)}$ and $\hat{r}_t^{(dn)}$, lower-level decision variables. Constraints (3-34) and (3-35) limit the scheduled generation and reserves range (up and down) to generators' physical generation limits. Constraints (3-36) and (3-37) limit the maximum amount of reserves that can be allocated in each generating unit, and (3-38) ensures positiveness of the generation and reserve decision vectors of the second level. We highlight that G_a and G_p are cost functions represented by linear programs. Although their objective functions have the same structure, their above-described constraints are different.

The proposed application-driven framework provides system operators with the flexibility to either jointly optimize the load forecast and the reserve requirements or only one of them. Additionally, this model is powerful because the load forecasts and both the size and location of reserve requirements are defined in the best way possible to maximize the ex-post performance of the operation. While the lower level, (3-29)–(3-38), ensures an *ex-ante* (planning) generation schedule and reserve allocation compatible with the system operator's information level (best schedule given the previous hour conditional forecast for t), the upper level selects the parameters of the forecast model aiming to minimize the average *ex-post* (assessment/implementation) operating cost for a large dataset. In this sense, depending on the network details considered in the assessment part of the model, it also helps in mitigating reserve deliverability issues associated with *ad hoc* procedures used in industry practices.

In the next section, we will compare a few variants of this problem by optimizing all or some of the parameters θ_D, $\theta_{R^{(up)}}$, and $\theta_{R^{(dn)}}$. As described in the contributions section, two novel methods are the one that optimizes all

parameters and the one that gets a fixed θ_D and only optimizes reserves ($\theta_{R^{(up)}}$, and $\theta_{R^{(dn)}}$). The case of fixed reserves and optimized θ_D will also be presented but is not particularly meaningful in practice.

3.8 Case Studies

This section presents case studies to demonstrate the methodology's applicability and how the closed-loop framework can outperform the classic open-loop scheme in multiple variants of the load forecasting and reserve sizing problem defined in Section 3.7. First, we show that the Heuristic method of Section 3.6.2 can achieve close to optimal solutions in a fraction of the time required by the Exact method of Section 3.6.1. Second, we study the estimated parameters' and forecasts' empirical properties and contrast them with the classical least squares (LS) estimators. After that, we briefly explore how the method can estimate a reserves model with features. Moreover, we apply the method to highlight the relationship between load-shed cost and the estimated parameters. Finally, we show that the heuristic algorithm finds good quality local-optimal parameters systematically outperforming the LS open-loop benchmark for instances far larger than those solved in previously reported works tackling closed-loop bilevel frameworks. We used the same Dell Notebook (Intel i7 8th Gen with 4 cores at 1.99Ghz, 16Gb RAM) for all studies except when otherwise noted.

3.8.1 Power systems cases and datasets

We consider multiple power system cases throughout this section. The first is a single bus system defined by us, with 1 zone, 1 load (with a long-term average of 6) and 4 generators (with capacities $5, 5, 2.5, 2.5$ and costs $1, 2, 4, 8$). The other three are typical test systems based on realistic networks used by the power system community, namely, "IEEE 24bus rts" (38 lines, 33 generators, 17 loads, 4 zones), "IEEE 118 bus" (186 lines, 54 generators, 99 loads, 7 zones) and "IEEE 300 bus" (411 lines, 69 generators, 191 loads, 10 zones). The base datasets were obtained from PG-LIB-OPF [195]. The zone definition is standard for the 24 bus case. For the 118 and 300 bus cases, we used the zones defined by [196]. We also considered very large cases, with thousands of buses, by connecting copies of the 300-bus system. Finally, we consider realistic power systems with more than 6000 buses.

Henceforth, we will refer to the test systems by their number of buses. We only considered loads in buses with positive demand in the original files. We

performed some modifications in the case data: all flow limits were set to 75% of their original rate; we modified the demand values of the cases 24, 118 and 300 by the factors: 0.9, 1.3 and 0.9, respectively. These modifications were made to stress the systems. Deficit and generation curtailment costs were defined, respectively, as 8 and 3 times the most expensive generator cost. All generators were allowed to have up to 30% their capacity allocated to reserves, and their reserve allocation costs were set to 30% their nominal costs. We only considered the linear component of the generators' costs in all instances. In all datasets, we used demand values as the long-term average of AR(1) processes for each bus. The AR(1) coefficients were set to 0.9 and the AR(0) coefficients were set so that we get the desired long-term averages. For the sake of simplicity, load profiles were generated independently. The coefficient of variation of all simulated load stochastic processes was 0.4. Negative demands were truncated to zero, although they could represent an excess renewable generation.

3.8.2
Studied models and notation

In most of the following sections, we will consider simple forecast models so that we can detail experiment results clearly. Therefore, unless otherwise mentioned, the model used to forecast loads in each node is the following:

$$\hat{D}_t = \Psi_D(\theta_D, x_t) = \theta_D(0) + \theta_D(1)D_{t-1}, \tag{3-39}$$

For the single bus case, we set the "real", or population, values as $\theta_D(0) = 0.6$ and $\theta_D(1) = 0.9$, resulting in the long-term average $\theta_D(0)/(1 - \theta_D(1)) = 6$, defined in section 3.8.1. Note that such a choice of coefficients ensures that the input process is stationary and ergodic (see, e.g., [190]), thereby satisfying the condition in Theorem 2. Because the stochastic model for loads, (3-39), is homoscedastic, we set the reserve models to AR(0) – i.e., a number that does not depend on previous values of reserves – since it is customary to set the reserves just in terms of variability of loads:

$$\hat{R}_t^{(up)} = \Psi_{R^{(up)}}(\theta_{R^{(up)}}, x_t) = \theta_{R^{(up)}}(0), \tag{3-40}$$

$$\hat{R}_t^{(dn)} = \Psi_{R^{(dn)}}(\theta_{R^{(dn)}}, x_t) = \theta_{R^{(dn)}}(0). \tag{3-41}$$

3.8.3
Exact vs heuristic method comparison

In this first experiment, we aim to compare the exact and the heuristic methods to check the quality of the latter for instances in which the exact

method is capable of reaching global optimal solutions. To that end, we consider the single-bus test system.

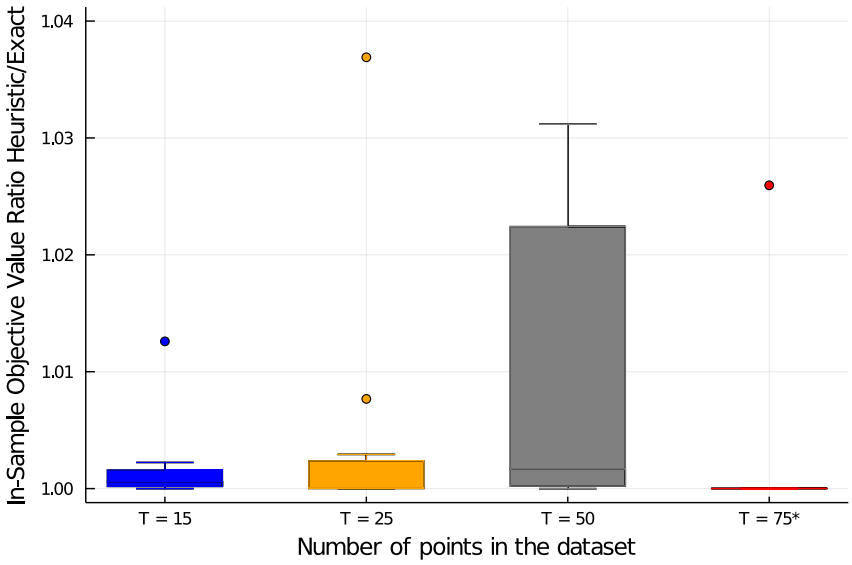
We started by solving 10 instances for each $T \in \{15, 25, 50, 75\}$. All instances solved with the exact method converged within a gap lower than 0.1% using the Gurobi solver or stopped after two hours. The heuristic method was terminated when the objective function presented a decrease lower than 10^{-7} between consecutive iterations. We used a Nelder-Mead implementation found in [197]. To compare the results, we plotted the ratio of objective values in Figure 3.2(a) and the time ratio in Figure 3.2(b). We can observe that the heuristic method achieves high-quality solutions for almost all instances. Although the exact method is competitive for $T \in \{15, 25\}$, the heuristic method is much faster with an average solve time of $4.4s$, for $T = 50$, and $5.9s$, for $T = 75$, compared to $1200s$ and $6670s$ for the exact method. Four instances with $T = 75$ did not converge with the exact method after two hours.

Next, we further analyze the shape of the objective cost, G_a in (3-1), as a function of the parameters, θ, to better understand how good the heuristic solutions can be. Given one dataset with 250 points, we fixed the demand autoregressive parameters to the LS estimation and plot in $\mathbb{R}^3$ the cost as a function of the reserve requirement parameters. Two views of this function are presented in Figure 3.3 (a) and (b). We also plot the cost as a function of the AR(0) and AR(1) coefficients of the demand forecasting model, in this case, the reserve requirement parameters were fixed to the exogenous values of +/-1.96 standard deviations of the LS estimation of load forecast. This is presented in two views in Figure 3.3 (c) and (d).

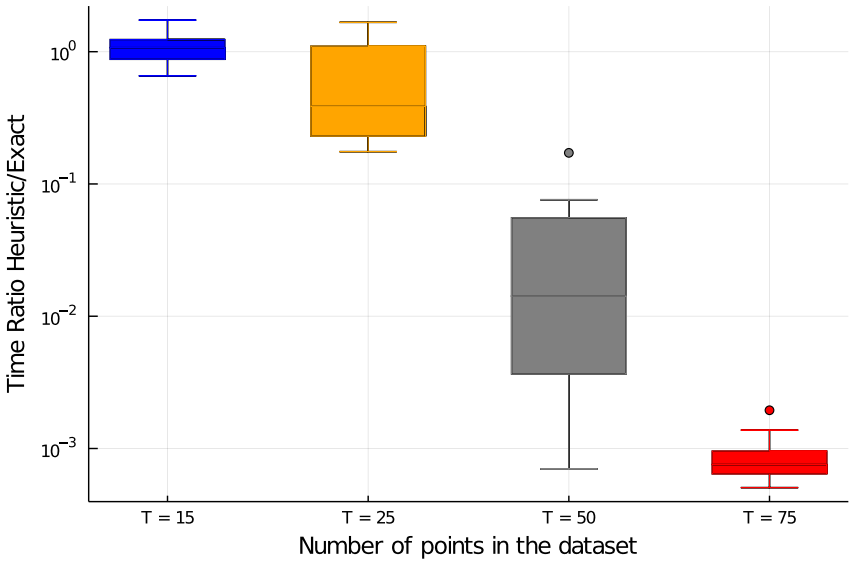
We can note that both functions are reasonably well-behaved and suited for local search algorithms, even though the functions are non-convex. This is a relevant feature supporting the choice for our heuristic approach as previously described at the end of Section 3.6.2. We also note a smoothing effect due to the average in the objective function [153]. Therefore, it is expected to have more well-behaved functions as the sample size grows.

3.8.4 Asymptotic behavior and biased estimation

Now we focus only on the heuristic method to analyze how the estimates behave with respect to the dataset size variation. We will see that they actually converge in our experiments. Moreover, we empirically show through out-of-sample studies that using the closed-loop model is strictly better than the open-loop one provided we have a reasonable dataset size in the training step.



(3.2(a)) In-Sample objective value comparison

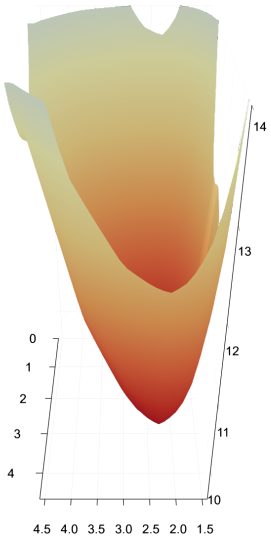


(3.2(b)) Running time comparison

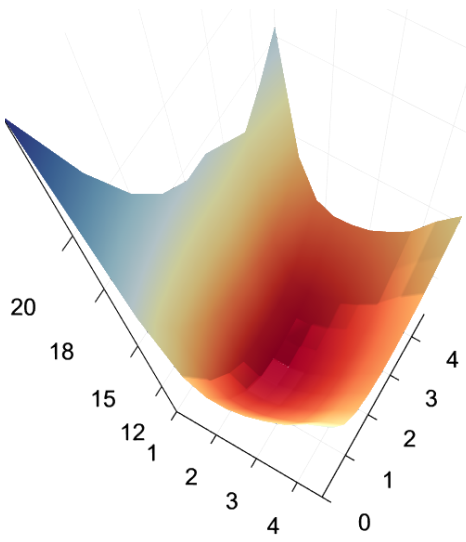
Figure 3.2: (a) Objective of Heuristic method divided by the objective of Exact method for the same datasets. *Four problems not considered for $T = 75$: the exact method found no solution in the given time. (b) Time to solve the same problem (in log scale): Heuristic method divided by Exact method.

It will be possible to see that a method with too many parameters might overfit the model for reduced dataset sizes and not generalize well enough. From now on, we will use the following nomenclature and color code to refer to the different models:

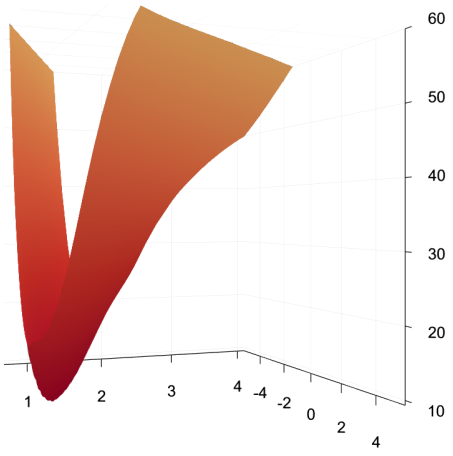
– LS-Ex (red): This is the benchmark model representing the classical open-loop approach. It uses LS to estimate demand and an exogenous reserve requirement.



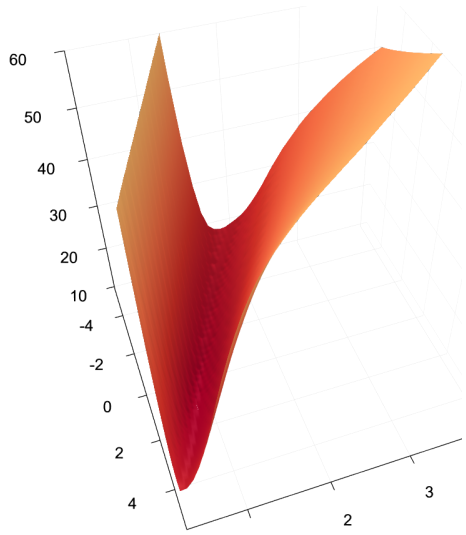
(3.3(a)) Front view: optimizing reserves



(3.3(b)) Top view: optimizing reserves



(3.3(c)) Front view: optimizing load forecast



(3.3(d)) Top view: optimizing load forecast

Figure 3.3: Training cost as a functions of optimization parameters. (a) and (b): fixed load AR coefficients, optimizing reserves $\theta_{R^{(up)}}(0), \theta_{R^{(dn)}}(0)$. (c) and (d): fixed reserves, optimizing load AR coefficients $\theta_D(0), \theta_D(1)$. A dataset with 250 points was used to evaluate the function values in a grid with resolution 0.05 units.

- LS-Opt (blue): This is a partially optimized model, where least squares are used to estimate demand and only reserve requirements are optimized with the application-driven framework.
- Opt-Ex (yellow): This is also a partially optimized model, where demand is optimized, whereas reserve requirements are still exogenously defined. This model is not particularly meaningful in practice. We show it in some

studies for completeness.

– Opt-Opt (green): This is the fully optimized model, where both demand forecast and reserve requirements are jointly optimized.

Both (closed-loop) methods LS-Opt and Opt-Opt are novel contributions proposed in this work, to be contrasted with the benchmark (open-loop) method, LS-Ex. For didactic purposes, in all cases tested in this section, up and down reserve requirements were defined as +/- 1.96 standard deviations, respectively, of the estimated residuals from the LS demand forecast. Note, however, that other exogenous *ad hoc* methods could be used [139].

We empirically compare and analyze the convergence of the four demand and reserve requirement forecast models mentioned above. We varied the dataset size used in the estimation process from 50 to 1000 observations. For each dataset size, we performed 100 trial estimations, with different datasets generated from the same process, to study the convergence. To evaluate the out-of-sample performance of each one of the 100 estimates for each dataset size, we compute the objective function, G_a in (3-1), for a single fixed dataset with $10,000$ new observations (generated with the same underlying process but different from all other data used in the estimation processes). In the following plots, lines represent mean values among the 100 estimated costs with the in-sample or out-of-sample data, and shaded areas represent the respective 10% and 90% quantiles.

The average operation cost for each dataset size is presented in Figure 3.4. The vertical axis shows in-sample costs, while the horizontal axis shows the dataset size used for the estimation procedure. It is possible to see that the method that co-optimizes reserves requirements and demand forecasts finds lower costs than the others. This is expected because this method has more degrees of freedom (it is a relaxed version of the others) on the parameter estimation and this is the objective function being minimized. Also, as expected, the LS plus exogenous reserves requirement model finds higher costs than the others. For the same reason, it does not allow for improvements by the local optimization method as it can be seen as a constrained version of the others. The other two methods are always in between and *Opt-Ex* is always below *LS-Opt*, which shows that demand forecasting might have a larger effect than reserve allocation in this test system.

Figures 3.5 (a) and (b) depict the same costs but in the out-of-sample data. Hence, they measure how well the models generalize to data it has never seen before. We can see that the models allowing more parameters to be endogenously optimized perform much better than models with exogenously defined forecasts. Thus, we see that the application-driven learning framework

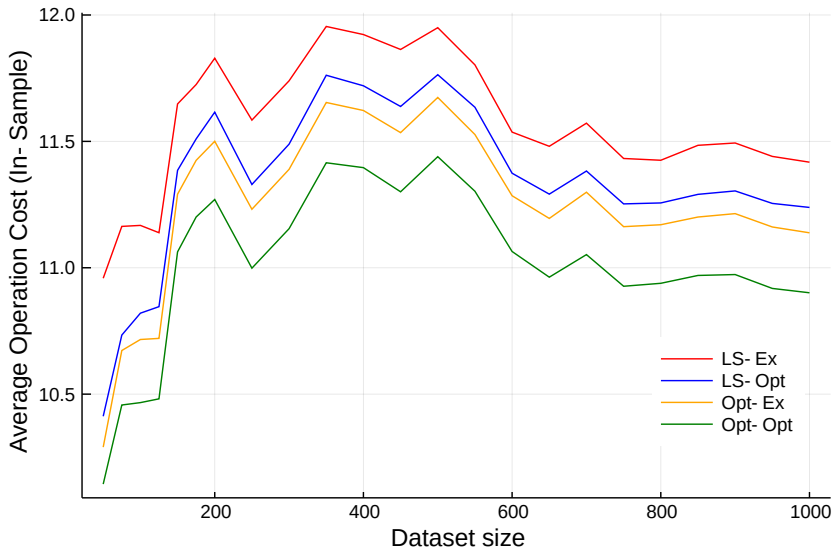
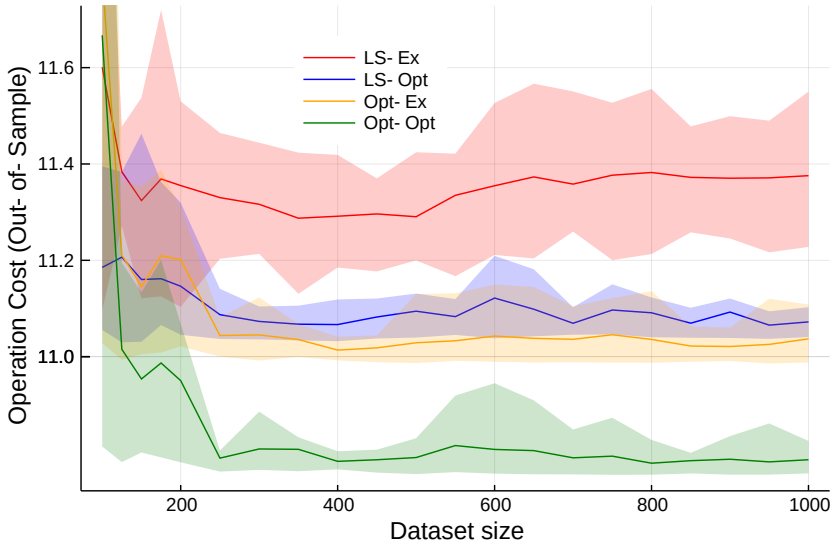


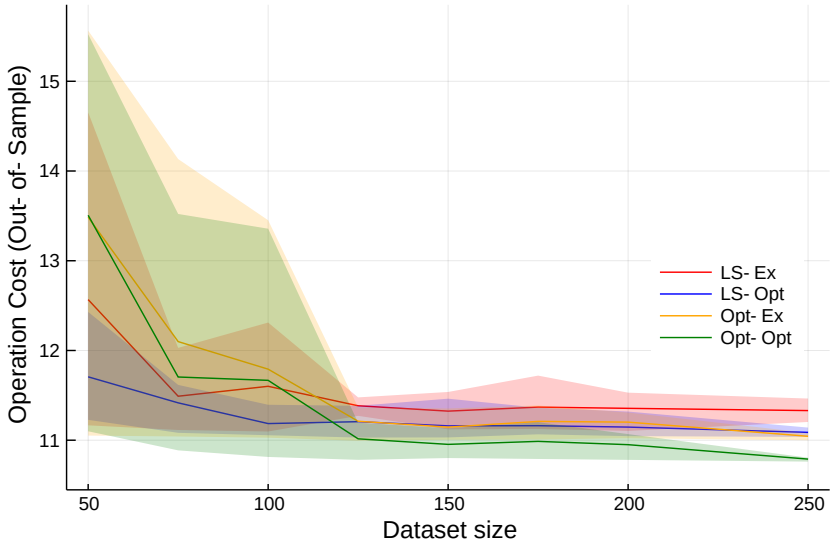
Figure 3.4: Average operation cost in training sample (in-sample) versus dataset size.

works successfully on out-of-sample data when estimated with datasets larger than 150. However, we note that these steady improvements require more data than the classic exogenous models, as shown in Figure 3.5 (b). Between 50 and 120 points, the model with more optimization flexibility, Opt-Opt, exhibits a more significant cost variance. This is due to excessive optimization in a small dataset that led to overfitting and poor generalization. Note that, in this work, we did not consider any regularization procedure to avoid this issue. However, our optimization-based framework is suitable for well-known shrinkage operators [198] that can be readily added in the objective function (3-1).

Figure 3.6 shows how the estimated parameters behave as functions of the estimation dataset size. In Figure 3.6 (a) and (b) we can see that the load model parameters are indeed converging to long-run values. It is also clear to see the bias in those parameters. The constant term is greatly increased while the autoregressive coefficient is slightly reduced. Ultimately this leads to a larger forecast value, which can be interpreted as the application risk adjustment due to the asymmetric imbalance penalization costs (load-shed is much higher than the spillage cost). Thus, the Opt-Opt model will do the best possible to balance these costs, thereby prioritizing the load-shed by increasing the forecast level. The fixed reserves model (Opt-Ex) is less biased because the fixed reserves constrain how much the load model can bias due to the risk of not having enough reserves to address lower demand realizations. Note that the red (LS-Ex) is on top of the purple (LS-Opt) since both use the same LS



(3.5(a)) Starting from dataset size 200.

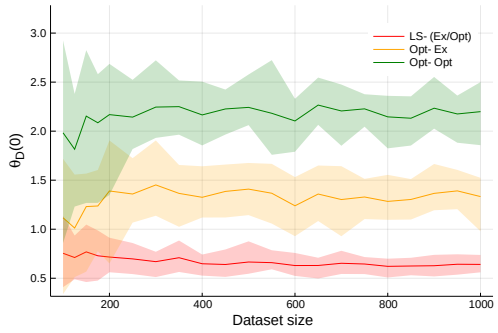


(3.5(b)) Dataset size from 50 to 200.

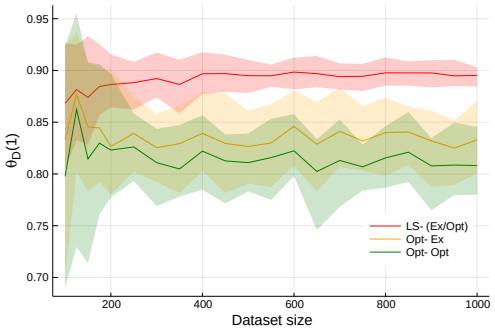
Figure 3.5: Out-of-sample average operation cost versus (in-sample) dataset size. Lines represent the average of the 100 estimation trials. Shaded areas represent the 10% and 90% quantiles. All trials are evaluated on a single out-of-sample dataset with size 10,000 observations.

estimates for demand, which exhibits the lowest variance.

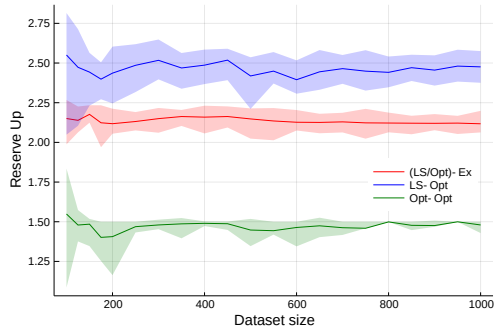
In Figure 3.6 (c) and (d), we see that the Opt-Opt model greatly increases the downward reserve and decreases the upward reserve, both consistent with the change in the demand forecast parameters. Closed-loop estimation of only reserves led to increased up reserves that are the most expensive to violate, while downward reserves are mostly unaffected, the latter might be an artifact of the estimation model that uses the open-loop estimation as a starting point. The Opt-Opt model is limited to 3 because that is the maximum reserve that



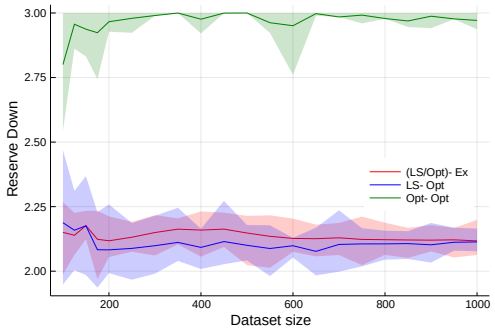
(3.6(a)) $\theta_D(0)$, from (3-39)



(3.6(b)) $\theta_D(1)$, from (3-39)



(3.6(c)) $\theta_{R^{(up)}}(0)$, from (3-40)



(3.6(d)) $\theta_{R^{(dn)}}(0)$, from (3-41)

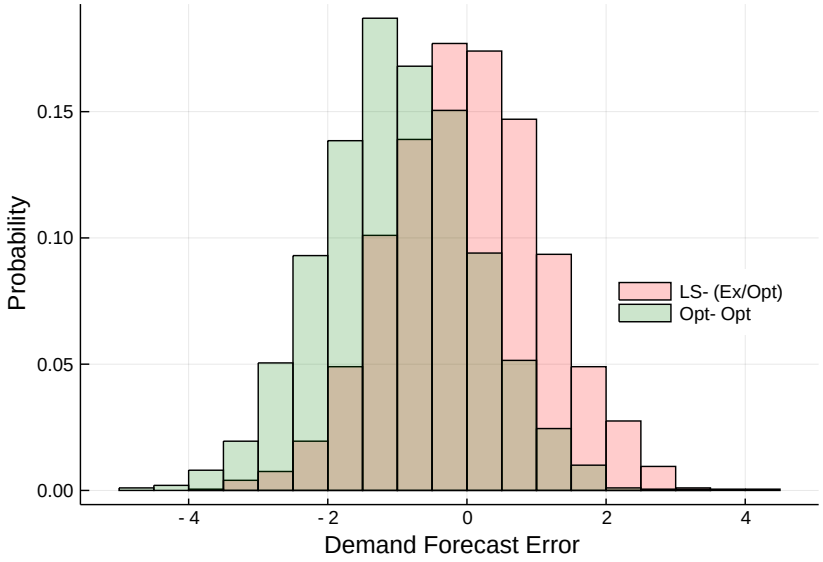
Figure 3.6: Estimated parameters versus dataset size. Lines represent the average of the 100 estimation trials. Shaded areas represent the 10% and 90% quantiles. (a) and (b) Load coefficients, the models LS–Ex and LS–Opt coincide, thereby are presented as LS–(Ex/Opt). (c) and (d) Reserve coefficients The models Ls–Ex and Opt–Ex coincide, thereby are presented as (LS/Opt)–Ex.

can be allocated (30% of the generators' capacity).

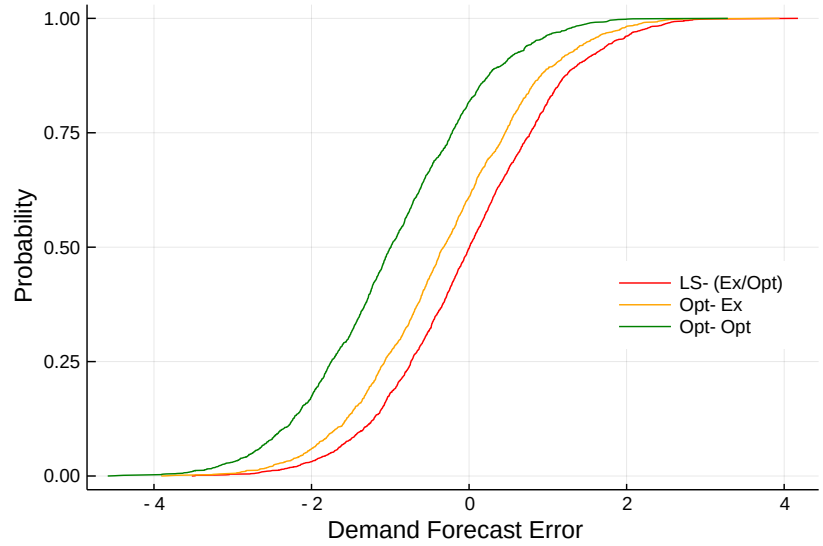
To highlight the bias on load forecast we present, in Figure 3.7 (a), a histogram of deviations: $error := realization - forecast$. Negative values mean that the forecast value was above the realization. The LS estimation leads to an unbiased estimator, seen in the red histogram centered on zero. On the other hand, the forecast from the fully endogenous model is clearly biased, as it consistently forecasts higher values than the realizations. This fact is corroborated by the cumulative distribution functions displayed in Figure 3.7 (b).

3.8.5
A reserves model with features

This experiment aims to show that it is also possible to consider a reserves model with features within the proposed scheme, i.e., conditioned to external information being dynamically revealed to the system operator.



(3.7(a)) Two histograms are shown, the third color is their intersection (LS–Ex and LS–Opt coincide).



(3.7(b)) Accumulated Probability – out of the four models, three are shown here (LS–Ex and LS–Opt coincide).

Figure 3.7: Forecast error (observation – forecast) in a histogram, comparing fully optimized model with least squares estimation. Negative values mean forecast was larger than actual realization.

There are examples of works in the literature that considered the load to be heteroscedastic [172]. Hence, we will study here a simple formulation of demand time series with time-varying variance: We considered an exogenous variable E_t that follows an autoregressive process of order one, for which the noise term, ε_t, has zero mean and variance σ_E^2, (3-42). The variance of the demand process

is the square of E_t, as shown in (3-43).

$$E_t = \phi(0) + \phi(1)E_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim N(0, \sigma_E^2), \tag{3-42}$$

$$D_t = \theta_D(0) + \theta_D(1)D_{t-1} + \epsilon_t, \quad \epsilon_t \sim N(0, E_t^2), \tag{3-43}$$

Formulations in (3-42) and (3-43) were only used to generate synthetic input data for this case study. In this study, we did not modify the demand forecast model (3-39) that is used to specify (3-22). On the other hand, we allowed for features in the reserve sizing, that is, the reserve will vary with external information. This time dependency will be considered through contextual information (features, explanatory variables). As a driver for demand variance changes, E_t is reasonable contextual information for estimating dynamic reserve margins. Thus, we replaced (3-40) and (3-41) that specify (3-23) and (3-24) by the following models for the reserve requirements, (3-44) and (3-45):

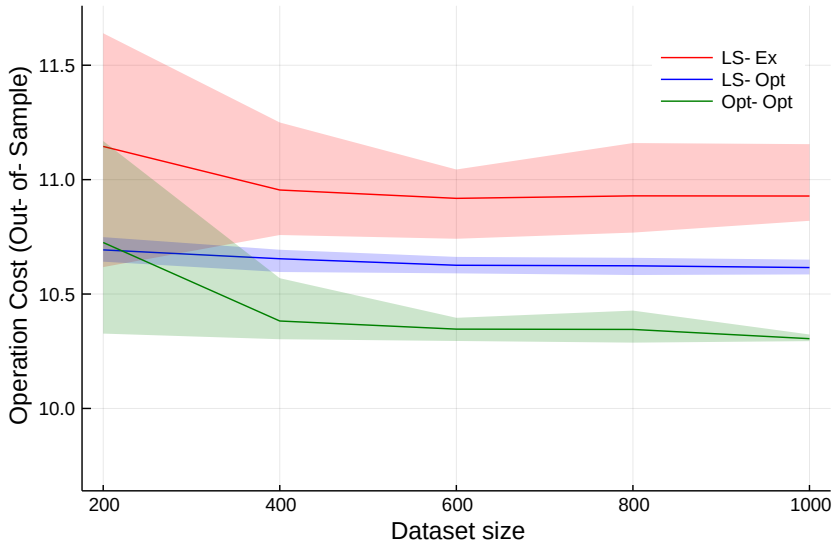
$$\hat{R}_t^{(up)} = \Psi_{R^{(up)}}(\theta_{R^{(up)}}, x_t) = \theta_{R^{(up)}}(0) + \theta_{R^{(up)}}(1)E_t, \tag{3-44}$$

$$\hat{R}_t^{(dn)} = \Psi_{R^{(dn)}}(\theta_{R^{(dn)}}, x_t) = \theta_{R^{(dn)}}(0) + \theta_{R^{(dn)}}(1)E_t. \tag{3-45}$$

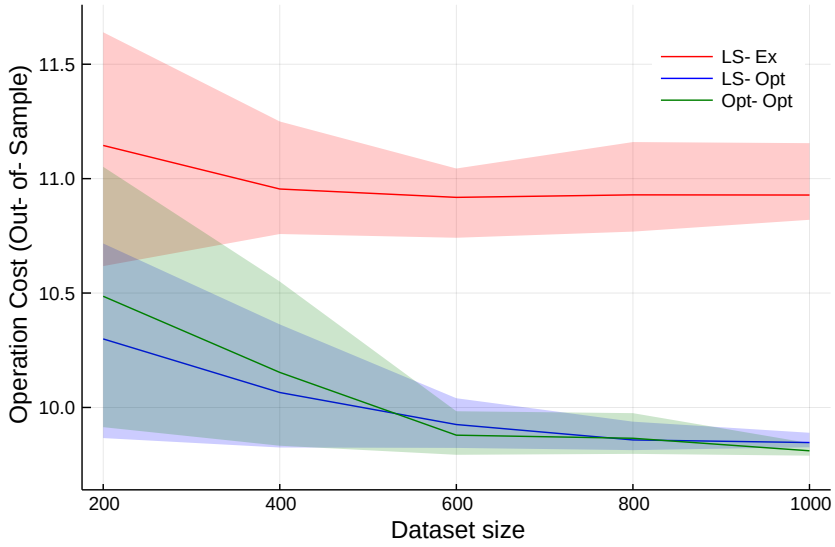
The results of this experiment are depicted in Figure 3.8. We refer to the model from the previous sections as *simple reserves model*, Figure 3.8 (a), and the model defined here as *reserves model with features*, Figure 3.8 (b). The system cost was clearly reduced by considering features in the reserves for both Opt-Opt and LS-Opt models. In the case with features, the Opt-Opt model gain over the LS-Opt is significantly reduced because the reserve sizing method given by (3-44) and (3-45) is able to capture asymmetries by incorporating the variable E_t, therefore this reduces the pressure for an increased bias on demand forecast. Moreover, the demand forecast models do not depend explicitly on E_t for both Opt-Opt and LS-Opt. Hence, in these instances, the reserve sizing, dependent on E_t, is able to optimize up- and down-reserve requirements that incorporate most of the asymmetries on the cost function even with a demand forecast with zero bias. This is a relevant insight that the proposed framework provides.

3.8.6
Reserves and demand forecast as a function of the load-shed cost

This section aims to spotlight the dependency of the estimated parameters on the deficit cost, which is the largest violation penalty in this problem. We varied the deficit cost between 15 and 100 and estimated parameters with one single dataset with $1,000$ observations.



(3.8(a)) Simple Reserves Model



(3.8(b)) Reserves Model with Features

Figure 3.8: Variance driven by exogenous variable. Lines represent the average of the 100 estimation trials. Shaded areas represent the 10% and 90% quantiles.

Figure 3.9 depicts the up and down reserves and steady-state demand $(\theta_D(0)/(1-\theta_D(1))$. We added the solid line in red, *Demand LS*, to represent the steady-state demand and dashed red, *Reserve Ex*, to show the obtained reserve requirement from the residue of the LS estimation model. So, as these two values are exogenously calculated, they do not vary with the load-shed cost. The solid green line, *Demand Opt–Opt*, shows an increasing bias as the load-shed cost grows, corroborating the expected behavior. The dashed green line also shows that the best response is to increase reserve levels as the load-shed cost grows. The dashed blue lines are the reserve margins around the solid

LS-based demand (red line) and they are also clearly affected by the load-shed cost.

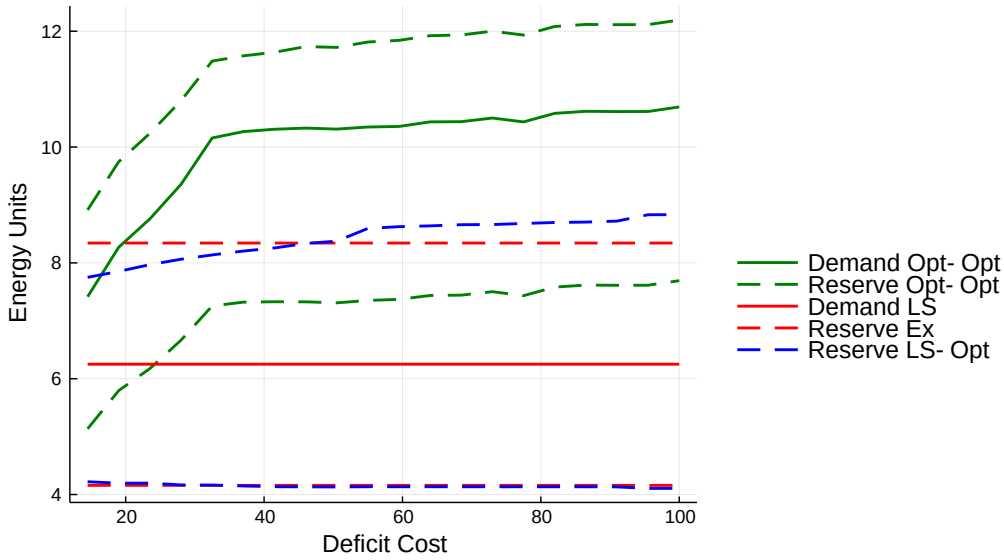


Figure 3.9: Long-run averages and reserve margins, in units of energy, as functions of deficit cost

3.8.7 Large-scale optimization

In this experiment, we apply the closed-loop estimation methodology to larger power system networks. As in the previous section, we focus our attention on the three models: LS-Ex, LS-Opt and Opt-Opt. Our primary goal in this case study is to demonstrate that the methodology is capable of consistently obtaining high-quality solutions that significantly improve the standard procedure in larger systems. For each of the 24, 118 and 300 bus systems, we considered 100 datasets, each of which with $1,000$ observations for the training stage. Each instance considered a time limit of 15 minutes for training, i.e., for solving (3-21)–(3-38) with $T = 1,000$. For the testing stage, we evaluated the out-of-sample cost using a common set of $10,000$ observations for each of the 100 obtained solutions. In Table 3.1, we present the mean and standard deviation of the 100 estimates of the operation costs for both the test and training stages.

As can be seen in Table 3.1, some training times are slightly above 15 minutes because of problem-building time. Compared to the benchmark method, LS-Ex, we can see improvements of 5% in the 24 bus system and 3% in the 118 system. In the 300 bus test case, we have a 4.8% improvement by endogenously sizing reserves and an 11.4% improvement by jointly estimating

the closed-loop demand forecast and reserve size. The latter case hit the time limit frequently, hence results could be even better with more time.

System	Model	Test Cost($)		Train Cost ($)		Train Time (s)	
		Mean	Std	Mean	Std	Mean	Std
24	LS-Ex	414.70	1.14	397.01	4.39	0.00	0.00
	LS-Opt	398.48	0.69	378.49	93.00	450.81	90.35
	Opt-Opt	398.20	1.04	376.35	14.04	643.89	11.16
118	LS-Ex	2956.01	11.33	3163.74	3.78	0.00	0.00
	LS-Opt	2829.86	4.20	3041.28	4.86	639.87	4.97
	Opt-Opt	2815.85	4.57	3029.93	13.20	911.43	12.37
300	LS-Ex	7697.25	40.78	7646.84	26.09	0.00	0.00
	LS-Opt	7329.44	36.62	7278.88	18.75	803.15	34.08
	Opt-Opt	6820.47	37.67	6787.65	294.53	923.38	25.12

Table 3.1: Results for the 24, 118 and 300-bus systems

3.8.8 Scalability of the proposed method

Although many systems are frequently represented with networks with a few hundred buses—for instance, [199] considers 255 buses in Peru and [200] considers 140 buses in Panama—we further analyze the algorithm's scalability and its consequent applicability to an even broader range of very large power system networks. We created instances with the number of buses ranging from 600 to 6000. These instances were created by connecting multiple copies of the 300 bus case. The optimization was performed on an Intel Xeon E5-2680 with 12 cores at 2.50GHz, 128Gb RAM. We generated a single training dataset with $1,000$ observations for each instance and optimized each problem considering four- and twelve-hour time limits. Then, we evaluated the solutions obtained with each method and computational time limit with a common dataset of $10,000$ out-of-sample observations.

The results, in terms of Test and Train cost, are depicted in Table 3.2. It is clear that the proposed methods (LS-Opt and Opt-Opt) consistently outperformed the least-squares benchmark (LS-Ex) despite reaching the time limit for training. Moreover, the co-optimization scheme (Opt-Opt) was again better than the other two methods (LS-Ex and LS-Opt) in both in-sample and out-of-sample evaluation. Also, the method performed well, generalizing from training to testing. For the larger systems, we see smaller relative improvements in the cost functions. This is due to (i) the increased dimension of linear programs solved in each iteration, and (ii) the number of parameters, which increases with the system size. Reasons (i) and (ii) imply that, for a given computational budget, fewer iterations are run in the training stage, possibly

yielding sub-optimal solutions. Thus, it is conceivable that much better results could be obtained with more time (or processing capacity), as the 12-hour runs lead to improvements that are typically more than three times those obtained with 4-hour runs. Although we were limited to 12 cores, one could use one core (or more) per observation in the training dataset, leading to massive speed-ups. It is worth mentioning that, in all cases, the LS estimation found coefficients very close to the true ones. This fact, together with the gains shown in Tables 3.1 and 3.2 for many instances of different sizes, indicate that the forecast bias introduced by our methodology is consistent in promoting an improved operation. Notably, relevant gains were found even in cases of very large-scale power systems. For instance, for the 3000-bus system, in this simulation run the gain was 13% with the 12-hour time limit. Therefore, the results provide strong evidence that our method is capable of producing meaningful gains in practice.

		Test					Train				
Buses	Time	Opt-Opt		LS-Opt		LS-Ex	Opt-Opt		LS-Opt		LS-Ex
	(h)	($)	(%)	($)	(%)	($)	($)	(%)	($)	(%)	($)
600	4	15256	21.51	16741	13.87	19437	14898	22.73	16552	14.16	19281
600	12	15226	21.66	16739	13.88	19437	14873	22.86	16563	14.10	19281
1200	4	35261	26.38	40103	16.28	47899	34985	26.87	39994	16.41	47843
1200	12	35158	26.60	38843	18.91	47899	34842	27.17	38787	18.93	47843
1800	4	60043	18.61	69355	5.98	73770	61033	17.97	69915	6.03	74405
1800	12	53378	27.64	62786	14.89	73770	54294	27.03	63224	15.03	74405
2400	4	81018	20.23	94675	6.78	101561	78686	19.57	91278	6.70	97832
2400	12	80330	20.91	93739	7.70	101561	77969	20.30	90413	7.58	97832
3000	4	120389	5.18	125465	1.19	126972	119214	5.27	124328	1.20	125841
3000	12	110302	13.13	122560	3.47	126972	109231	13.20	121382	3.54	125841
3600	4	149141	3.51	153484	0.69	154558	147110	3.50	151404	0.68	152439
3600	12	136479	11.70	150303	2.75	154558	134664	11.66	148332	2.69	152439
4200	4	177451	1.72	179785	0.43	180555	174395	1.70	176651	0.43	177406
4200	12	165963	8.08	177444	1.72	180555	162820	8.22	174309	1.75	177406
4800	4	206358	1.22	208300	0.29	208910	209816	1.28	211927	0.29	212546
4800	12	197707	5.36	206539	1.13	208910	201294	5.29	210197	1.11	212546
5400	4	232071	1.05	233992	0.23	234524	228660	1.23	231018	0.21	231514
5400	12	225203	3.97	232658	0.80	234524	222018	4.10	229769	0.75	231514
6000	4	260427	0.96	262493	0.18	262963	262049	0.99	264198	0.18	264668
6000	12	255384	2.88	261500	0.56	262963	257062	2.87	263118	0.59	264668

Table 3.2: Results for very large-scale systems. LS-Ex is the reference, only costs in $ are shown. For both Opt-Opt and LS-Opt costs are shown in $ and their improvement compared to LS-Ex is shown in %.

3.8.9 Tests on very large systems

In this section, we emulate a realistic usage of the method, by testing it in realistic large-scale instances. The ones marked with "*rte*" represent the French grid in 2013, using large-scale instances based on real systems [201]. The ones marked with "*pegase*" are synthetic but based on the European grid

[202]. All these instances are available in [195]. In this section, all system data in the test cases are considered without any modifications, except for the loads that come from a AR(1) process analogous to the previous sections. Moreover, we limited the computing time to 30 minutes so that the method can be used for dynamic reserve sizing and load forecast in hour-ahead dispatch. Note that using the method in other time scales, such as day-ahead dispatch, would also have great potential to benefit real systems around the globe. Even systems with slower dynamics running week-ahead dispatch could be benefited. The number of observations used for training was 600, and the number of out-of-sample observations for testing was 10000. We used larger servers with 64 processes in parallel and 1024 Gb of RAM to perform the computations, all instances are available in Amazon Web Services, AWS, [203].

	Test					Train				
Dataset	Opt-Opt		LS-Opt		LS-Ex	Opt-Opt		LS-Opt		LS-Ex
	($)	(%)	($)	(%)	($)	($)	(%)	($)	(%)	($)
6468-rte	19870	8.56	20026	7.85	21731	20278	8.50	20462	7.67	22163
6470-rte	25579	9.21	25849	8.25	28174	24004	10.38	24235	9.52	26785
6495-rte	28726	9.77	28988	8.94	31835	28400	10.52	28666	9.68	31739
6515-rte	34397	2.01	34557	1.56	35103	33757	2.01	33874	1.55	34354
9241-pegase	76438	12.89	76662	12.62	87750	75315	10.31	75509	10.08	83973
13659-pegase	82608	7.97	82908	7.63	89760	83930	7.50	84196	7.21	90737

Table 3.3: Results for very large-scale systems. LS-Ex is the reference, only costs in $ are shown. For both Opt-Opt and LS-Opt costs are shown in $ and their improvement compared to LS-Ex is shown in %.

Results are shown in Table 3.3. The number in the dataset name represents the number of buses in the system. These results provide even stronger evidence that our method is capable of producing meaningful gains in practice.

3.9 Conclusions

A mathematical framework for application-driven learning is proposed building upon the ideas of bilevel optimization. Asymptotic convergence is demonstrated, and we show that the proposed solution converges to the best forecast model on average for the selected application. Two solution techniques are presented: one exact, which is based on the KKT conditions of the second-level problem, and one heuristic, which scales well for very large instances while showcasing relevant gains in comparison to the benchmark. We apply the general application-driven framework to jointly estimate the parameters of a load and dynamic reserve requirements forecast model in a closed-loop fashion. Following the same framework, we also presented a novel (closed-

loop) method to estimate dynamic reserves if the demand forecast is fixed. The proposed methods are compared with the classical sequential open-loop procedure (benchmark), where the forecast models are estimated based on least squares and used in the decision-making process. Regarding our application case, the proposed framework finds support in current industry practices, where *ad hoc* procedures are implemented to bias load forecasts aiming to reduce risks empirically. The application of our model provides not only a theoretically-grounded understanding of such procedures but also a flexible computational tool for testing current practices and jointly determining the optimal bias and reserve requirements.

The reported numerical experience allows highlighting the following main empirical results and insights: 1) There exists an optimal bias in the load forecast maximizing the performance of a system or market operator in the long run. Moreover, the optimal bias in the load forecast is not disconnected from the optimal reserve requirements (reserve sizing problem). 2) Reserve requirements sizing and location are intrinsically dependent on the load-shed cost and system's characteristics, and a policy to dynamically allocate them can be optimally determined by our framework even in the case where we impose an exogenous estimate for the demand forecast (e.g., least squares or other methods). 3) Our model can endogenously define the optimal reserve sizing across the network by defining zonal reserve requirements that will best perform given the system operator's description of the network. 4) We show for realistic test systems, e.g., IEEE 300-bus system, that both models (optimizing only reserves and co-optimizing load forecast and reserves) were capable of significantly improving the long-run operation cost. Moreover, we demonstrated that the method scales to even larger power systems with thousands of buses and leads to consistent improvements in the long-run operation costs even with limited computational resources. Finally, we empirically proved that the method scales well and is able to handle large-scale realistic power systems, ranging from 6,468 to 13,659 buses, with a limited training time of 30 minutes resorting to cloud computing. Consequently, the method might be well fit for realistic hour-ahead planning besides day- and week-ahead. 5) We show that the proposed heuristic solution method can provide high-quality solutions in reasonable computational time. This is mostly due to the selected approach, which i) initializes the search method with the traditional least-square estimates, thereby only moving to another point if an improvement in performance is found, and ii) allows the estimation problem to be decomposed per observation and the second-level problem to be solved till global optimality in polynomial time. Additionally, it also leverages mature linear

programming-based warm-start technologies and algorithms to scale up the performance in larger instances. This pattern is consistently observed in all test systems, corroborating the proposed framework's effectiveness in finding improved estimates for both load and reserve requirements.

The proposed framework is fairly general, but we focused on right-hand side uncertainty and linear models as this was the relevant setting for the reserve requirement forecasting problem. Possible extensions of this work could aim at generalizing this setting. Indeed, the exact method would work for objective-function uncertainty and for non-linear models with strong duality like many conic programs. The heuristic method is even more flexible, it could accommodate uncertainty anywhere in the models, and it could be used in many non-linear (even integer) problems. Also, the heuristic method can be benefited from online optimization approaches and warm starts based on the solution of a previous execution of the algorithm with similar data as would occur in a real-time application. This could bring relevant gains to the optimization of sequential hour-ahead operation. Other aspects of estimation procedures to be explored include regularization with shrinkage operators [198], or the addition of LS estimates in the objective [157] with a penalizing coefficient to provide a balance between classical and application-driven forecasts. Moreover, it would be interesting to conduct experiments to verify the empirical performance of various statistical models (ψ), such as vector autoregressive models. Proving convergence conditions for many of the aforementioned changes in the model would provide nice contributions to the literature.

4
Long-term Hydrothermal Bid-based Market Simulator

Simulating long-term hydrothermal bid-based markets considering strategic agents is a challenging task. The representation of strategic agents considering inter-temporal constraints within a stochastic framework brings additional complexity to the already difficult single-period bilevel, thus, non-convex, optimal bidding problem. Thus, we propose a simulation methodology that effectively addresses these challenges for large-scale hydrothermal power systems. We demonstrate the effectiveness of the framework through a case study with real data from the large-scale Brazilian power system. In the case studies, we show the effects of market concentration in power systems and how contracts can be used to mitigate them. In particular, we show how market power might affect the current setting in Brazil. The developed method can strongly benefit policy makers, market monitors, and market designers as simulations can be used to understand existing power systems and experiment alternative designs.

Nomenclature

Sets and Indices

I – Set of agents indexed by i.
I_{-i} – Set of agents excluding agent i.
I^* – Set of price maker agents, indexed by i.
J^G – Set of thermal plants indexed by j.
J^H – Set of hydro plants indexed by j.
J^M – Set of Markov states indexed by j.
J^R – Set of renewable plants indexed by j.
$J^U(j)$ – Set of hydro plants upstream of plant j, indexed by y.
J^V – Set of vertices (points) representing a convex hull.
L – Set of lag in an autoregressive model, indexed by l.
S – Set of sampled scenario indices, indexed by s.
T – Set of time indices, indexed by t.

K – SDDP Iteration index.
$[K]$ – Set of cut indices at iteration K.
J_i – Subset of a set J with the indices of elements that belong to agent i.

Subscripts related to stage, t, and scenario, s, will be omitted for simplicity when they are not essential for the reader's understanding of sampling and chronological relations.

Constants

$\mathcal{A}$ – Vector of inflows of all hydros, for all stages and sampled scenarios.
C_j – Operating cost of thermal j.
E_j^Q – Energy quantity for an element j of a convex hull.
E_j^R – Energy revenue for an element j of a convex hull.
G_j – Maximum generation of thermal j.
P_i – Price offer of agent i.
$\mathcal{P}$ – Vector price offer of all agents, for all stages and sampled scenarios.
P^F – Forward contract price.
Q_i – Quantity offer of agent i.
$\mathcal{Q}$ – Vector of quantity offer of all agents, for all stages and sampled scenarios.
Q^F – Forward contract quantity.
$\tilde{R}_j(\omega)$ – Maximum generation of renewable k at scenario ω.
$\mathcal{R}$ – Vector of maximum generation of all renewables, for all stages and sampled scenarios.
U_j – Maximum flow through turbine of hydro j.
V_j – Maximum storage of hydro j.
β^k – Constant term of Benders cut k.
γ_j^k – Coefficient of state v_j^{t+1} of Benders cut k.
$\delta_{j,l}^k$ – Coefficient of state a_j^{t+1-l} of Benders cut k.
$\tilde{\varepsilon}_i^t(\omega)$ – Inflow noise coefficient of hydro j, stage t and scenario ω.
$\phi_{j,l}$ – Inflow auto-regressive coefficient of hydro j and lag l.
π_n – Spot price at system n.
$\mathit{\Pi}$ – Vector of spot prices, for all stages and sampled scenarios.
$M_{\mu|m}$ – Transition probability from Markov state μ to state m.
$\mathcal{M}$ – Vector of transition probabilities matrices, for all stages.

Indexing vectors in calligraphic $(\mathcal{P}, \mathcal{Q})$ by i stands for the sub-vector where elements belong to agent i and by $-i$ stands for the sub-vector where elements belong to all agents except i.

Optimization Variables

a_j^t – Inflow at hydro j, stage t. $a^{[t]}$ stands for the vector of all inflows before stage t.
$\mathbf{a}^{[t]}$ – Vector of inflows at all hydros, for all stages before stage t.
e – Energy offer
g_j – Generation of thermal j.
q – Bid quantity accepted in a dispatch.
r_j – Generation of renewable j.
u_j – Turbine flow at hydro j.
v_j^t – Storage at hydro j, at the beginning of stage t, and at the end of stage $t-1$.
$\mathbf{v}^t$ – Vector of storage at all hydros, at the beginning of stage t, and at the end of stage $t-1$.
z_j – Spill flow at hydro j.
α – Epigraph variable for Benders cuts.
λ_j – Convex hull value for vertex j.

Indexing vectors in bold $(\mathbf{a}, \mathbf{v})$ by i stands for the sub-vector where elements belong to agent i.

Functions and Functionals

$\mathbb{E}_\omega[\cdot]$ – Expected value over the random variable ω.
$\tilde{B}(\cdot,\cdot)$ – Future cost function as a function of states and uncertainty.
$\tilde{\Lambda}(e,\omega)$ – Revenue as a function of the energy bid and the scenario ω.
$\rho(\cdot)$ – Hydro generation as a function of turbine flow. It can also be a function of volume.
$|\cdot|$ – Cardinality of a set

4.1 Introduction

Hydropower is one of the most widely used energy sources around the globe, the most used renewable energy source responsible for over four thousand TWh per year according to the International Energy Agency [204]. Many countries rely on hydro plants for a meaningful share of their generation, but some countries have hydropower as their main energy source, for instance, Brazil, New Zealand, Colombia, and Norway.

The operation of systems with large penetration of hydropower is very challenging since reservoirs are storage devices that create a strong temporal coupling and inflows are random variables. The framework of Multi-Stage Stochastic Optimization (MSO) is the most used framework to handle this type of problem and the key algorithm is Stochastic Dual Dynamic Programming, SDDP, which was motivated by the centralized cost-based hydrothermal power system operation [205].

The changes in the regulation of power systems in the last decades have led to liberalized bid-based power markets, including the hydrothermal ones [206]. A new variant of SDDP that combines discretization ideas from the less scalable standard Stochastic Dynamic Programming (SDP) was developed in [207] for the usage of market participants that want to incorporate price uncertainty in their scheduling and bid decisions. The latter is also known as the Markovian SDDP [208].

The bidding problem for hydrothermal power systems is reviewed in [209], which classifies the problem into two main variants depending on whether the bidder is a price taker or a price maker. A price taker, or non-strategic agent, is an agent that is either too small to affect market prices with changes in its operation or a large agent that is not willing to do so. Conversely, a price maker, or a strategic agent, is an agent that can change prices by changing its operation, this is also known as market power.

The review also contrasts the hydrothermal version of the problem, which includes both time coupling and uncertainty, with the purely thermal version of the problem, where time coupling is ignored and uncertainties are frequently disregarded as well. The main conclusion is that the price-taker versions of the problem are mostly solved in the literature since thermal generators will offer their variable operating costs [210]. Meanwhile, hydro agents will bid based on their water marginal costs, which can be viewed from the operator's perspective [211] or from the agent's perspective [207].

On the other hand, the price-maker versions of the problem are much harder to handle computationally. Simulating the behavior of one single price maker agent has been done using Bilevel Optimization [69] where the leader is a price maker, also known as a strategic bidder and the follower problem represents the cost minimization market dispatch. An extension of such a framework for the case of purely hydro strategic agents where the remainder of the market is purely thermal was developed in the seminal work [212]. The latter embeds a bilevel program in the SDDP algorithm but convexifies each sub-problem to follow the assumptions of SDDP. A variant of the previous method was proposed in [213] that handles the non-convex nature of the sub-

problem with Lagrangian relaxation combined with SDDP. [214] uses a purely Lagrangian decomposition scheme to solve a multistage MIP, but this time for a price maker demand side agent. [215] simulates a hydro agent acting as a price taker in the energy market and acting as a price maker in the reserves market. The reserves market is represented as a simple linear curve, and the problem is solved with SDP due to hydro commitment representation. Similarly, [216] considers a battery that is a price taker for energy but a price maker for reserve, the market price curve is a step function like in [212], but it is modeled as MIP since the entire problem will be passed to a MIP solver with no decomposition.

Even more challenging is the simulation of multiple agents acting as price-makers in a market. One key framework is game theory. For instance, in the purely thermal case, [217] models the thermal version as a Nash equilibrium using KKT conditions of the market operation and binary expansions to optimize an approximation of the numerical problem with MIP solvers. This technique was recently improved by using column constraint generation to solve larger instances [218].

Again, the hydrothermal case is more challenging. One of the earliest works, [219], models a small two-reservoir problem with 10 plants in a so-called Dual Dynamic Programming scheme (different from the one of [205]) where each sub-problem is a Cournot duopoly. Next, [220] developed a Stochastic Dynamic Programming scheme to better consider the effect of stochastics in a problem with 2 hydros and 23 thermals. In the two works, contracts are tested to mitigate market power. [221] uses deterministic dynamic programming to simulate 3 agents (but only 2 with reservoirs) in a short-term market. In the last reference, the single-stage sub-problems are solved with an iterative scheme, also called diagonalization in the literature. Iterative schemes that resemble the diagonalization method are the Nikaido-Isoda Function method employed in [222] to solve a deterministic 3-player game modeling day-ahead market in Chile and the ADMM-based approach applied to the risk-based capacity expansion in [223].

More recently, [224] uses a deterministic dynamic programming strategy to simulate the behavior of two hydro reservoirs competing where each stage requires the solution of a MIP that searches for Pareto optimal equilibrium points. [225] solves the problem with a modified dynamic programming scheme that accounts for uncertainty, with 15 scenarios, and solves a three-agent subproblem iteratively to approximate the commitment decision while the equilibrium in each stage is modeled as a MIP similar to [217]. Finally, [226] employs Sampling Stochastic Dynamic Programming (SSDP) where stagewise subproblems are solved via diagonalization.

The second group of methods used to simulate the interaction between multiple price-maker agents in a hydrothermal market does not use dynamic programming to decompose the problem. Consequently, these methods are used for shorter-term or medium-term analysis with very few time stages. [227] is a seminal work in the subject, which models the multistage competition problem in a monolithic Equilibrium Program with Equilibrium Constraints (EPEC) derived from KKT conditions and solves a 7-stage instance with an EPEC solver. [228] also models the problem as an EPEC and solves smaller instances with an EPEC solver, but the 1-year problem with 4 scenarios is solved via diagonalization. Similarly, [229] solves a 24-hour problem with 7 players by modeling an EPEC and solving it by diagonalization. A similar strategy is presented in [230]. [231] describes each agent subproblem as a tree-based MSO and connects them with equilibrium constraints. The case study is a two-stage problem with 10 scenarios solved by a specialized EPEC solver. [232] models an EPEC and solves the problems with a modified Benders decomposition.

The above-referenced literature makes it clear that simulating multiple agents in a hydrothermal long-term bid-based market is extremely challenging because it is necessary to handle multiple stages, uncertainty, and the interplay of multiple agents, which leads to non-convexities. Consequently, most works strongly limit the number of stages, agents, or scenarios. Although some of the above methods can handle all these features, in theory, the simulations are done in small problems.

Therefore, the main contributions of this work are:

- Developing a methodology that can handle, at the same time, multiple stages, agents, and uncertainty scenarios. Such methodology will be based on solving multistage stochastic strategic bidding problems for each strategic agent, while the coupling between agents is achieved by an iterative procedure based on diagonalization.
- Analysing sensitivities of the competition of multiple agents with different market shares.
- Analysing how contracts can affect the market simulations as a mitigation strategy for market power.
- Simulating competition of multiple price maker agents in the large-scale Brazilian power system with real data.

The remainder of this work is organized as follows: Section 4.2 introduces basic notation describing a centralized model for hydro-thermal power systems.

Section 4.3 describes the optimization of a single price maker agent and highlights how to incorporate contracts. Section 4.4 details an algorithm to combine the above models to simulate long-term hydro-thermal bid-based markets in the presence of multiple strategic agents. Section 4.5 presents case studies to test the proposed algorithm and simulates a real case of the Brazilian power system. Section 4.6 exposes the main conclusions.

4.2 Long-term hydro-thermal dispatch and SDDP

The hydrothermal power system dispatch is a very complex problem because it is a multi-stage stochastic optimization problem that includes many physical and policy-driven constraints. We present a simple yet general optimization model that has all the main core features that are required in the market simulator proposed in this work. We describe the problem in the form of a Bellman recursion as in [205]. Therefore, (4-1)–(4-7) presents the objective function and constraints of a given stage, t, and random event ω^t. Index t is omitted from most variables when easily understood from context aiming at a lighter notation.

The first formula, (4-1), states that the future cost of stage $t-1$, namely $\tilde{B}^t$, given the states $\{\mathbf{v}^t, \mathbf{a}^{[t-1]}\}$ and the random event ω^t, is defined as the minimization of the problem (4-1)–(4-7), whose objective function can be split into two pieces. The first piece is the immediate cost in the form of a thermal cost while the second piece is the expected value of the future cost $\tilde{B}^{t+1}$, with $\tilde{B}^{|T|+1}(\cdot) = 0$. The equation (4-2) represents the load balance of the system, we have generation and energy flows on the left-hand side and the demand on the right-hand side. The demand is considered deterministic to simplify the developments, but everything that follows can be trivially extended to the stochastic demand case. (4-3) describes the water mass balance: storage at the end of stage t equals the storage at the beginning plus the net sum of incoming water and outflows. (4-4) constrains hydro storage, turbine flow, and spillage, while (4-5) limits the thermal generation and (4-6) limits the renewable generation that has a stochastic upper bound depending on events like sun and wind. Finally, (4-7) describes the inflow auto-regressive stochastic process [205, 233].

$$\tilde{B}^t\Big(\mathbf{v}^t, \mathbf{a}^{[t-1]}, \omega^t\Big) =$$

$$\min \sum_{j \in J^G} C_j g_j + \mathbb{E}_{\omega^{t+1}}\left[\tilde{B}^{t+1}\Big(\mathbf{v}^{t+1}, \mathbf{a}^{[t]}, \omega^{t+1}\Big)\right] \tag{4-1}$$

$s.t.$

$$\sum_{j \in J^G} g_j + \sum_{j \in J^H} \rho_j(u_j) + \sum_{j \in J^R} r_j = D \tag{4-2}$$

$$v_j^{t+1} = v_j^t - u_j - z_j + \sum_{n \in J^U(j)} (u_n + z_n) + a_j^t, \quad \forall j \in J^H \tag{4-3}$$

$$0 \leq v_j \leq V_j, \quad 0 \leq u_j \leq U_j, \quad 0 \leq z_j, \quad \forall j \in J^H \tag{4-4}$$

$$0 \leq g_j \leq G_j, \quad j \in J^G \tag{4-5}$$

$$0 \leq r_j \leq \tilde{R}_j(\omega^t), \quad \forall j \in J^R \tag{4-6}$$

$$a_j^t = \sum_{l \in L} \phi_{j,l} a_j^{t-l} + \tilde{\varepsilon}_j(\omega^t), \quad \forall j \in J^H \tag{4-7}$$

The application of the SDDP algorithm requires that we consider samples, $\{\omega_s^t, s \in S\}$, of the random events ω^t for each $t \in T$, and rewrite of the Bellman recursion considering approximations of the future cost function (FCF). $\tilde{B}_K^t$ is the approximation of the FCF at stage t and iteration K. The future cost function of $t+1$ is approximated by using an epigraph formulation with the variable α and the Benders cuts (4-9) that belong to the set of cuts generated up to iteration K, this set will be referred to as $[K]$.

$$\tilde{B}_K^t\Big(\mathbf{v}^t, \mathbf{a}^{[t-1]}, \omega^t\Big) =$$

$$\min \sum_{j \in J^G} C_j g_j + \alpha \tag{4-8}$$

$s.t.$

Constraints (4-2) $-$ (4-7)

$$\alpha \geq \beta^k + \sum_{j \in J^H} \left(\gamma_j^k v_j^{t+1} + \sum_{l \in L} \delta_{j,l}^k a_j^{t+1-l}\right), \quad \forall k \in [K] \tag{4-9}$$

In a very high-level description, the SDDP algorithm starts sampling a scenario for each $t \in T$, then solves the stage subproblems in chronological order generating a feasible solution (a candidate operation of the power system), which is called the forward step. After that, in the so-called backward step, problems are solved in the reverse order of time, generating cuts to improve the representation of the FCF and, thus, propagating information from the future to the present. These steps are repeated until a specified stopping criterion is reached. This process is also known as policy optimization

(or training). For details of the SDDP algorithm, the reader is directed to [205, 234]. After convergence is declared, it is usual to proceed with a final forward pass in which we sample scenarios $s \in S$, and for each of them, we solve all stage subproblems in chronological order. This last procedure is known as simulation, and it results in what we call a solution for each primal (and dual) variable. In other words, we finish with values for each stage in T and sample scenario in S for all optimization variables. In the case of volumes, we describe this set of vectors of volumes as $\{\mathbf{v}_{t,s}\}_{t\in T,s\in S}$. For the case of spot prices, i.e., the dual variable associated with the load balance, (4-2), we will represent it as $\mathit{\Pi} = \{\pi_{t,s}\}_{t\in T,s\in S}$, meaning that we have solutions for each sampled scenario $s \in S$ and stage $t \in T$.

In this section, we focused our presentation on the expected value case of MSO for simplicity. However, everything written in the above, as well as in the next sections, holds for other risk measures that can be represented in the SDDP scheme [235, 236]. Another simplification in this work is to consider the $\rho(\dot{})$ function, in (4-3), as linear function [64, 208, 237], non-linear models can also be used with approximations to satisfy the requirements of SDDP [238].

4.3 Strategic agents

The optimization of an independent agent, or owner, can also be modeled as a multi-stage stochastic optimization problem. It was modeled and first solved with SDDP in [212]. Model (4-10)–(4-16), representing the optimization of an individual agent, i, is very similar to (4-1)–(4-7), the centralized operation model. (4-10) represents a Bellman recursion analogous to the one of the centralized dispatch. The main difference in the objective function is that there is a new term to represent the revenue of the agent in the given stage t for a random event ω^t as a function of the energy e produced by the agent. (4-11) states that e equals the total generation of all resources possessed by an agent. Finally, (4-12)–(4-16) are almost the same as (4-3)–(4-7) but only considering plants owned by the given agent, i.

$$\tilde{B}^t\Big(\mathbf{v}_i^t, \mathbf{a}_i^{[t-1]}, \omega^t\Big) =$$

$$\min \quad -\tilde{\Lambda}(e, \omega^t) + \sum_{j \in J_i^G} C_j g_j + \mathbb{E}_{\omega^{t+1}}\left[\tilde{B}^{t+1}\Big(\mathbf{v}_i^{t+1}, \mathbf{a}_i^{[t]}, \omega^{t+1}\Big)\right] \tag{4-10}$$

$s.t.$

$$e = \sum_{j \in J_i^G} g_j + \sum_{j \in J_i^H} \rho_j(u_j) + \sum_{j \in J_i^R} r_j \tag{4-11}$$

$$v_j^{t+1} = v_j^t - u_j - z_j + \sum_{y \in J^U(j)} (u_y + z_y) + a_j^t, \quad \forall j \in J_i^H \tag{4-12}$$

$$0 \leq v_j \leq V_j, \quad 0 \leq u_j \leq U_j, \quad 0 \leq z_j, \forall j \in J_i^H \tag{4-13}$$

$$0 \leq g_j \leq G_j, \quad \forall j \in J_i^H \tag{4-14}$$

$$0 \leq r_j \leq \tilde{R}_j(\omega^t), \quad \forall j \in J_i^R \tag{4-15}$$

$$a_j^t = \sum_{l \in L} \phi_{j,l} a_j^{t-l} + \tilde{\varepsilon}_j^t(\omega^t), \quad \forall j \in J_i^H \tag{4-16}$$

One fundamental challenge in this procedure is that it requires that all hydro plants that are connected must belong to a single agent. This hypothesis is also assumed in previous works like [213]. Handling different owners in the same river system can be done by considering a water wholesale market besides the energy market as done in [239]. Alternatively, other market designs, such as 1) *hydro slices*, in which agents are owners of a fraction of the cascade and, consequently, can optimize their strategies as if they did not share the cascade, and 2) *virtual hydro reservoirs*, in which all hydro plants of a system are aggregated in a single reservoir which is then split proportionally among all agents, for more details see [240].

Analogously to the previous section, we present the approximated version of the problem to be solved by the SDDP algorithm in (4-17)–(4-18). The key change is that the revenue function uncertainty might be time-dependent, just like the inflow processes. While the inflow time dependency is handled by an expanded state-space considering inflow lags, the time dependency in the revenue function is modeled with a Markov Chain. Hence, we apply the Markovian SDDP as in [207] – this Markovian representation was not needed in [212, 213] that considers a time-independent revenue function, perfectly determined by the system's thermal plants. Therefore, we have the epigraph of multiple FCFs, one FCF for each possible Markov state, m, represented by α_m, and the respective cuts in (4-16).

$$\tilde{B}^t_{\mu,K}\left(\mathbf{v}^t_i, \mathbf{a}^{[t-1]}_i, \omega^t_\mu\right) =$$

$$\min \quad -\tilde{\Lambda}(e, \omega^t_\mu) + \sum_{j \in J^G_i} C_j g_j + \sum_{m \in J^M} M_{\mu|m} \alpha_m \tag{4-17}$$

$s.t.$

(4-11) – (4-16)

$$\alpha_m \geq \beta^k_m + \sum_{j \in J^H_i} \left(\gamma^k_{j,m} v^{t+1}_j + \sum_{l \in L} \delta^k_{j,l,m} a^{t+1-l}_j\right), \forall m \in J^M, k \in [K] \tag{4-18}$$

In the case of price-taker agents, their operation can be optimized by considering the revenue function, as in [207]:

$$\tilde{\Lambda}(e, \omega) = \pi(\omega) e \tag{4-19}$$

where $\pi(\omega)$ is a scenario of spot prices that might be obtained from any model, including a previous optimization of the system centralized dispatch from the previous section.

On the other hand, price-maker agents have the ability to affect spot prices with their energy offers. In other words, π is no longer an exogenous input data, it is a function of the energy offer of agent i: e. In this case, price and quantity bids from other agents are considered random variables depending on the uncertainty ω, and are given $\{(P_j(\omega), Q_j(\omega)), j \in I_{-i}\}$. With those bids, we follow the same rationale of [212, 213]. Therefore, we write the following model that expresses a simple bid-based dispatch problem:

$$\pi(e, \omega) \in \arg\min \sum_{j \in I_{-i}} P_j(\omega) q_j \tag{4-20}$$

$$s.t. \sum_{j \in I} q_j = D \quad : \quad \pi \tag{4-21}$$

$$0 \leq q_i \leq e \tag{4-22}$$

$$0 \leq q_j \leq Q_j(\omega), \quad \forall j \in I_{-i} \tag{4-23}$$

This is analogous to a market clearing problem, in which a system operator selects the optimal amounts of energy to attain the demand in (4-21), under the limits (4-22)–(4-23) given by the current agent offer and the other players' offers. The main result is the spot prices, the dual variable of (4-21). Now we simply define:

$$\tilde{\Lambda}(e, \omega) = \pi(e, \omega) e \tag{4-24}$$

Combining (4-10)–(4-16), (4-24) and (4-20)–(4-23) completes the definition of the strategic agent MSO problem, where each sub-problem is a bilevel optimization problem as follows:

$$\tilde{B}^t\left(\mathbf{v}_i^t, \mathbf{a}_i^{[t-1]}, \omega^t\right) =$$

$$\min \quad -\pi(e,\omega^t)e + \sum_{j\in J_i^G} C_j g_j + \mathbb{E}_{\omega^{t+1}}\left[\tilde{B}^{t+1}(\mathbf{v}_i^{t+1}, \mathbf{a}_i^{[t]}, \omega^{t+1})\right] \tag{4-25}$$

s.t.

$$e = \sum_{j\in J_i^G} g_j + \sum_{j\in J_i^H} \rho_j(u_j) + \sum_{j\in J_i^R} r_j \tag{4-26}$$

$$v_j^{t+1} = v_j^t - u_j - z_j + \sum_{y\in J^U(j)} (u_y + z_y) + a_j^t, \quad \forall j \in J_i^H \tag{4-27}$$

$$0 \le v_j \le V_j, \quad 0 \le u_j \le U_j, \quad 0 \le z_j, \forall j \in J_i^H \tag{4-28}$$

$$0 \le g_j \le G_j, \quad \forall j \in J_i^H \tag{4-29}$$

$$0 \le r_j \le \tilde{R}_j(\omega^t), \quad \forall j \in J_i^R \tag{4-30}$$

$$a_j^t = \sum_{l\in L} \phi_{j,l} a_j^{t-l} + \tilde{\varepsilon}_j^t(\omega^t), \quad \forall j \in J_i^H \tag{4-31}$$

$$\pi(e,\omega^t) \in \arg\min \sum_{j\in I_{-i}} P_j(\omega^t) q_j \tag{4-32}$$

$$s.t. \tag{4-33}$$

$$\sum_{j\in I} q_j = D \qquad : \ \pi \tag{4-34}$$

$$0 \le q_i \le e \tag{4-35}$$

$$0 \le q_j \le Q_j(\omega^t), \quad \forall j \in I_{-i} \tag{4-36}$$

However, this strategic agent revenue function is a saw-shaped non-convex piece-wise linear discontinuous function as detailed in [212, 213]. Thus, to satisfy the SDDP convexity requirements, we follow the method proposed in [212] and represent its convex hull of $\tilde{\Lambda}(e,\omega)$ with respect to e for a fixed ω:

$$convhull(\tilde{\Lambda}(e,\omega)) = \max_{\lambda_j \ge 0} \sum_{j\in J^V} \lambda_j E_j^R(\omega) \tag{4-37}$$

$$s.t. \sum_{j\in J^V} \lambda_j E_j^Q(\omega) = e \tag{4-38}$$

$$\sum_{j\in J^V} \lambda_j = 1 \tag{4-39}$$

where each pair $(E_j^Q(\omega), E_j^R(\omega))$ represents one vertex (in the the set of vertices J^V) of the convex hull for the hypo-graph of $\tilde{\Lambda}(e,\omega)$. We keep the references

to ω in the vertices of the convex hull to remember their dependency on the random variables of the problem. We note that, theoretically, it would be possible to represent the non-convex function with no approximations with methods like SDDiP [24], but the computational cost for the problem optimization would increase considerably.

Given the above reformulation, we can rewrite the model (4-25)–(4-36), using (4-17)–(4-18) and (4-37)–(4-39), to obtain the convexified Markovian SDDP form:

$$\tilde{B}^t_{\mu,K}\left(\mathbf{v}^t_i, \mathbf{a}^{[t-1]}_i, \omega^t_\mu\right) =$$

$$\min \quad -\sum_{j\in J^V} \lambda_j E^R_j(\omega^t_\mu) + \sum_{j\in J^G_i} C_j g_j + \sum_{m\in J^M} M_{\mu|m}\alpha_m \tag{4-40}$$

$s.t.$

$$e = \sum_{j\in J^G_i} g_j + \sum_{j\in J^H_i} \rho_j(u_j) + \sum_{j\in J^R_i} r_j \tag{4-41}$$

$$v^{t+1}_j = v^t_j - u_j - z_j + \sum_{y\in J^U(j)} (u_y + z_y) + a^t_j, \quad \forall j \in J^H_i \tag{4-42}$$

$$0 \le v_j \le V_j, \quad 0 \le u_j \le U_j, \quad 0 \le z_j, \forall j \in J^H_i \tag{4-43}$$

$$0 \le g_j \le G_j, \quad \forall j \in J^H_i \tag{4-44}$$

$$0 \le r_j \le \tilde{R}_j(\omega^t_\mu), \quad \forall j \in J^R_i \tag{4-45}$$

$$a^t_j = \sum_{l\in L} \phi_{j,l} a^{t-l}_j + \tilde{\varepsilon}^t_j(\omega^t_\mu), \quad \forall j \in J^H_i \tag{4-46}$$

$$\lambda_j \ge 0 \tag{4-47}$$

$$\sum_{j\in J^V} \lambda_j E^Q_j(\omega^t_\mu) = e \tag{4-48}$$

$$\sum_{j\in J^V} \lambda_j = 1 \tag{4-49}$$

$$\alpha_m \ge \beta^k_m + \sum_{j\in J^H_i} \left(\gamma^k_{j,m} v^{t+1}_j + \sum_{l\in L} \delta^k_{j,l,m} a^{t+1-l}_j\right), \forall m \in J^M, k \in [K] \tag{4-50}$$

which is clearly a linear subproblem.

As one of the key mechanisms to mitigate market power, we must also be able to represent forward contracts [220]. This was not described in the previous MSO models solved by SDDP. However, it is simple to modify the revenue function to consider two additional terms as follows:

$$\tilde{\Lambda}(e, \omega) = P^F Q^F - \pi(e, \omega) Q^F + \pi(e, \omega) e \tag{4-51}$$

the first term is the fixed revenue of the forward contract, the second represents the energy that must be delivered due to the contract, and the third is

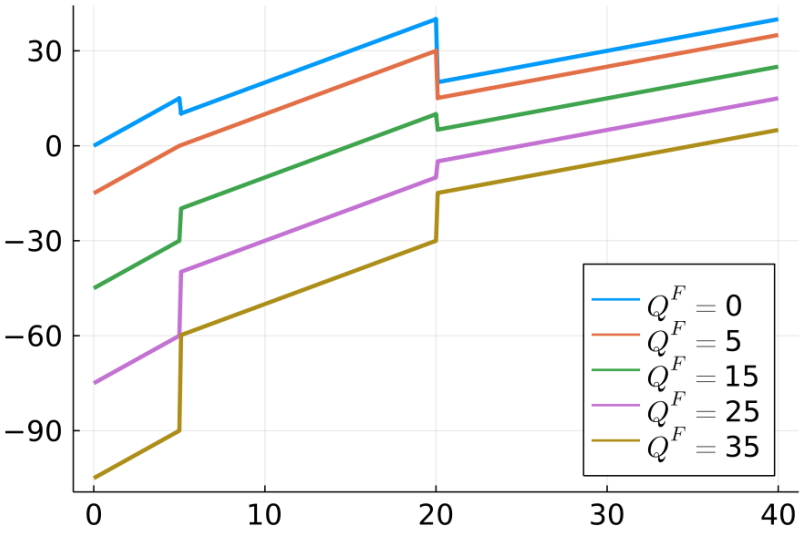


Figure 4.1: Revenue curves, $\tilde{\Lambda}(e, \omega)$, for various values of Q^F.

the previously represented earnings from the spot market. The constants P^F and Q^F are input data. Consequently, they are not decision variables in the optimization problems. Figure 4.1 shows revenue curves for a simple case where $P^F = 0\$$, $D = 40MWh$, and we consider 3 price-quantity offers from the other agents $[(3\$, 10MWh), (2\$, 15MWh), (1\$, 20MWh)]$. Note that the total quantity from the sum of the other agents' offers is $45MWh$, which is higher than the demand, and, hence, no deficit occurs even if $e = 0MWh$. This function is also non-convex, but we can represent its convex hull in the optimization problem in the exact same way as the previously described case with no contracts.

Finally, we remark that in both [212] and [213], the revenue function is not stochastic since it only considers thermal bids in the form of thermal installed capacities and operating costs and does not require the Markovian SDDP. In this setting, we allow stochastic and time-dependent energy bids from other agents. The more general framework of this work requires the usage of Markovian SDDP, just like in the price taker case with spot price uncertainty [207].

4.4 Multiple agents

Now we will combine the methodologies depicted in the previous sections to describe a novel simulation algorithm that approximates the interaction between multiple price-maker and possible price-taker agents. All the interactions between agents will be in terms of price and quantity bid offers as in a real competitive energy market.

First, we note that we can obtain an initial hint on the bids of multiple agents from a simulation of the centralized operation of the power system. As described in Section 4.2, the results of a simulation of the power system include primal and dual variable solutions for each t and s. By collecting the generation decisions of all plants of an agent, i, at a given stage and scenario, we have the quantity bid, while the price part of the bid is given by the corresponding spot price. We denote the set of price and quantity bids of an agent i as $(\mathcal{P}_i, \mathcal{Q}_i) = \{P_{i,t,s}, Q_{i,t,s}\}_{t\in T, s\in S}$, which means that there is one pair of price and quantity for each sampled scenario, $s \in S$, and stage, $t \in T$. We will name the procedure of solving a centralized dispatch and obtaining both spot prices $\mathit{\Pi} = \{\pi_{s,t}\}_{t\in T, s\in S}$ and bids for all agents $(\mathcal{P}, \mathcal{Q})$ as $\mathit{CentralizedOperation}(\mathcal{A}, \mathcal{R})$. The inputs $(\mathcal{A}, \mathcal{R})$ represent scenarios of inflows and renewable maximum generation, which are samples of the random variables for all stages. Because bids and spot prices, $(\mathcal{P}, \mathcal{Q}, \mathit{\Pi})$, are obtained from a procedure that depends on inflows and renewable generation, $(\mathcal{A}, \mathcal{R})$, all these have to be considered dependent random variables.

The self-optimization of strategic agents requires bids from all other agents as described in Section 4.3. We also highlighted that all price and quantity bids from other agents could be seen as time-dependent random variables in case they come from non-purely thermal price-taker agents. Therefore, solving this multi-stage stochastic problem will require the Markovian SDDP to handle the time dependency of the random variables. We can estimate a Markov Chain based on inflows, renewable generation, spot price, and bid data, $(\mathcal{A}, \mathcal{R}, \mathit{\Pi}, \mathcal{P}, \mathcal{Q})$ that are all the random variables associated to the strategic agent optimization. The estimation of the Markov chain results in transition probabilities $M^t_{\mu|m}$ between the Markov states μ and m for each stage t. The collection of transition probabilities, $M^t_{\mu|m}$, between all Markov states for all stages, $t \in T$, will be denoted as $\mathcal{M}$. This process that takes the tuple $(\mathcal{A}, \mathcal{R}, \mathit{\Pi}, \mathcal{P}, \mathcal{Q})$ and returns $\mathcal{M}$, will be denoted as $EstimateMarkovChain(\mathcal{A}, \mathcal{R}, \mathit{\Pi}, \mathcal{P}, \mathcal{Q})$.

Meanwhile, the strategic optimization of an agent, i, as a function of the inflows, $\mathcal{A}$, renewable generation, $\mathcal{R}$, bids of other agents, $(\mathcal{P}_{-i}, \mathcal{Q}_{-i})$ and the Markov transition probabilities, $\mathcal{M}$, will be labeled $\mathit{StrategicBid}(\mathcal{A}, \mathcal{R}, \mathcal{P}_{-i}, \mathcal{Q}_{-i}, \mathcal{M})$. Analogous to the simulation step of $\mathit{CentralizedOperation}(\mathcal{A}, \mathcal{R})$, the $\mathit{StrategicBid}(\cdot)$ will return an updated bid, $(\mathcal{P}_i, \mathcal{Q}_i)$, for the optimized agent, i: the quantities will be the primal solutions of the energy offer e, and prices will be obtained by computing the respective spot price in (4-20)–(4-23).

Finally, we can obtain updated spot prices, $\mathit{\Pi}$, by clearing the market

for each stage and scenario given bids from all agents, $(\mathcal{P}, \mathcal{Q})$. We will label the procedure: $ClearMarket(\mathcal{P}, \mathcal{Q})$.

The complete simulation algorithm is based on the diagonalization method extensively used in the literature of competitive hydro-power markets [221, 228, 229, 226, 230]. We will resort to the above-defined procedures to initialize and then iteratively update the bids of one price maker agent at a time while the bids from other agents are fixed. The process stops when changes in those bids are within some given tolerance. The main goal is to simulate the power market in an agent-based fashion. If convergence is strictly attained, we might have reached a Nash equilibrium. However, such an equilibrium might not even exist in this setting.

Algorithm 2 depicts the proposed method. The algorithm starts by computing bids, $(\mathcal{P}, \mathcal{Q})$, for all price taker and price maker agents with the $CentralizedOperation(\mathcal{A}, \mathcal{R})$ procedure. The bids from price taker agents will be frozen from this point onward as they are assumed to believe that their optimal bids are their opportunity costs [56, 207]. Then, a first estimate of the Markov process is made with the $EstimateMarkovChain(\mathcal{A}, \mathcal{R}, \Pi, \mathcal{P}, \mathcal{Q})$ procedure. After that, there will be a loop through all the price maker agents, $i \in I^*$. For each agent i, new bids, $(\mathcal{P}_i, \mathcal{Q}_i)$, will be obtained from the $StrategicBid(\mathcal{A}, \mathcal{R}, \mathcal{P}_{-i}, \mathcal{Q}_{-i}, \mathcal{M})$ procedure. Then, spot prices and the Markov process estimation will be updated by calls to $ClearMarket(\mathcal{P}, \mathcal{Q})$ and $EstimateMarkovChain(\mathcal{A}, \mathcal{R}, \Pi, \mathcal{P}, \mathcal{Q})$. Finally, if some convergence criterion is attained, the algorithm stops, otherwise, it continues to a new round of bid updates of the price maker agents. The convergence criterion considered in this work is assuring the maximum absolute variation of the price and quantity bids of all agents vary less than a given small number: 1% of their values in the centralized operation.

Algorithm 2: Multiple Agent Simulation

$(\mathcal{P}, \mathcal{Q}, \Pi) \leftarrow CentralizedOperation(\mathcal{A}, \mathcal{R})$
$\mathcal{M} \leftarrow EstimateMarkovChain(\mathcal{A}, \mathcal{R}, \mathcal{P}, \mathcal{Q}, \Pi)$
while *no convergence* **do**
  for $i \in I^*$ **do**
    $(\mathcal{P}_i, \mathcal{Q}_i) \leftarrow StrategicBid(\mathcal{A}, \mathcal{R}, \mathcal{P}_{-i}, \mathcal{Q}_{-i}, \mathcal{M})$
  end
  $\Pi \leftarrow ClearMarket(\mathcal{P}, \mathcal{Q})$
  $\mathcal{M} \leftarrow EstimateMarkovChain(\mathcal{A}, \mathcal{R}, \mathcal{P}, \mathcal{Q}, \Pi)$
end

4.5 Case studies

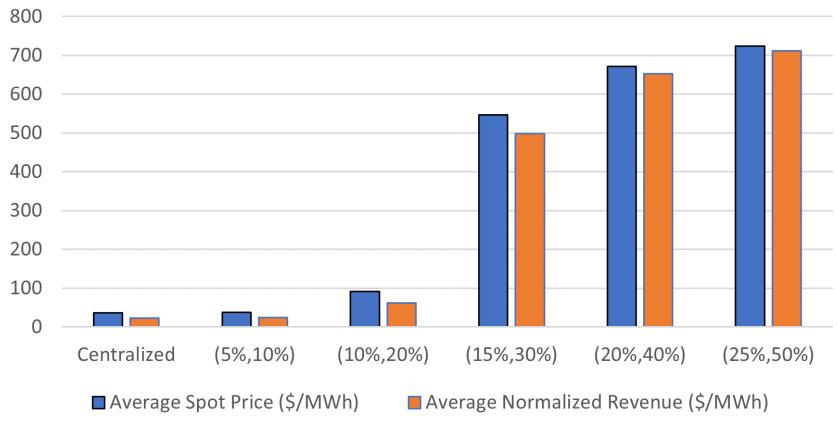
Now, we apply the simulation procedure of Section 4.4 in various settings. We start with a sensitivity analysis in a representation of the Brazilian Southeast subsystem, which accounts for about 55% of the Brazilian hydro resources and about 50% of the system's total installed capacity. We will analyze diverse possible market concentration and contracting schemes. In a second case study, we will consider a database developed from the complete Brazilian system that is very large scale. We will apply the proposed methodology to contrast results from simulations with and without contracts. Many simplifications were done in both power systems, including not considering many thermal, hydro, and other policy constraints in order to keep the analysis straightforward and demonstrate the capabilities of the proposed simulation procedure without distracting the reader.

4.5.1 The Brazilian Southeastern System

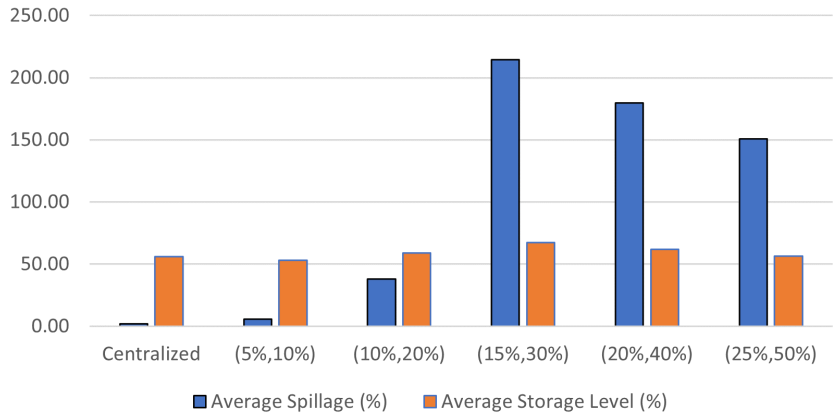
This system was constructed from real-life data considering the Brazilian system expansion scenario for 2025 and contains 46 thermal plants that were kept as close as possible to the original ones and 21 hydro plants with a simplified topology of the system. Note that this is already more than the current 13 aggregated reservoirs considered in the official model. The overall hydro generation installed capacity represents 70% of the system's installed capacity. We will consider 3 hydro plant owners: one price taker agent and two price maker agents. The 21 hydro plants are split into 3 groups of 7 plants. One group for each of the two price maker agents and the third group for the price taker agent. All thermal plants are considered individual price-taker agents. We considered a single load block for simplicity. The following simulations were carried out in a 5-year monthly horizon, that is, 60 stages. We considered 1000 sampled scenarios. The large-scale simulations were performed in a *PSRCloud* cluster of 8 servers, each one with 64 cores and 128GB of RAM. Each simulation took approximately 3 hours.

4.5.2 Market concentration analysis

In this section, we present 5 simulations with different market concentrations. We adapted the hydro plants so that we have different databases represented by tuples $(share_1\%, share_2\%)$ where $share_1$ and $share_2$ represent the percent shares of the hydro system belonging to each of the two price maker



(4.2(a))



(4.2(b))

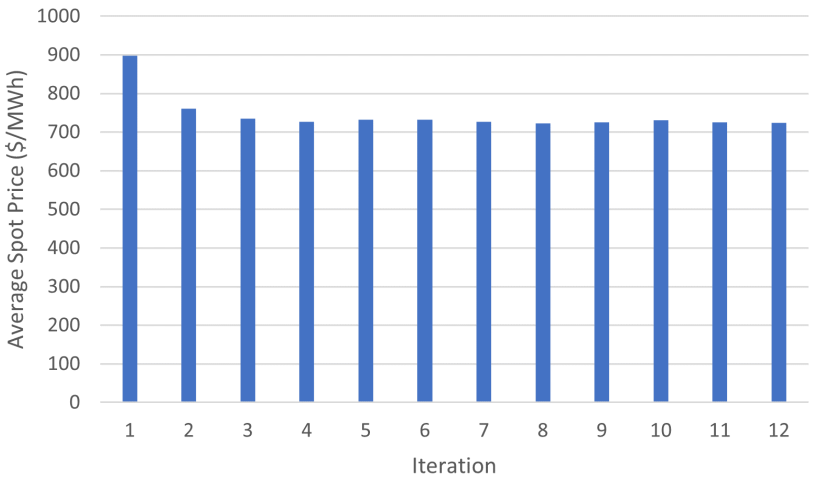
Figure 4.2: Results for simulations of Brazil's Southeast under different market concentrations with *no contracts*. In parentheses in the horizontal axis, we have the percent share of each price maker agent. Average values of Spot Prices, Normalized Revenue, Spillage, and Storage Level are with respect to all stages and scenarios. Spillage and Storage % are with respect to the maximum amount of water that can be stored in the system.

agents. The remainder of the hydro system is allocated to the price-taker hydro agent.

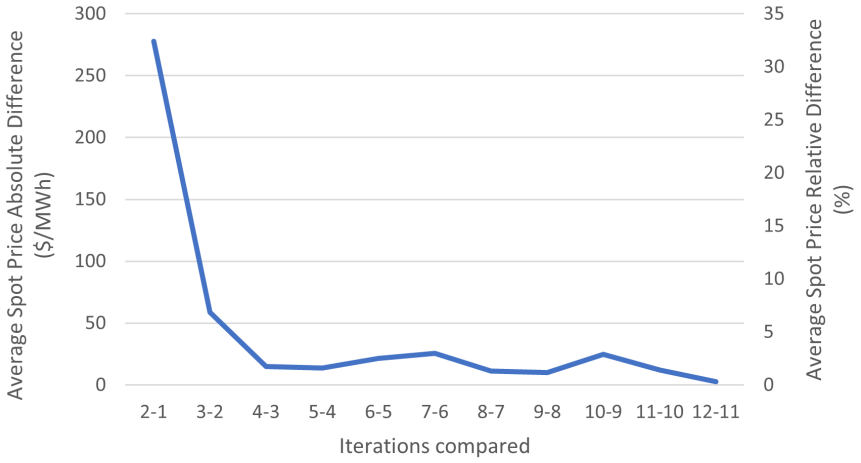
In Figure 4.2(a), we show the average spot prices as a function of the market distributions in a bar plot, we add the first bar with the spot price of the centralized dispatch. As expected, the spot prices rapidly grow as the concentration increases. The Figure also shows the average value of energy sold which is the total revenue of the agent divided by the total energy generated by the agent, which we will call captured price. We present results in such normalized forms so that we can compare the results of different-sized agents.

Figure 4.2(b) shows additional results with average reservoir levels of the price maker agents throughout the study period and the average spillage.

Figure 4.3(a) and Figure 4.3(b) present a typical convergence profile of



(4.3(a))



(4.3(b))

Figure 4.3: Convergence profile from Brazil's southeast with $(25\%, 50\%)$ and *no contracts*. See Figure 4.2(a). (a) Average spot prices of each iteration. (b) average spot price absolute and relative differences between two consecutive iterations: $(i+1)$–(i).

the proposed iterative method. These profiles were taken from the case study with price maker shares of $(25\%, 50\%)$ and no contracts. Figure 4.3(a) shows spot price averages for 12 consecutive iterations, the last value is the one reported in Figure 4.2(a). Figure 4.3(a) show both average absolute and relative differences between two consecutive iterations. First, absolute (or relative) differences are computed for each stage and scenario, and then, they are averaged.

4.5.3 Contract analyses

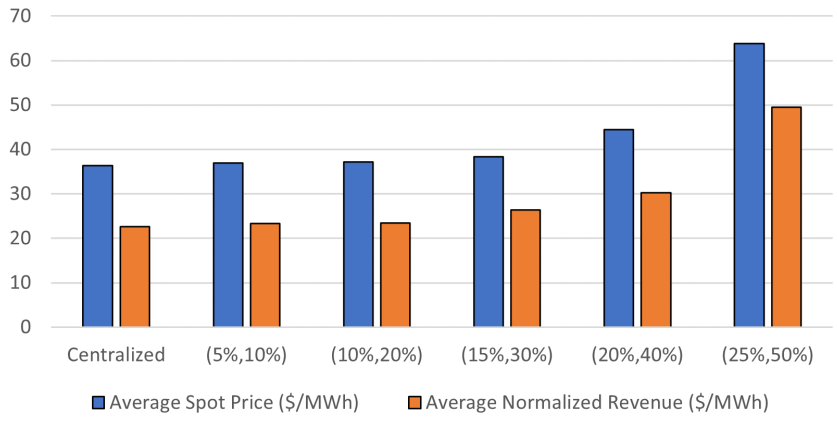
In this section, we repeat the above analyses but with contracts. To come up with monthly contract quantities, we take average generation values from the centralized dispatch and prices from the average spot prices in the centralized dispatch. This will stimulate the agents to produce energy and reduce the spot prices since being short in the contract together with high spot prices will lead the agent to have expenses due to the second term in (4-51).

Figures 4.4 and 4.5 contain the same metrics presented in the previous section but for the cases of agents that are 75% and 100% contracted, respectively. In the case studies, we can clearly see that the contracts completely eradicated the market power, and the resulting spot prices are very close to the ones obtained in the centralized dispatch. At the same time, the captured revenues and reservoir levels were also moved to values very close to the ones from the centralized dispatch.

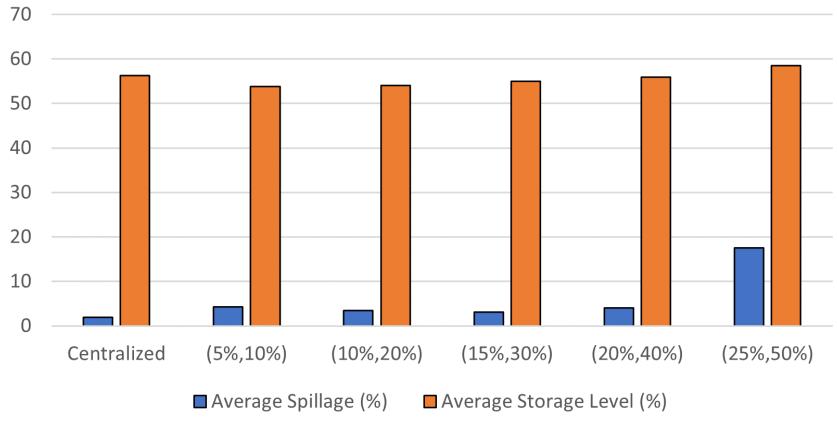
4.5.4 Brazilian system data

This database was created from the original data of the Brazilian system and contains 137 thermal plants representing 18% of the installed capacity, 364 renewable energy plants (wind and solar) representing 19% of the installed capacity and 32 hydropower plants representing the remaining 63% of the installed capacity. Here, we improve the load duration curve accuracy and consider 5 load blocks instead of just one to consider peak demand hours. Just like in the previous studies, we consider 5 years (60 monthly stages), 1000 scenarios and the same *PSRCloud* cluster configuration.

In this study, we consider an approximation of the real market concentration in the system. We represent 3 price maker agents, with respectively 32%, 9% and 7% of the energy resources representing the 3 largest power plant owners. Meanwhile, the remaining 55% of the resources are distributed among one purely hydro price taker agent and each thermal plant being yet another small price taker, representing all the smaller agents that are assumed to behave as price takers due to their comparatively reduced size. Each simulation took approximately 11 hours.



(4.4(a))



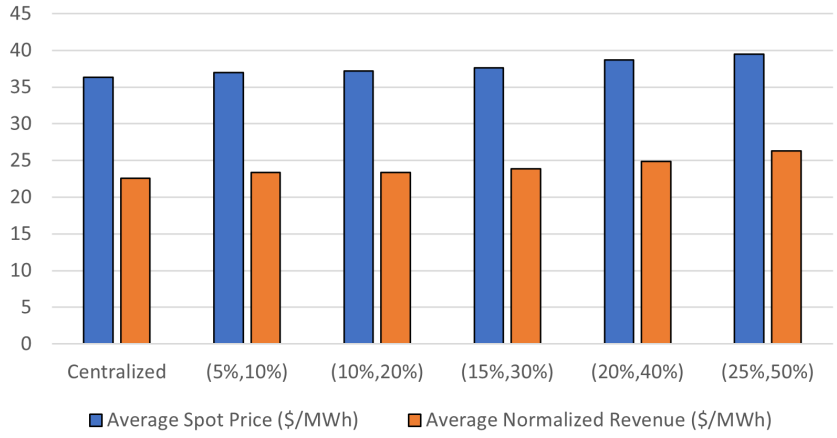
(4.4(b))

Figure 4.4: Results for simulations of Brazil's Southeast under different market concentrations with 75% *of contracting level*. In parentheses in the horizontal axis, we have the percent share of each price maker agent. Average values of Spot Prices, Normalized Revenue, Spillage, and Storage Level are with respect to all stages and scenarios. Spillage and Storage % are with respect to the maximum amount of water that can be stored in the system.

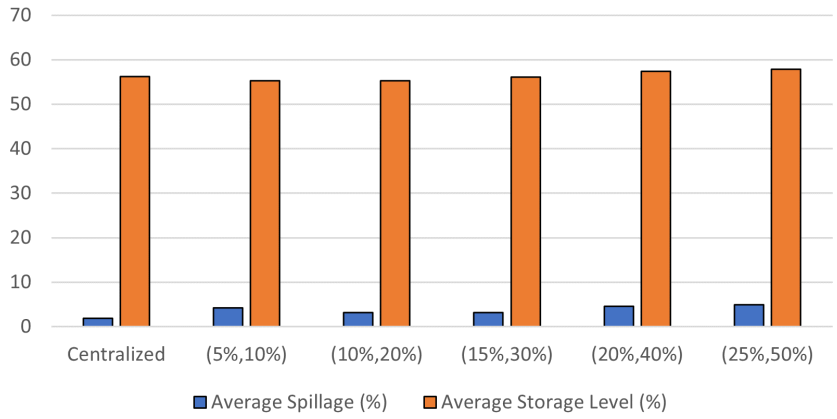
4.5.5 Brazilian system simulations

First, we simulated the centralized operation, the bid-based market with no contracts, then we considered agents with multiple contracting levels in the bid-based market. In particular, we analysed the cases of: 25%, 50%, 75% and 100% of contracting level. The key results from each of the 6 simulations are depicted in Figures 4.6 and 4.7.

Figure 4.6 shows the spot price under each of the above 6 conditions. We can clearly see the progression from the highest spot prices when there are no contracts in the bid-based market, allowing for agents to exercise their market power until the fully contracted case, in which the average spot price is



(4.5(a))



(4.5(b))

Figure 4.5: Results for simulations of Brazil's Southeast under different market concentrations with 100% *of contracting level*. In parentheses in the horizontal axis, we have the percent share of each price maker agent. Average values of Spot Prices, Normalized Revenue, Spillage, and Storage Level are with respect to all stages and scenarios. Spillage and Storage % are with respect to the maximum amount of water that can be stored in the system.

much closer to the one obtained from the centralized operation of the system. Given the number of resources allocated to the price maker agents, 45%, we note that a 75% contracting level is already very effective in reducing the gap to the centralized operation. This follows closely what we have seen in the simulations of the southeastern system. In particular, 75% was also very effective in containing market power in the case where the first price-maker agent had 15%, and the second price maker had 30%, of the resources resulting in 45% in the hands of price-maker agents. Finally, we note that even very low contracting levels (25%) make a difference in the final average spot prices.

Figure 4.7 presents the overall revenue, including operation and contract costs and spot and contract revenue throughout the 5-year horizon for each of

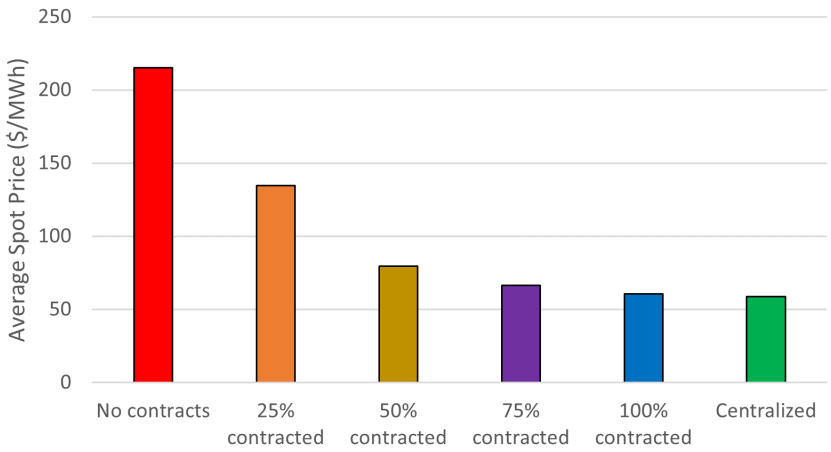


Figure 4.6: Average Spot Price ($/MWh) for the Brazilian system under different contracting levels. Averages with respect to stages and scenarios.

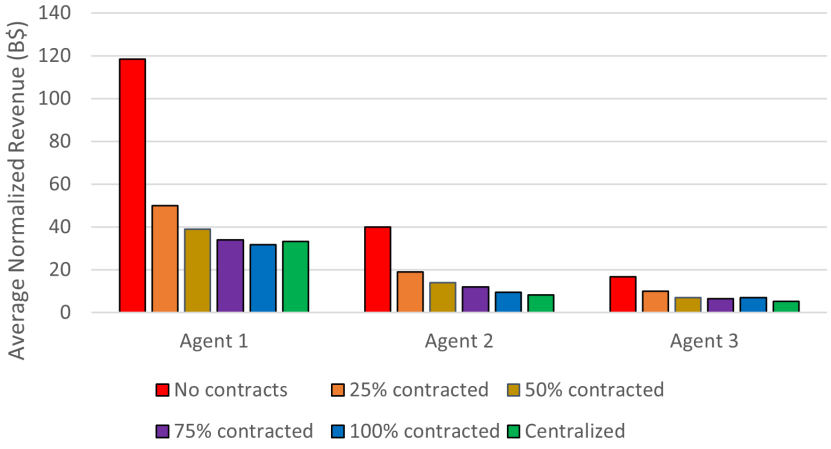


Figure 4.7: Average normalized revenue ($/MWh) for each of the 3 price maker agents under different contracting levels. Averages with respect to stages and scenarios.

the 3 price maker agents. Overall, the figure reproduces information already verified in Figure 4.6 with excessive revenues for all agents for the bid-based cases with no contracts. Notably, the 100% contracting case leads to revenues that are very similar to the ones from the centralized dispatch. Again, 75% seems to be an interesting value since it leaves the system very close to the centralized version in terms of agent remuneration.

4.6 Conclusion

Based on relevant works on long-term hydrothermal power markets, we combined three key pieces to develop a new and effective market simulator, namely: 1) SDDP applied to the centralized hydrothermal dispatch to initialize

and benchmark the process, 2) a multistage bilevel stochastic model for strategic bidding of a single-player to obtain decisions from price maker agents, and 3) an iterative diagonalization-based method to simulate the interactions among agents in the market. In contrast to the existing literature on the subject, with our new algorithm, we could jointly consider multiple reservoirs (32), multiple stages (60) and scenarios (1,000), and multiple price-maker agents (3). Therefore, a realistic large-scale case study based on the Brazilian power system (one of the largest in the world) could be addressed, and relevant insights could be obtained.

More specifically, we successfully performed multiple simulations of large-scale power systems. In a first sensitivity analysis in the Brazilian southeastern system, we could observe how the variation of market power concentration can lead to situations where market power is clearly detrimental to the market. In the follow-up analysis, we showed how contracts could be used to mitigate the market power leading to much more reasonable situations even at a 75% contracting level, and, at the 100% contracting level, the market power is completely eradicated. In the final study, we simulated a power system based on Brazil's real data, including all the most important generators of the system. We attempted to represent three large strategic agents in realistic proportions, given their current portfolios. The final results demonstrate that in the current setting of Brazilian market concentration, it is susceptible to market power. Finally, we showed that contracts could indeed be a very important tool to mitigate the market power and improve the market design towards the solution for social wealth maximization.

We highlight that a tool such as this can be extremely useful for market monitors. Simulating a power system to understand what might be happening is a first step towards understanding what is currently happening in an existing power system. Also, policy developers and market designers can profit from such tooling to experiment with alternative designs and come up with new solutions. On the algorithmic side, the method is amenable to parallel computing and is flexible to be extended to consider other physical and regulatory details. Notwithstanding, we showcase that a real-world problem formulated as a multistage bilevel stochastic problem can be solved under reasonable assumptions and provide relevant insights to decision-makers and regulators.

5
Conclusions and Future Directions

In this thesis, we presented a selection of 3 works developed during the Ph.D. program. Each of the works, in its own way, contributes to the state of the art of Bilevel Optimization. Moreover, they also contribute to the fields of Stochastic Optimization, Optimization Software and Power Systems. Contributions vary in a large range, including methodologies, algorithms, theoretical developments, open-source software, and case studies.

We applied bilevel optimization, stochastic programs and computational methods to important problems in the power systems industry. In applications (Chapters 3 and 4), we were especially interested in methodologies that can be applied to real-world problems, and, hence, we developed efficient methods that cleverly employ optimization techniques and parallel computing to tackle such problems. Due to our needs, we developed a package to model and solve bilevel programs that were made open-source.

Firstly, Chapter 2 presents the `BilevelJuMP.jl` open source package that extended the JuMP modeling language to handle bilevel optimization. The software implements both a modeling interface and solution techniques that allow users with different backgrounds to formulate and solve their first bilevel program or select specific methods that better perform for a given problem. The package also included innovative features such as the possibility of modeling lower-level conic programs or using different linearization techniques for each constraint.

The bilevel optimization software can be extended to handle more classes of problems, such as the ongoing developments on mixed integer lower level. Other fronts of development should include approximation algorithms that return good feasible solutions. Once the foundational package MathOptInterface is generalized so that nonlinear constraints are first class, `BilevelJuMP.jl` can be modified to accept non-linear constraints in the lower level.

Secondly, in Chapter 3 we presented a general application-driven learning algorithm that can be used in a broad range of applications. Besides proposing and mathematically describing the innovative framework, we developed both exact and heuristic algorithms. We highlight that the developments of Chapter 2 were not only fundamental during the experimentation process but also were

used in the exact method. Then we described an application of the framework to demand forecasting and reserve optimization. And we concluded the work by empirically demonstrating its properties and scalability in systems with thousands of buses.

In the future, application-driven learning should be applied to other problems both in and out of the power systems area. Exploring other cases for the demand forecast and reserve problem and using larger servers to improve results. Other algorithmic and theoretical developments can help the methodology achieve outstanding results. Finally, a package can be developed to facilitate the usage on the practitioner side.

Finally, in Chapter 4, we presented a methodology to simulate long-term bid-based hydro-thermal power systems. The methodology combines previous works and its main characteristic is its scalability. Which permits simulations of large-scale power systems in settings with multiple scenarios and stages. In particular, we applied the proposed methodology to the Brazilian system, one of the largest power systems in the world with large hydro penetration, considering a study horizon of 60 stages and 1000 scenarios.

The market simulator can be extended to consider many other constraints from real-world power systems and we can experiment with variants of the algorithms to improve performance and the quality of the results. Additional parallel schemes can also be used to improve performance. Applying the method to other power systems would allow the user to validate additional functionality.

We close our remarks by highlighting the power of bilevel optimization that enabled us to model extremely relevant power systems problems. Moreover, we both acknowledge the challenges posed by this class of optimization problems and we underline the importance of well-tested benchmark exact methods, approximate algorithms and high-performance computing that allowed us to achieve meaningful results.

Bibliography

[1] DIAS GARCIA, J.; BODIN, G. ; STREET, A.. **BilevelJuMP.jl: Modeling and Solving Bilevel Optimization in Julia**. arXiv preprint arXiv:2205.02307, 2022.

[2] DIAS GARCIA, J.; STREET, A.; HOMEM-DE MELLO, T. ; MUÑOZ, F. D.. **Application-driven learning via joint prediction and optimization of demand and reserves requirement**. arXiv preprint arXiv:2102.13273, 2021.

[3] DIAS GARCIA, J.; STREET, A. ; PEREIRA, M.. **Long-term hydrothermal bid-based market simulator with case studies in the brazilian system**. Unpublished, 2022.

[4] DE LUJÁN LATORRE, M.; GRANVILLE, S.. **The stackelberg equilibrium applied to ac power systems-a noninterior point algorithm**. Power Systems, IEEE Transactions on, 18(2):611–618, 2003.

[5] FRANCESCHI, L.; FRASCONI, P.; SALZO, S.; GRAZZI, R. ; PONTIL, M.. **Bilevel programming for hyperparameter optimization and meta-learning**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 1568–1577. PMLR, 2018.

[6] JEROSLOW, R. G.. **The polynomial hierarchy and a simple model for competitive analysis**. Mathematical programming, 32(2):146–164, 1985.

[7] WRIGHT, S. J.. **Optimization**. Encyclopedia Britannica. www.britannica.com/science/optimization. Accessed 1 July 2022., July 2021.

[8] DANTZIG, G. B.. **Linear programming**. Operations research, 50(1):42–47, 2002.

[9] DANTZIG, G. B.. **Linear programming and extensions**, volumen 48. Princeton university press, 1998.

[10] DANTZIG, G. B.; THAPA, M. N.. **Linear programming: Theory and extensions**, volumen 2. Springer, 2003.

[11] KANTOROVICH, L. V.. **Mathematical methods of organizing and planning production**. Management science, 6(4):366–422, 1960.

[12] FOURIER, J. B. J.. **Solution d'une question particuliere du calcul des inégalités**. Nouveau Bulletin des Sciences par la Société philomatique de Paris, 99:100, 1826.

[13] CHVATAL, V.. **Linear programming**. Macmillan, 1983.

[14] BERTSIMAS, D.; TSITSIKLIS, J. N.. **Introduction to linear optimization**, volumen 6. Athena Scientific Belmont, MA, 1997.

[15] BOYD, S.; BOYD, S. P. ; VANDENBERGHE, L.. **Convex optimization**. Cambridge university press, 2004.

[16] CALAFIORE, G. C.; EL GHAOUI, L.. **Optimization models**. Cambridge university press, 2014.

[17] BEN-TAL, A.; NEMIROVSKI, A.. **Lectures on modern convex optimization: analysis, algorithms, and engineering applications**. SIAM, 2001.

[18] BAZARAA, M. S.; SHERALI, H. D. ; SHETTY, C. M.. **Nonlinear programming: theory and algorithms**. John Wiley & Sons, 2013.

[19] LUENBERGER, D. G.; YE, Y. ; OTHERS. **Linear and nonlinear programming**, volumen 2. Springer, 1984.

[20] WOLSEY, L. A.; NEMHAUSER, G. L.. **Integer and combinatorial optimization**, volumen 55. John Wiley & Sons, 1999.

[21] KOCH, T.; BERTHOLD, T.; PEDERSEN, J. ; VANARET, C.. **Progress in mathematical programming solvers from 2001 to 2020**. EURO Journal on Computational Optimization, p. 100031, 2022.

[22] BIRGE, J. R.; LOUVEAUX, F.. **Introduction to stochastic programming**. Springer Science & Business Media, 2011.

[23] SHAPIRO, A.; PHILPOTT, A.. **A tutorial on stochastic programming**. Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html, 17, 2007.

[24] ZOU, J.; AHMED, S. ; SUN, X. A.. **Stochastic dual dynamic integer programming**. Mathematical Programming, 175(1-2):461–502, 2019.

[25] SHAPIRO, A.. **Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming**. European Journal of Operational Research, 288(1):1–13, 2021.

[26] PFLUG, G. C.; PICHLER, A.. **A distance for multistage stochastic optimization models**. SIAM Journal on Optimization, 22(1):1–23, 2012.

[27] BEN-TAL, A.; EL GHAOUI, L. ; NEMIROVSKI, A.. **Robust optimization**, volumen 28. Princeton university press, 2009.

[28] NEMIROVSKI, A.; SHAPIRO, A.. **Scenario approximations of chance constraints**. Probabilistic and randomized methods for design under uncertainty, p. 3–47, 2006.

[29] DEMPE, S.. **Foundations of bilevel programming**. Springer Science & Business Media, 2002.

[30] BARD, J. F.. **Practical bilevel optimization: algorithms and applications**, volumen 30. Springer Science & Business Media, 2013.

[31] BECK, Y.; SCHMIDT, M.. **A gentle and incomplete introduction to bilevel optimization**. Technical report, Lecture Notes, url: http://www. optimization-online. org/DB_HTML/2021/06/8450. html, 2021.

[32] DEMPE, S.. **Bilevel optimization: theory, algorithms and applications**. TU Bergakademie Freiberg, Fakultät für Mathematik und Informatik, 2018.

[33] BECK, Y.; LJUBIC, I. ; SCHMIDT, M.. **A survey on bilevel optimization under uncertainty**. 2022.

[34] AMPL. **AMPL complementarity problems**. `https://ampl.com/products/solvers/all-solvers-for-ampl/`, 2021. Accessed: 2021-04-17.

[35] GAMS DEVELOPMENT CORPORATION. **General Algebraic Modeling System (GAMS)**. `https://www.gams.com/latest/docs/UG_MAIN.html`, 2022. Accessed: 2022-07-04.

[36] HART, W. E.; LAIRD, C. D.; WATSON, J.-P.; WOODRUFF, D. L.; HACKEBEIL, G. A.; NICHOLSON, B. L. ; SIIROLA, J. D.. **Pyomo-optimization modeling in python**, volumen 67. Springer, 2017.

[37] LUBIN, M.; DUNNING, I.. **Computing in operations research using julia**. INFORMS Journal on Computing, 27(2):238–248, 2015.

[38] DUNNING, I.; HUCHETTE, J. ; LUBIN, M.. **Jump: A modeling language for mathematical optimization**. SIAM Review, 59(2):295–320, 2017.

[39] LUBIN, M.; DOWSON, O.; GARCIA, J. D.; HUCHETTE, J.; LEGAT, B. ; VIELMA, J. P.. **Jump 1.0**. arXiv preprint arXiv:2206.03866, 2022.

[40] VAN ROSSUM, G.; DRAKE, F. L.. **Python 3 Reference Manual**. CreateSpace, Scotts Valley, CA, 2009.

[41] BEZANSON, J.; EDELMAN, A.; KARPINSKI, S. ; SHAH, V. B.. **Julia: A fresh approach to numerical computing**. SIAM review, 59(1):65–98, 2017.

[42] GUROBI OPTIMIZATION, L.. **Gurobi optimizer reference manual**, 2020.

[43] FICO. **FICO Xpress Optimizer Reference Manual**, 2021.

[44] CONEJO, A. J.; BARINGO, L.. **Power system operations**, volumen 11. Springer, 2018.

[45] CHOWDHURY, B. H.; RAHMAN, S.. **A review of recent advances in economic dispatch**. IEEE transactions on power systems, 5(4):1248–1259, 1990.

[46] ANJOS, M. F.; CONEJO, A. J. ; OTHERS. **Unit commitment in electric energy systems**. Foundations and Trends® in Electric Energy Systems, 1(4):220–310, 2017.

[47] PADHY, N. P.. **Unit commitment-a bibliographical survey**. IEEE Transactions on power systems, 19(2):1196–1205, 2004.

[48] HONG, T.; FAN, S.. **Probabilistic electric load forecasting: A tutorial review**. International Journal of Forecasting, 32(3):914–938, 2016.

[49] FRANK, S.; STEPONAVICE, I. ; REBENNACK, S.. **Optimal power flow: a bibliographic survey i**. Energy systems, 3(3):221–258, 2012.

[50] TAYLOR, J. A.. **Convex optimization of power systems**. Cambridge University Press, 2015.

[51] MOLZAHN, D. K.; HISKENS, I. A. ; OTHERS. **A survey of relaxations and approximations of the power flow equations**. Foundations and Trends® in Electric Energy Systems, 4(1-2):1–221, 2019.

[52] ROALD, L.; POZO, D.; PAPAVASILIOU, A.; MOLZHAN, D.; KAZEMPOUR, J. ; CONEJO, A.. **Power systems optimization under uncertainty: A review of methods and applications**. 22nd Power Systems Computation Conference, 2022.

[53] CHEN, Y.; GRIBIK, P. ; GARDNER, J.. **Incorporating post zonal reserve deployment transmission constraints into energy and ancillary service co-optimization**. IEEE Transactions on Power Systems, 29(2):537–549, 2013.

[54] DE VOS, K.; STEVENS, N.; DEVOLDER, O.; PAPAVASILIOU, A.; HEBB, B. ; MATTHYS-DONNADIEU, J.. **Dynamic dimensioning approach for operating reserves: Proof of concept in belgium**. Energy policy, 124:272–285, 2019.

[55] PEREIRA, M.; PINTO, L.. **Stochastic optimization of a multireservoir hydroelectric system: a decomposition approach**. Water resources research, 21(6):779–792, 1985.

[56] STEEGER, G.; BARROSO, L. A. ; REBENNACK, S.. **Optimal Bidding Strategies for Hydro-Electric Producers: A Literature Survey**. IEEE Transactions on Power Systems, 29(4):1758–1766, 2014.

[57] PEREIRA, M.; GRANVILLE, S.; FAMPA, M.; DIX, R. ; BARROSO, L.. **Strategic bidding under uncertainty: a binary expansion approach**. IEEE Transactions on Power Systems, 20(1):180–188, 2005.

[58] VARIAN, H. R.. **Intermediate microeconomics: a modern approach: ninth international student edition**. WW Norton & Company, 2014.

[59] BENGIO, Y.. **Using a financial training criterion rather than a prediction criterion**. International Journal of Neural Systems, 8(04):433–443, 1997.

[60] SHARMA, A.; BESANÇON, M.; GARCIA, J. D. ; LEGAT, B.. **Flexible differentiable optimization via model transformations**. arXiv preprint arXiv:2206.06135, 2022.

[61] SOUTO, M.; GARCIA, J. D. ; VEIGA, Á.. **Exploiting low-rank structure in semidefinite programming by approximate operator splitting**. Optimization, 71(1):117–144, 2022.

[62] SOARES, A.; STREET, A.; ANDRADE, T. ; GARCIA, J. D.. **An integrated progressive hedging and benders decomposition with multiple master method to solve the brazilian generation expansion problem**. IEEE Transactions on Power Systems, 2022.

[63] LEGAT, B.; DOWSON, O.; GARCIA, J. D. ; LUBIN, M.. **Mathoptinterface: a data structure for mathematical optimization problems**. INFORMS Journal on Computing, 34(2):672–689, 2022.

[64] ROSEMBERG, A. W.; STREET, A.; GARCIA, J. D.; VALLADÃO, D. M.; SILVA, T. ; DOWSON, O.. **Assessing the cost of network simplifications in long-term hydrothermal dispatch planning models**. IEEE Transactions on Sustainable Energy, 13(1):196–206, 2021.

[65] DA COSTA, L. C.; THOMÉ, F. S.; GARCIA, J. D. ; PEREIRA, M. V.. **Reliability-constrained power system expansion planning: A stochastic risk-averse optimization approach**. IEEE Transactions on Power Systems, 36(1):97–106, 2020.

[66] ROSEMBERG, A. W.; STREET, A.; GARCIA, J. D.; SILVA, T.; VALLADÃO, D. M. ; DOWSON, O.. **HydroPowerModels. jl: A julia/jump package for hydrothermal economic dispatch optimization**. In: PROCEEDINGS OF THE JULIACON CONFERENCES, volumen 1, p. 35, 2020.

[67] VON STACKELBERG, H.. **The theory of the market economy**. Oxford University Press, 1952.

[68] LABBÉ, M.; MARCOTTE, P. ; SAVARD, G.. **A bilevel model of taxation and its application to optimal highway pricing**. Management science, 44(12-part-1):1608–1622, 1998.

[69] POZO, D.; SAUMA, E. ; CONTRERAS, J.. **Basic theoretical foundations and insights on bilevel models and their applications to power systems**. Annals of Operations Research, 254(1-2):303–334, 2017.

[70] BRACKEN, J.; MCGILL, J. T.. **Defense applications of mathematical programs with optimization problems in the constraints**. Operations Research, 22(5):1086–1096, 1974.

[71] KÜÇÜKAYDIN, H.; ARAS, N. ; ALTINEL, I. K.. **Competitive facility location problem with attractiveness adjustment of the follower: A bilevel programming model and its solution**. European Journal of Operational Research, 208(3):206–220, 2011.

[72] DEMPE, S.; KALASHNIKOV, V.; PÉREZ-VALDÉS, G. A. ; KALASHNYKOVA, N.. **Bilevel programming problems**. Energy Systems. Springer, Berlin, 2015.

[73] VICENTE, L. N.; CALAMAI, P. H.. **Bilevel and multilevel programming: A bibliography review**. Journal of Global optimization, 5(3):291–306, 1994.

[74] COLSON, B.; MARCOTTE, P. ; SAVARD, G.. **An overview of bilevel optimization**. Annals of operations research, 153(1):235–256, 2007.

[75] KALASHNIKOV, V. V.; DEMPE, S.; PÉREZ-VALDÉS, G. A.; KALASHNYKOVA, N. I. ; CAMACHO-VALLEJO, J.-F.. **Bilevel programming and applications**. Mathematical Problems in Engineering, 2015, 2015.

[76] BIALAS, W.; KARWAN, M.. **On two-level optimization**. IEEE transactions on automatic control, 27(1):211–214, 1982.

[77] KIM, Y.; LEYFFER, S. ; MUNSON, T.. **Mpec methods for bilevel optimization problems**. In: BILEVEL OPTIMIZATION, p. 335–360. Springer, 2020.

[78] BARD, J. F.. **Convex two-level optimization**. Mathematical programming, 40(1-3):15–27, 1988.

[79] HANSEN, P.; JAUMARD, B. ; SAVARD, G.. **New branch-and-bound rules for linear bilevel programming**. SIAM Journal on scientific and Statistical Computing, 13(5):1194–1217, 1992.

[80] FORTUNY-AMAT, J.; MCCARL, B.. **A representation and economic interpretation of a two-level programming problem**. Journal of the operational Research Society, 32(9):783–792, 1981.

[81] SIDDIQUI, S.; GABRIEL, S. A.. **An sos1-based approach for solving mpecs with a natural gas market application**. Networks and Spatial Economics, 13(2):205–227, 2013.

[82] FLETCHER, R.; LEYFFER, S.. **Solving mathematical programs with complementarity constraints as nonlinear programs**. Optimization Methods and Software, 19(1):15–40, 2004.

[83] SCHOLTES, S.. **Convergence properties of a regularization scheme for mathematical programs with complementarity constraints**. SIAM Journal on Optimization, 11(4):918–936, 2001.

[84] RALPH, D.; WRIGHT, S. J.. **Some properties of regularization and penalization schemes for mpecs**. Optimization Methods and Software, 19(5):527–556, 2004.

[85] NOCEDAL, J.. **Knitro: An integrated package for nonlinear optimization**. In: LARGE-SCALE NONLINEAR OPTIMIZATION, p. 35–60. Springer, 2006.

[86] FLETCHER, R.; LEYFFER, S.. **filterMPEC neos solvers**. `https://neos-server.org/neos/solvers/mpec:filterMPEC/AMPL.html`, 2021. Accessed: 2021-04-17.

[87] FERRIS, M.; GAMS. **NLPEC - GAMS solvers**. `https://www.gams.com/33/docs/S_NLPEC.html`, 2021. Accessed: 2021-04-17.

[88] PINEDA, S.; BYLLING, H. ; MORALES, J.. **Efficiently solving linear bilevel programming problems using off-the-shelf optimization software**. Optimization and Engineering, 19(1):187–211, 2018.

[89] FALK, J. E.; LIU, J.. **On bilevel programming, part i: general nonlinear cases**. Mathematical Programming, 70(1-3):47–72, 1995.

[90] FAMPA, M. H.; MELO, W. A. ; MACULAN, N.. **Semidefinite relaxation for linear programs with equilibrium constraints**. International Transactions in Operational Research, 20(2):201–212, 2013.

[91] AIYOSHI, E.; SHIMIZU, K.. **A solution method for the static constrained stackelberg problem via penalty method**. IEEE Transactions on Automatic Control, 29(12):1111–1114, 1984.

[92] WHITE, D. J.; ANANDALINGAM, G.. **A penalty function approach for solving bi-level linear programs**. Journal of Global Optimization, 3(4):397–419, 1993.

[93] KLEINERT, T.; SCHMIDT, M.. **Computing feasible points of bilevel problems with a penalty alternating direction method**. INFORMS Journal on Computing, 2020.

[94] BYEON, G.; VAN HENTENRYCK, P.. **Benders subproblem decomposition for discrete-continuous bilevel problems**. arXiv preprint arXiv:1902.04375, 2019.

[95] WU, S.; MARCOTTE, P. ; CHEN, Y.. **A cutting plane method for linear bilevel programs**. Systems Science and Mathematical Sciences, 11:125–133, 1998.

[96] FISCHETTI, M.; LJUBIĆ, I.; MONACI, M. ; SINNL, M.. **A new general-purpose algorithm for mixed-integer bilevel linear programs**. Operations Research, 65(6):1615–1637, 2017.

[97] TAHERNEJAD, S.; RALPHS, T. K. ; DENEGRE, S. T.. **A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation**. Mathematical Programming Computation, 12(4):529–568, 2020.

[98] RALPHS, T.; TAHERNEJAD, S.; VIGERSKE, S. ; BESANÇON, M.. **coin-or/mibs: Version 1.1.3**, Dec. 2019.

[99] FISCHETTI, M.; LJUBIĆ, I.; MONACI, M. ; SINNL, M.. **Bilevel integer programming and interdiction problems - solver for mixed-integer bilevel linear problems**. `https://msinnl.github.io/pages/bilevel.html`, 2017. Accessed: 2021-04-17.

[100] KOLSTAD, C. D.; LASDON, L. S.. **Derivative evaluation and computational experience with large bilevel mathematical programs**. Journal of optimization theory and applications, 65(3):485–499, 1990.

[101] VICENTE, L.; SAVARD, G. ; JÚDICE, J.. **Descent approaches for quadratic bilevel programming**. Journal of Optimization Theory and Applications, 81(2):379–399, 1994.

[102] BARD, J. F.. **An efficient point algorithm for a linear two-stage optimization problem**. Operations Research, 31(4):670–684, 1983.

[103] SINHA, A.; MALO, P. ; DEB, K.. **A review on bilevel optimization: From classical to evolutionary approaches and applications**. IEEE Transactions on Evolutionary Computation, 22(2):276–295, 2017.

[104] GAMS EMP. **Bilevel programs**. `https://www.gams.com/latest/docs/UG_EMP_Bilevel.html`, 2021. Accessed: 2021-04-17.

[105] FERRIS, M. C.; DIRKSE, S. P.; JAGLA, J.-H. ; MEERAUS, A.. **An extended mathematical programming framework**. Computers & Chemical Engineering, 33(12):1973–1982, 2009.

[106] LOFBERG, J.. **Yalmip: A toolbox for modeling and optimization in matlab**. In: 2004 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (IEEE CAT. NO. 04CH37508), p. 284–289. IEEE, 2004.

[107] YALMIP. **YALMIP bilevel programming**. https://yalmip.github.io/tutorial/bilevelprogramming/, 2021. Accessed: 2021-04-17.

[108] HART, W.; CASTILLO, A.. **PAO**. [Computer Software] https://doi.org/10.11578/dc.20201105.1, dec 2019.

[109] YUE, D.; GAO, J.; ZENG, B. ; YOU, F.. **A projection-based reformulation and decomposition algorithm for global optimization of a class of mixed integer bilevel linear programs**. Journal of Global Optimization, 73(1):27–57, 2019.

[110] BESANÇON, M.. **A julia package for bilevel optimization problems**. Journal of Open Source Software, 4(39):1278, 7 2019.

[111] DUNNING, I.; HUCHETTE, J. ; LUBIN, M.. **JuMP: A modeling language for mathematical optimization**. SIAM Review, 59(2):295–320, 2017.

[112] AIMMS. **AIMMS complementarity problems**. https://documentation.aimms.com/language-reference/optimization-modeling-components/mixed-complementarity-problems/complementarity-problems.html, 2021. Accessed: 2021-04-17.

[113] LEGAT, B.; DOWSON, O.; GARCIA, J. D. ; LUBIN, M.. **Mathoptinterface: a data structure for mathematical optimization problems**. arXiv preprint arXiv:2002.03447, 2020.

[114] BYLLING, H. C.; GABRIEL, S. A. ; BOOMSMA, T. K.. **A parametric programming approach to bilevel optimisation with lower-level variables in the upper level**. Journal of the Operational Research Society, 71(5):846–865, 2020.

[115] DOWSON, O.; KAPELEVICH, L.. **SDDP.jl: a julia package for stochastic dual dynamic programming**. INFORMS Journal on Computing, 33(1):27–33, 2021.

[116] LEGAT, B.; WEISSER, T.; KAPELEVICH, L.; HUCHETTE, J. ; CONTRIBUTORS. **jump-dev/sumofsquares.jl: v0.4.6**, Apr. 2021.

[117] PULSIPHER, J. L.; ZHANG, W.; HONGISTO, T. J. ; ZAVALA, V. M.. **A unifying modeling abstraction for infinite-dimensional optimization**. arXiv preprint arXiv:2106.12689, 2021.

[118] BESTUZHEVA, K.; BESANÇON, M.; CHEN, W.-K.; CHMIELA, A.; DONKIEWICZ, T.; VAN DOORNMALEN, J.; EIFLER, L.; GAUL, O.; GAMRATH, G.; GLEIXNER, A.; GOTTWALD, L.; GRACZYK, C.; HALBIG, K.; HOEN, A.; HOJNY, C.; VAN DER HULST, R.; KOCH, T.; LÜBBECKE, M.; MAHER, S. J.; MATTER, F.; MÜHMER, E.; MÜLLER, B.; PFETSCH, M. E.; REHFELDT, D.; SCHLEIN, S.; SCHLÖSSER, F.; SERRANO, F.; SHINANO, Y.; SOFRANAC, B.; TURNER, M.; VIGERSKE, S.; WEGSCHEIDER, F.; WELLNER, P.; WENINGER, D. ; WITZIG, J.. **The SCIP Optimization Suite 8.0**. Technical report, Optimization Online, December 2021.

[119] BODIN, G.; GARCIA, J. D.; LEGAT, B.; BESANÇON, M.; LUBIN, M.; DOWSON, O. ; CONTRIBUTORS. **jump-dev/Dualization.jl: v0.3.4**, Apr. 2021.

[120] VANDENBERGHE, L.. **The cvxopt linear and quadratic cone program solvers**. Online: http://cvxopt. org/documentation/coneprog. pdf, 2010.

[121] KLEINERT, T.; SCHMIDT, M.. **Why there is no need to use a big-m in linear bilevel optimization: A computational study of two ready-to-use approaches**. Technical report, Tech. rep. url: http://www. optimization-online. org/DB_HTML/2020/10/8065. html, 2020.

[122] BEALE, E. M. L.; TOMLIN, J. A.. **Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables**. OR, 69(447-454):99, 1970.

[123] BELOTTI, P.; BONAMI, P.; FISCHETTI, M.; LODI, A.; MONACI, M.; NOGALES-GÓMEZ, A. ; SALVAGNIN, D.. **On handling indicator constraints in mixed integer programming**. Computational Optimization and Applications, 65(3):545–566, 2016.

[124] PINEDA, S.; MORALES, J. M.. **Solving linear bilevel problems using big-ms: Not all that glitters is gold**. IEEE Transactions on Power Systems, 34(3):2469–2471, 2019.

[125] KLEINERT, T.; LABBÉ, M.; PLEIN, F. A. ; SCHMIDT, M.. **There's no free lunch: on the hardness of choosing a correct big-m in bilevel optimization**. Operations research, 68(6):1716–1721, 2020.

[126] ANDRADE, T.; OLIVEIRA, F.; HAMACHER, S. ; EBERHARD, A.. **Enhancing the normalized multiparametric disaggregation tech-**

nique for mixed-integer quadratic programming. Journal of Global Optimization, 73(4):701–722, 2019.

[127] DIAS GARCIA, J.. **joaquimg/QuadraticToBinary.jl: v0.2.4**, Apr. 2021.

[128] PEREIRA, M. V.; GRANVILLE, S.; FAMPA, M. H.; DIX, R. ; BARROSO, L. A.. **Strategic bidding under uncertainty: a binary expansion approach**. IEEE Transactions on Power Systems, 20(1):180–188, 2005.

[129] ZARE, M. H.; BORRERO, J. S.; ZENG, B. ; PROKOPYEV, O. A.. **A note on linearized reformulations for a class of bilevel linear integer problems**. Annals of Operations Research, 272(1-2):99–117, 2019.

[130] KUNISCH, K.; POCK, T.. **A bilevel optimization approach for parameter learning in variational models**. SIAM Journal on Imaging Sciences, 6(2):938–983, 2013.

[131] MACKAY, M.; VICOL, P.; LORRAINE, J.; DUVENAUD, D. ; GROSSE, R.. **Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions**. arXiv preprint arXiv:1903.03088, 2019.

[132] BENNETT, K. P.; HU, J.; JI, X.; KUNAPULI, G. ; PANG, J.-S.. **Model selection via bilevel optimization**. In: THE 2006 IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORK PROCEEDINGS, p. 1922–1929. IEEE, 2006.

[133] IBM. **CPLEX Optimizer User's Manual**, 2021.

[134] HUANGFU, Q.; HALL, J. A. J.. **Parallelizing the dual revised simplex method**. Mathematical Programming Computation, 10(1):119–142, 2018.

[135] WÄCHTER, A.; BIEGLER, L. T.. **On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming**. Mathematical programming, 106(1):25–57, 2006.

[136] FANZERES, B.; AHMED, S. ; STREET, A.. **Robust strategic bidding in auction-based markets**. European Journal of Operational Research, 272(3):1158–1172, 2019.

[137] KLEINERT, T.; LABBÉ, M.; PLEIN, F. ; SCHMIDT, M.. **Closing the gap in linear bilevel optimization: a new valid primal-dual inequality**. Optimization Letters, 15(4):1027–1040, 2021.

[138] CHI, X.; WAN, Z. ; HAO, Z.. **The models of bilevel programming with lower level second-order cone programs**. Journal of Inequalities and Applications, 2014(1):1–23, 2014.

[139] ELA, E.; MILLIGAN, M. ; KIRBY, B.. **Operating reserves and variable generation**. Technical Report NREL/TP-5500-51978, National Renewable Energy Lab. (NREL), Golden, CO (United States), 2011.

[140] WANG, B.; HOBBS, B. F.. **A flexible ramping product: Can it help real-time dispatch markets approach the stochastic dispatch ideal?** Electric Power Systems Research, 109:128–140, 2014.

[141] PAPAVASILIOU, A.; OREN, S. S. ; ROUNTREE, B.. **Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration**. IEEE Transactions on Power Systems, 30(3):1109–1120, 2014.

[142] PAPAVASILIOU, A.; OREN, S. S.. **Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network**. Operations Research, 61(3):578–592, 2013.

[143] WANG, B.; HOBBS, B. F.. **Real-time markets for flexiramp: A stochastic unit commitment-based analysis**. IEEE Transactions on Power Systems, 31(2):846–860, 2015.

[144] PJM. **PJM Manual 11 : Energy and Ancillary Services Market Operations**. (97):200, 2018.

[145] CAISO. **2019 Annual Report on Market Issues and Performance. Department of Market Monitoring, California Independent System Operator**, 2020.

[146] BERTSIMAS, D.; KALLUS, N.. **From predictive to prescriptive analytics**. Management Science, 2019.

[147] ROCKAFELLAR, R. T.; URYASEV, S. ; ZABARANKIN, M.. **Risk tuning with generalized linear regression**. Mathematics of Operations Research, 33(3):712–729, 2008.

[148] ZELLNER, A.. **Bayesian estimation and prediction using asymmetric loss functions**. Journal of the American Statistical Association, 81(394):446–451, 1986.

[149] ZELLNER, A.. **Biased predictors, rationality and the evaluation of forecasts**. Economics Letters, 21(1):45–48, 1986.

[150] VARIAN, H.. **A bayesian approach to real estate assessment**. In: Fienberg, S. E.; Zellner, A., editors, STUDIES IN BAYESIAN ECONOMETRICS AND STATISTICS IN HONOR OF LEONARD J. SAVAGE. North-Holland, Amsterdam, 1975.

[151] ELMACHTOUB, A. N.; GRIGAS, P.. **Smart "predict, then optimize"**. Management Science, 2021.

[152] MUÑOZ, M. A.; PINEDA, S. ; MORALES, J. M.. **A bilevel framework for decision-making under uncertainty with contextual information**. Omega, 108:102575, 2022.

[153] SHAPIRO, A.; DENTCHEVA, D. ; RUSZCZYŃSKI, A.. **Lectures on stochastic programming : modeling and theory**. SIAM, 2nd edition, 2014.

[154] THE EUROPEAN COMMISSION. **Commission regulation (eu) 2017/1485 - establishing a guideline on electricity transmission system operation**, August 2017. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32017R1485&from=EN.

[155] GHOSN, J.; BENGIO, Y.. **Multi-task learning for stock selection**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 946–952, 1997.

[156] GARCIA, R.; GENÇAY, R.. **Pricing and hedging derivative securities with neural networks and a homogeneity hint**. Journal of Econometrics, 94(1-2):93–115, 2000.

[157] KAO, Y.-H.; ROY, B. V. ; YAN, X.. **Directed regression**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 889–897, 2009.

[158] DONTI, P.; AMOS, B. ; KOLTER, J. Z.. **Task-based end-to-end model learning in stochastic optimization**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 5484–5494, 2017.

[159] WILDER, B.; DILKINA, B. ; TAMBE, M.. **Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization**. In: PROCEEDINGS OF THE AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, volumen 33, p. 1658–1665, 2019.

[160] EL BALGHITI, O.; ELMACHTOUB, A.; GRIGAS, P. ; TEWARI, A.. **Generalization bounds in the predict-then-optimize framework**. In:

ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 14389–14398, 2019.

[161] MANDI, J.; STUCKEY, P. J.; GUNS, T. ; OTHERS. **Smart predict-and-optimize for hard combinatorial optimization problems**. In: PROCEEDINGS OF THE AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, volumen 34, p. 1603–1610, 2020.

[162] RYZHOV, I. O.; POWELL, W. B.. **Information collection for linear programs with uncertain objective coefficients**. SIAM Journal on Optimization, 22(4):1344–1368, 2012.

[163] DEN BOER, A. V.; SIERAG, D. D.. **Decision-based model selection**. European Journal of Operational Research, 290(2):671–686, 2021.

[164] BEZERRA, B.; VEIGA, Á.; BARROSO, L. A. ; PEREIRA, M.. **Stochastic long-term hydrothermal scheduling with parameter uncertainty in autoregressive streamflow models**. IEEE Transactions on Power Systems, 32(2):999–1006, 2016.

[165] SEN, S.; DENG, Y.. **Learning enabled optimization: Towards a fusion of statistical learning and stochastic programming**. Optimization Online, 2018.

[166] DENG, Y.; LIU, J. ; SEN, S.. **Coalescing data and decision sciences for analytics**. In: INFORMS TutORials IN OPERATIONS RESEARCH, p. 20–49. INFORMS, 2018.

[167] LIU, J.; LI, G. ; SEN, S.. **Coupled learning enabled stochastic programming with endogenous uncertainty**. Mathematics of Operations Research, 47(2):1681–1705, 2022.

[168] POWELL, W. B.; FRAZIER, P.. **Optimal learning**. In: INFORMS TutORials IN OPERATIONS RESEARCH, p. 213–246. Informs, 2008.

[169] POWELL, W. B.; RYZHOV, I. O.. **Optimal learning**, volumen 841. John Wiley & Sons, 2012.

[170] POWELL, W. B.. **The knowledge gradient for optimal learning**. Wiley Encyclopedia of Operations Research and Management Science, 2010.

[171] FRAZIER, P. I.. **Bayesian optimization**. In: INFORMS TutORials IN OPERATIONS RESEARCH, p. 255–278. Informs, 2018.

[172] VAN DER MEER, D. W.; WIDÉN, J. ; MUNKHAMMAR, J.. **Review on probabilistic forecasting of photovoltaic power production and electricity consumption**. Renewable and Sustainable Energy Reviews, 81:1484–1512, 2018.

[173] ORWIG, K. D.; AHLSTROM, M. L.; BANUNARAYANAN, V.; SHARP, J.; WILCZAK, J. M.; FREEDMAN, J.; HAUPT, S. E.; CLINE, J.; BARTHOLOMY, O.; HAMANN, H. F. ; OTHERS. **Recent trends in variable generation forecasting and its value to the power system**. IEEE Transactions on Sustainable Energy, 6(3):924–933, 2014.

[174] SWEENEY, C.; BESSA, R. J.; BROWELL, J. ; PINSON, P.. **The future of forecasting for renewable energy**. Wiley Interdisciplinary Reviews: Energy and Environment, 9(2):e365, 2020.

[175] HOLTTINEN, H.; MILLIGAN, M.; ELA, E.; MENEMENLIS, N.; DOBSCHINSKI, J.; RAWN, B.; BESSA, R. J.; FLYNN, D.; GOMEZ-LAZARO, E. ; DETLEFSEN, N. K.. **Methodologies to determine operating reserves due to increased wind power**. IEEE Transactions on Sustainable Energy, 3(4):713–723, 2012.

[176] STRBAC, G.; SHAKOOR, A.; BLACK, M.; PUDJIANTO, D. ; BOPP, T.. **Impact of wind generation on the operation and development of the uk electricity systems**. Electric Power Systems Research, 77(9):1214–1227, 2007.

[177] BUCKSTEEG, M.; NIESEN, L. ; WEBER, C.. **Impacts of dynamic probabilistic reserve sizing techniques on reserve requirements and system costs**. IEEE Transactions on Sustainable Energy, 7(4):1408–1420, 2016.

[178] ZHENG, Q. P.; WANG, J. ; LIU, A. L.. **Stochastic optimization for unit commitment—a review**. IEEE Transactions on Power Systems, 30(4):1913–1924, 2014.

[179] ARAVENA, I.; PAPAVASILIOU, A.. **Asynchronous lagrangian scenario decomposition**. Mathematical Programming Computation, p. 1–50, 2020.

[180] KNUEVEN, B.; OSTROWSKI, J. ; WATSON, J.-P.. **On mixed-integer programming formulations for the unit commitment problem**. INFORMS Journal on Computing, 32(4):857–876, 2020.

[181] GADE, D.; HACKEBEIL, G.; RYAN, S. M.; WATSON, J.-P.; WETS, R. J.-B. ; WOODRUFF, D. L.. **Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs**. Mathematical Programming, 157(1):47–67, 2016.

[182] KAZEMPOUR, J.; PINSON, P. ; HOBBS, B. F.. **A stochastic market design with revenue adequacy and cost recovery by scenario: Benefits and costs**. IEEE Transactions on Power Systems, 33(4):3531–3545, 2018.

[183] BORRELLI, F.; BEMPORAD, A. ; MORARI, M.. **Geometric algorithm for multiparametric linear programming**. Journal of optimization theory and applications, 118(3):515–540, 2003.

[184] MEGIDDO, N.; CHANDRASEKARAN, R.. **On the ε-perturbation method for avoiding degeneracy**. Operations Research Letters, 8(6):305–308, 1989.

[185] NOCEDAL, J.; WRIGHT, S.. **Numerical optimization**. Springer Science & Business Media, 2006.

[186] BÖHM, V.. **On the continuity of the optimal policy set for linear programs**. SIAM Journal on Applied Mathematics, 28(2):303–306, 1975.

[187] GAL, T.. **Postoptimal Analyses, Parametric Programming, and Related Topics: degeneracy, multicriteria decision making, redundancy**. Walter de Gruyter, 2010.

[188] BROCKWELL, P. J.; DAVIS, R. A.. **Time series: theory and methods**. Springer science & business media, 2009.

[189] WHITE, H.. **Asymptotic theory for econometricians**. Academic press, 2014.

[190] BILLINGSLEY, P.. **Probability and Measure**. Wiley Series in Probability and Statistics. Wiley, 1986.

[191] DOMOWITZ, I.; EL-GAMAL, M. A.. **A consistent test of stationary-ergodicity**. Econometric Theory, 9(4):589–601, 1993.

[192] CONN, A. R.; SCHEINBERG, K. ; VICENTE, L. N.. **Introduction to derivative-free optimization**. SIAM, 2009.

[193] HENNINGSEN, A.; TOOMET, O.. **maxlik: A package for maximum likelihood estimation in r**. Computational Statistics, 26(3):443–458, 2011.

[194] KIRSCHEN, D. S.; STRBAC, G.. **Fundamentals of power system economics**. John Wiley & Sons, 2018.

[195] BABAEINEJADSAROOKOLAEE, S.; BIRCHFIELD, A.; CHRISTIE, R. D.; COFFRIN, C.; DEMARCO, C.; DIAO, R.; FERRIS, M.; FLISCOUNAKIS, S.; GREENE, S.; HUANG, R. ; OTHERS. **The power grid library for benchmarking ac optimal power flow algorithms**. arXiv preprint arXiv:1908.02788, 2019.

[196] LI, H.; BOSE, A. ; VENKATASUBRAMANIAN, V. M.. **Wide-area voltage monitoring and optimization**. IEEE Transactions on Smart Grid, 7(2):785–793, 2015.

[197] MOGENSEN, P. K.; RISETH, A. N.. **Optim: A mathematical optimization package for julia**. Journal of Open Source Software, 3(24), 2018.

[198] TIBSHIRANI, R.. **Regression shrinkage and selection via the lasso: a retrospective**. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(3):273–282, 2011.

[199] COES. **Programa mediano plazo operación, 2021, mayo, informes**, 2021.

[200] CND. **Centro nacional de despacho, informe de operaciones, 2021**, 2021.

[201] JOSZ, C.; FLISCOUNAKIS, S.; MAEGHT, J. ; PANCIATICI, P.. **Ac power flow data in matpower and qcqp format: itesla, rte snapshots, and pegase**. arXiv preprint arXiv:1603.01533, 2016.

[202] FLISCOUNAKIS, S.; PANCIATICI, P.; CAPITANESCU, F. ; WEHENKEL, L.. **Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions**. IEEE Transactions on Power Systems, 28(4):4909–4917, 2013.

[203] AMAZON. **EC2 amazon web services, aws, 2022**, 2022.

[204] IEA. **Hydropower**. https://www.iea.org/reports/hydropower, June 2022.

[205] PEREIRA, M. V.; PINTO, L. M.. **Multi-stage stochastic optimization applied to energy planning**. Mathematical Programming, 52(1-3):359–375, 1991.

[206] **Conversations on designs – wholesale electricity markets**. IEEE Power and Energy Magazine, 17(1):1–108, January 2019.

[207] GJELSVIK, A.; BELSNES, M. M. ; HAUGSTAD, A.. **An algorithm for stochastic medium-term hydrothermal scheduling under spot price uncertainty**. In: PROCEEDINGS OF 13TH POWER SYSTEMS COMPUTATION CONFERENCE, 1999.

[208] LÖHNDORF, N.; SHAPIRO, A.. **Modeling time-dependent randomness in stochastic dual dynamic programming**. European Journal of Operational Research, 273(2):650–661, 2019.

[209] STEEGER, G.; BARROSO, L. A. ; REBENNACK, S.. **Optimal bidding strategies for hydro-electric producers: A literature survey**. Power Systems, IEEE Transactions on, 29(4):1758–1766, 2014.

[210] GROSS, G.; FINLAY, D.. **Generation supply bidding in perfectly competitive electricity markets**. Computational & Mathematical Organization Theory, 6(1):83–98, 2000.

[211] PEREIRA, M. V. F.; PINTO, L. M. V. G.. **Multi-stage stochastic optimization applied to energy planning**. Mathematical Programming, 52(1-3):359–375, 1991.

[212] FLACH, B.; BARROSO, L. ; PEREIRA, M.. **Long-term optimal allocation of hydro generation for a price-maker company in a competitive market: latest developments and a stochastic dual dynamic programming approach**. IET generation, transmission & distribution, 4(2):299–314, 2010.

[213] STEEGER, G.; REBENNACK, S.. **Dynamic convexification within nested Benders decomposition using Lagrangian relaxation: An application to the strategic bidding problem**. European Journal of Operational Research, 257(2):669–686, 2017.

[214] HABIBIAN, M.; DOWNWARD, A. ; ZAKERI, G.. **Multistage stochastic demand-side management for price-making major consumers of electricity in a co-optimized energy and reserve market**. European Journal of Operational Research, 280(2):671–688, 2020.

[215] PÉREZ-DÍAZ, J. I.; GUISÁNDEZ, I.; CHAZARRA, M. ; HELSETH, A.. **Medium-term scheduling of a hydropower plant participating as a price-maker in the automatic frequency restoration reserve market**. Electric Power Systems Research, 185:106399, 2020.

[216] ARTEAGA, J.; ZAREIPOUR, H.. **A Price-Maker/Price-Taker Model for the Operation of Battery Storage Systems in Electricity Markets**. IEEE Transactions on Smart Grid, 10(6):6912–6920, 2019.

[217] BARROSO, L. A.; CARNEIRO, R. D.; GRANVILLE, S.; PEREIRA, M. V. ; FAMPA, M.. **Nash equilibrium in strategic bidding: a binary expansion approach**. Power Systems, IEEE Transactions on, 21(2):629, 2006.

[218] FANZERES, B.; STREET, A. ; POZO, D.. **A column-and-constraint generation algorithm to find nash equilibrium in pool-based electricity markets**. Electric Power Systems Research, 189:106806, 2020.

[219] SCOTT, T. J.; READ, E. G.. **Modelling hydro reservoir operation in a deregulated electricity market**. International Transactions in Operational Research, 3(3-4):243–253, 1996.

[220] KELMAN, R.; BARROSO, L. A. N. ; PEREIRA, M. V. F.. **Market power assessment and mitigation in hydrothermal systems**. Power Systems, IEEE Transactions on, 16(3):354–359, 2001.

[221] VILLAR, J.; RUDNICK, H.. **Hydrothermal market simulator using game theory: assessment of market power**. Power Systems, IEEE Transactions on, 18(1):91–98, 2003.

[222] MOLINA, J. P.; ZOLEZZI, J. M.; CONTRERAS, J.; RUDNICK, H. ; REVECO, M. J.. **Nash-cournot equilibria in hydrothermal electricity markets**. IEEE Transactions on Power Systems, 26(3):1089–1101, 2010.

[223] HÖSCHLE, H.; LE CADRE, H.; SMEERS, Y.; PAPAVASILIOU, A. ; BELMANS, R.. **An admm-based method for computing risk-averse equilibrium in capacity markets**. IEEE Transactions on Power Systems, 33(5):4819–4830, 2018.

[224] STEEGER, G.; REBENNACK, S.. **Strategic bidding for multiple price-maker hydroelectric producers**. IIE Transactions, 47(9):1013–1031, 2015.

[225] LÖSCHENBRAND, M.; WEI, W. ; LIU, F.. **Hydro-thermal power market equilibrium with price-making hydropower producers**. Energy, 164:377–389, 2018.

[226] WU, X.; CHENG, C.; MIAO, S.; LI, G. ; LI, S.. **Long-Term Market Competition Analysis for Hydropower Stations using SSDP-Games**. Journal of Water Resources Planning and Management, 146(6):04020037, 2020.

[227] BUSHNELL, J.. **A Mixed Complementarity Model of Hydrothermal Electricity Competition in the Western United States**. Operations Research, 51(1):80–93, 2003.

[228] ALMEIDA, K. C.; CONEJO, A. J.. **Medium-Term Power Dispatch in Predominantly Hydro Systems: An Equilibrium Approach**. IEEE Transactions on Power Systems, 28(3):2384–2394, 2013.

[229] CRUZ, M.; FINARDI, E.; MATOS, V. D. ; LUNA, J.. **Strategic bidding for price-maker producers in predominantly hydroelectric systems**. Electric Power Systems Research, 140:435–444, 2016.

[230] LIANG, Y.; LIN, Q.; HE, S.; LIU, Q.; CHEN, Z. ; LIU, X.. **Power Market Equilibrium Analysis with Large-scale Hydropower System Under Uncertainty**. 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 00:329–333, 2020.

[231] PHILPOTT, A.; FERRIS, M. ; WETS, R.. **Equilibrium, uncertainty and risk in hydro-thermal electricity systems**. Mathematical Programming, 157(2):483–513, 2016.

[232] MOISEEVA, E.; HESAMZADEH, M. R.. **Bayesian and Robust Nash Equilibria in Hydrodominated Systems Under Uncertainty**. IEEE Transactions on Sustainable Energy, 9(2):818–830, 2018.

[233] INFANGER, G.; MORTON, D. P.. **Cut sharing for multistage stochastic linear programs with interstage dependency**. Mathematical Programming, 75(2):241–256, 1996.

[234] SHAPIRO, A.. **Analysis of stochastic dual dynamic programming method**. European Journal of Operational Research, 209(1):63–72, 2011.

[235] SHAPIRO, A.; TEKAYA, W.; DA COSTA, J. P. ; SOARES, M. P.. **Risk neutral and risk averse stochastic dual dynamic programming method**. European journal of operational research, 224(2):375–391, 2013.

[236] DOWSON, O.; MORTON, D. P. ; PAGNONCELLI, B. K.. **Incorporating convex risk measures into multistage stochastic programming algorithms**. Annals of Operations Research, p. 1–25, 2022.

[237] DEBIA, S.; PINEAU, P.-O. ; SIDDIQUI, A. S.. **Strategic storage use in a hydro-thermal power system with carbon constraints**. Energy Economics, 98:105261, 2021.

[238] FREDO, G. L. M.; FINARDI, E. C. ; DE MATOS, V. L.. **Assessing solution quality and computational performance in the long-term generation scheduling problem considering different hydro production function approaches**. Renewable energy, 131:45–54, 2019.

[239] LINO, P.; BARROSO, L. A. N.; PEREIRA, M. V.; KELMAN, R. ; FAMPA, M. H.. **Bid-based dispatch of hydrothermal systems in competitive markets**. Annals of Operations Research, 120(1-4):81–97, 2003.

[240] BARROSO, L.; GRANVILLE, S.; JACKSON, P.; PEREIRA, M. ; READ, E.. **Overview of virtual models for reservoir management in competitive markets**. In: PROCEEDINGS 4TH IEEE/CIGRÉ INTERNATIONAL WORKSHOP ON HYDRO SCHEDULING IN COMPETITIVE MARKETS, BERGEN, NORWAY, 2012.