

Copyright
by
Anderson Rodrigo de Queiroz
2011

The Dissertation Committee for Anderson Rodrigo de Queiroz certifies that this is the approved version of the following dissertation:

**A Sampling-based Decomposition Algorithm with
Application to Hydrothermal Scheduling: Cut
Formation and Solution Quality**

Committee:

David P. Morton, Supervisor

Elmira Popova

Erhan Kutanoglu

Jonathan F. Bard

Ross Baldick

**A Sampling-based Decomposition Algorithm with
Application to Hydrothermal Scheduling: Cut
Formation and Solution Quality**

by

Anderson Rodrigo de Queiroz, B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2011

UMI Number: 3530295

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3530295

Published by ProQuest LLC (2012). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Dedicated to my wife Luana,
to my parents, to my sisters and to my grandparents.

“I learned about the strength you can get from a close family life. I learned to keep going, even in bad times. I learned not to despair, even when my world was falling apart. I learned that there are no free lunches. And I learned the value of hard work.” Lee Iacocca

Acknowledgments

I think there are no words to describe how grateful I am for all the help and support that my advisor David Morton has given me over these years. As a hero, David guided me with his brilliant ideas, his unique advices and his set of one of kind lectures since my first semester at the University of Texas at Austin. He taught me the art of linear programming, stochastic optimization and advanced math programming, always concerned with the theoretical development combined with the importance of real life applications. The development of this dissertation would not be possible without him. I have to say that David inspired me and that he changed my life.

I wish to thank Professor Elmira Popova for all her help and her encouragement throughout these years and also for her contributions as a committee member of this dissertation. Elmira was always magnificent and she never stopped to believe in me. I want to thank the other committee members for all their contributions, Professor Ross Baldick from ECE department and Professors Jonathan Bard and Erhan Kutanoglu from the ORIE department.

I would like to thank my sponsors Capes and Fulbright for the financial support and for making this dream a reality for me. I thank the Institute of International Education (IIE) for assisting me with my grant in the US. I acknowledge the Texas Advanced Computing Center (TACC) at The Uni-

versity of Texas at Austin for providing HPC resources that have contributed to the research results reported within this dissertation. I want to thank my friends from Federal University of Itajuba, Welinton Dias and Tiago Vitor, for helping me to acquire knowledge in C++. I would like to thank Professor Andre Marcato from Federal University of Juiz de Fora for his help and for our discussions about the Aggregate Reservoir Representation.

I want to embrace and say thanks to my entire family that always supported me with their shoulders and hearts. First to my mother Solange and to my sisters Amanda and Aline: Thanks for all your love and for being there for me every step of the way. To my grandparents Paulo and Mercedes Balasini: It would not be possible even to dream about this without you, thank you from all my heart. To my father-in-law Jose Wanderley Marangon Lima: You are like a father to me, thank you for showing me how to be a professional and to aspire always for more wisdom. I want to thank my father Norival, my uncle Paulinho, my aunts Silvia, Rosana and Cecilia, and my cousins Ana Paula and Andre.

A special thanks goes to my beloved and beautiful wife, Luana Medeiros Marangon Lima, for all her love and care. Thank you for being the kindest and the sweetest person to me. I will be forever proud to have walked side by side with you to achieve this fantastic goal.

Finally, I would like to thank God in heaven for all the wonderful things that I experience in my life. Thanks for given me strength, faith, hope and heart to pursuit the right path in this Earth.

A Sampling-based Decomposition Algorithm with Application to Hydrothermal Scheduling: Cut Formation and Solution Quality

Publication No. _____

Anderson Rodrigo de Queiroz, Ph.D.
The University of Texas at Austin, 2011

Supervisor: David P. Morton

We consider a hydrothermal scheduling problem with a mid-term horizon (HTSPM) modeled as a large-scale multistage stochastic program with stochastic monthly inflows of water to each hydro generator. In the HTSPM we seek an operating policy to minimize the sum of present and expected future costs, which include thermal generation costs and load curtailment costs. In addition to various simple bounds, problem constraints involve water balance, demand satisfaction and power interchanges.

Sampling-based decomposition algorithms (SBDAs) have been used in the literature to solve HTSPM. SBDAs can be used to approximately solve problem instances with many time stages and with inflows that exhibit inter-stage dependence. Such dependence requires care in computing valid cuts for the decomposition algorithm.

In order to help maintain tractability, we employ an aggregate reservoir representation (ARR). In an ARR all the hydro generators inside a specific region are grouped to effectively form one hydro plant with reservoir storage and generation capacity proportional to the parameters of the hydro plants used to form that aggregate reservoir. The ARR has been used in the literature with energy balance constraints, rather than water balance constraints, coupled with time series forecasts of energy inflows. Instead, we prefer as a model primitive to have the time series model forecast water inflows. This, in turn, requires that we extend existing methods to compute valid cuts for the decomposition method under the resulting form of interstage dependence.

We form a sample average approximation of the original problem and then solve this problem by these special-purpose algorithms. And, we assess the quality of the resulting policy for operating the system. In our analysis, we compute a confidence interval on the optimality gap of a policy generated by solving an approximation on a sampled scenario tree. We present computational results on test problems with 24 monthly stages in which the inter-stage dependency of hydro inflows is modeled using a dynamic linear model. We further develop a parallel implementation of an SBDA. We apply SBDA to solve the HTSPM for the Brazilian power system that has 150 hydro generators, 151 thermal generators and 4 regions that each characterize an aggregate reservoir. We create and solve four different HTSPM instances where we change the input parameters with respect to generation capacity, transmission capacity and load in order to analyze the difference in the total expected cost.

Table of Contents

| | |
|---|-------------|
| Acknowledgments | v |
| Abstract | vii |
| List of Tables | xiii |
| List of Figures | xvi |
| Chapter 1. Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Dissertation Outline | 2 |
| Chapter 2. Hydro-Thermal Scheduling Problem | 5 |
| 2.1 Problem Overview | 5 |
| 2.1.1 Problem Characteristics | 6 |
| 2.1.2 Model with Individual Hydro Plants | 8 |
| 2.1.3 Model with Aggregate Reservoir Representation | 9 |
| 2.2 Hydro Generation Power Plants | 11 |
| 2.2.1 The Hydro Plant Schemes | 13 |
| 2.2.1.1 Hydro Plants with Reservoirs | 13 |
| 2.2.1.2 Run-of-River Hydro Plants | 14 |
| 2.2.1.3 Small Hydro Plants | 15 |
| 2.2.2 Common Hydro Plants Parameters | 16 |
| 2.2.2.1 Reservoir Head | 16 |
| 2.2.2.2 Hydro Plant Power Output | 17 |
| 2.2.2.3 Hydro Plant Productivity | 18 |
| 2.2.2.4 Reservoir Surface Area | 19 |
| 2.3 The Aggregate System Reservoir Description | 19 |
| 2.3.1 Hydro Power System Configuration | 20 |

| | | |
|--|--|-----------|
| 2.3.2 | Maximum Energy Storage Capacity | 22 |
| 2.3.3 | Controllable Energy | 23 |
| 2.3.4 | Uncontrollable Energy | 24 |
| 2.3.5 | Natural Inflow Energy | 25 |
| 2.3.6 | Minimal Outflow Energy | 26 |
| 2.3.7 | Maximum Hydro Generation | 27 |
| 2.3.8 | Evaporation Energy Losses | 28 |
| 2.3.9 | Energy Losses to Fill Minimal Reservoir Volumes | 29 |
| 2.3.10 | Water Diversion Energy Losses | 30 |
| 2.3.11 | Energy from Hydro Plants Operating with Less Machines | 32 |
| 2.4 | Thermal Generation Power Plants | 33 |
| 2.5 | Power System Demand | 35 |
| 2.6 | Energy Exchanges in the System | 36 |
| 2.7 | Mathematical Model | 37 |
| Chapter 3. A Sampling-Based Decomposition Algorithm | | 46 |
| 3.1 | Literature Survey | 46 |
| 3.2 | A Two Stage Stochastic Linear Program | 52 |
| 3.3 | A Multi-Stage Stochastic Linear Program | 57 |
| 3.4 | Stochastic Process Governing Energy Inflows | 60 |
| 3.4.1 | Interstage Independent Case | 61 |
| 3.4.2 | Interstage Dependent Case | 61 |
| 3.5 | From an Infinite to a Finite Sampled Scenario Tree | 62 |
| 3.5.1 | Sampled Scenario Tree under Interstage Independence (SSTII) | 63 |
| 3.5.2 | Sampled Scenario Tree under Interstage Dependence Model (SSTIDM) | 65 |
| 3.5.3 | Sampled Scenario Tree Observations | 68 |
| 3.6 | Sampling-based Decomposition Algorithm | 69 |
| 3.7 | Cut-Sharing under Interstage Dependency Model | 75 |
| 3.8 | A Modified SLP- T Formulation | 78 |
| 3.8.1 | Basic Cut Computation | 81 |
| 3.8.2 | Expanding the State | 82 |

| | | |
|--|--|------------|
| 3.8.3 | An Equivalence Result | 84 |
| 3.8.4 | Cut-Sharing for SLP-3 | 88 |
| 3.8.5 | Cut-Sharing for SLP- T | 91 |
| 3.9 | Parallel Implementation | 93 |
| Chapter 4. Solution Quality Evaluation in a Multistage Stochastic Program | | 97 |
| 4.1 | Policy Generation Procedure and Cost Estimator | 97 |
| 4.1.1 | Policy Generation Procedure (PGP) | 99 |
| 4.2 | Lower Bound Estimation | 101 |
| 4.3 | Confidence Interval Construction on Policy Quality | 103 |
| 4.4 | Tightening the Confidence Interval | 106 |
| 4.5 | Solution Quality Analysis | 109 |
| 4.5.1 | Upper Bound Estimator Analysis | 110 |
| 4.5.2 | Lower Bound Estimator Analysis | 114 |
| 4.5.3 | Confidence Interval Analysis | 116 |
| 4.5.4 | Computational Time Analysis | 118 |
| Chapter 5. Case Study | | 121 |
| 5.1 | The Brazilian Interconnected Power System | 121 |
| 5.1.1 | Power Generation Production and Capacity | 122 |
| 5.1.2 | Power System Demand | 124 |
| 5.1.3 | Interconnections | 124 |
| 5.1.4 | Simulation Considerations | 126 |
| 5.1.4.1 | Power System Demand | 127 |
| 5.1.5 | Hydro Generation and Energy Parameters | 129 |
| 5.1.6 | Thermal Generation and Costs | 130 |
| 5.1.7 | Water Inflows | 132 |
| 5.2 | Problem Instances | 133 |
| 5.2.1 | Instance A - Base Case | 133 |
| 5.2.2 | Instance B - Addition of New Hydro Generation | 134 |
| 5.2.3 | Instance C - Addition of New Transmission Lines | 135 |
| 5.2.4 | Instance D - Increase System Demand | 135 |

| | | |
|-------------------|---|------------|
| 5.3 | Case Study Results | 136 |
| 5.3.1 | Energy Storage | 136 |
| 5.3.2 | Hydro and Thermal Generation | 137 |
| 5.3.3 | Power Exchanges | 138 |
| 5.3.4 | Total Cost | 139 |
| Chapter 6. | Conclusions and Future Work | 145 |
| 6.1 | Research Contributions | 145 |
| 6.2 | Final Remarks and Future Research Goals | 148 |
| | Appendices | 150 |
| | Appendix A. Solution Quality Instances | 151 |
| A.1 | Instance 01 - HTSPM: 24STG-10SCN-A | 151 |
| A.2 | Instance 02 - HTSPM: 24STG-20SCN-A | 155 |
| A.3 | Instance 03 - HTSPM: 24STG-60SCN-F-*SCN-O | 158 |
| A.4 | Instance 04 - HTSPM: 24STG-100SCN-F-*SCN-O | 163 |
| A.5 | Instance 05 - HTSPM: 24STG-200SCN-F-*SCN-O | 166 |
| A.6 | Instance 06 - HTSPM: 24STG-1000SCN-F-*SCN-O | 170 |
| A.7 | Instance 07 - HTSPM: 24STG-2000SCN-F-*SCN-O | 173 |
| | Appendix B. SDDP Convergence Tables | 177 |
| | Appendix C. Study Case Parameters | 185 |
| | Bibliography | 196 |
| | Vita | 205 |

List of Tables

| | | |
|------|--|-----|
| 4.1 | Upper Bound Estimator for PGPs From Scenario Trees of Growing Size | 111 |
| 4.2 | Paired Student- t Test for PGP with Different Scenario-Tree Sizes | 113 |
| 4.3 | Lower Bound Estimator LBEs From Scenario Trees of Growing Size | 114 |
| 4.4 | Gap Estimator and CI Width From Scenario Trees of Growing Size | 117 |
| | | |
| 5.1 | Net Demand for Each Region [MW-month] - Optimization Input | 128 |
| 5.2 | Initial Energy Storage for Each Aggregate Reservoir [MW-month] | 129 |
| 5.3 | Max Thermal Generation Capacity for Each Region [MW-month] | 131 |
| 5.4 | Load Deficit Levels and Costs | 131 |
| 5.5 | Hydro Energy Storage Decisions [MW-month] | 137 |
| 5.6 | Controllable Hydro Generation Decisions [MW-month] | 137 |
| 5.7 | Total Thermal Generation Decision for the System [MW-month] | 138 |
| 5.8 | Power Exchanges Between Regions for Instances A and C [MW-month] | 139 |
| 5.9 | Paired Student- t Test for Expected Cost of Different Instances | 141 |
| | | |
| A.1 | Output of PGP for Instance 01 | 154 |
| A.2 | Output of LBE for Instance 01 | 155 |
| A.3 | Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 01 | 155 |
| A.4 | CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 01 | 156 |
| A.5 | Output of PGP for Instance 02 | 156 |
| A.6 | Output of LBE for Instance 02 | 158 |
| A.7 | Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 02 | 159 |
| A.8 | CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 02 | 159 |
| A.9 | Output of PGP for Instance 03 | 159 |
| A.10 | Output of LBE for Instance 03 | 161 |

| | | |
|------|--|-----|
| A.11 | Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 03 | 161 |
| A.12 | CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 03 | 162 |
| A.13 | Output of PGP for Instance 04 | 163 |
| A.14 | Output of LBE for Instance 04 | 165 |
| A.15 | Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 04 | 165 |
| A.16 | CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 04 | 166 |
| A.17 | Output of PGP for Instance 05 | 168 |
| A.18 | Output of LBE for Instance 05 | 169 |
| A.19 | Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 05 | 169 |
| A.20 | CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 05 | 169 |
| A.21 | Output of PGP for Instance 06 | 170 |
| A.22 | Output of LBE for Instance 06 | 172 |
| A.23 | Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 06 | 172 |
| A.24 | CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 06 | 173 |
| A.25 | Output of PGP for Instance 07 | 175 |
| A.26 | Output of LBE for Instance 07 | 176 |
| A.27 | Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 07 | 176 |
| A.28 | CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 07 | 176 |
| | | |
| B.1 | SDDP Convergence Instance 01. Zlb is the lower bound, Zub is the sample mean upper bound, and Δ is a 95% confidence interval half-width associated with upper bound estimator. The column labeled time is the time in seconds, required for that iteration. | 178 |
| B.2 | SDDP Convergence Instance 02. The columns in this table are analogous to those in Table B.1. | 179 |
| B.3 | SDDP Convergence Instance 03. The columns in this table are analogous to those in Table B.1. | 180 |
| B.4 | SDDP Convergence Instance 04. The columns in this table are analogous to those in Table B.1. | 181 |
| B.5 | SDDP Convergence Instance 05. The columns in this table are analogous to those in Table B.1. | 182 |
| B.6 | SDDP Convergence Instance 06. The columns in this table are analogous to those in Table B.1. | 183 |
| B.7 | SDDP Convergence Instance 07. The columns in this table are analogous to those in Table B.1. | 184 |

| | | |
|-----|--|-----|
| C.1 | Electricity Demand in MW-month for Each Region | 185 |
| C.2 | Generation From Small Hydro Plants in MW-month for Regions | 186 |
| C.3 | Hydro Generation From Plants Operating in State I [MW-month] | 187 |
| C.4 | Minimal Thermal Generation for Each Region [MW-month] . | 188 |
| C.5 | Load Levels Duration | 189 |
| C.6 | Load Levels Base Multipliers | 190 |

List of Figures

| | | |
|-----|---|-----|
| 2.1 | Decision process for the hydro thermal scheduling problem . . . | 6 |
| 2.2 | Hydro Plants Cascade Representation | 9 |
| 2.3 | Power System Representation with Individual Hydro Plants . . | 10 |
| 2.4 | Aggregate Reservoir Representation | 11 |
| 2.5 | Power System Representation with Aggregate Hydro Plants . . | 12 |
| 2.6 | Representation of a Hydro Generator with a Reservoir | 14 |
| 2.7 | Representation of a Run-of-River Plant, source: Douglas (2007) | 15 |
| 2.8 | Demand at each load level | 35 |
| 2.9 | Power system with 4 regions and 1 virtual region | 38 |
| | | |
| 3.1 | Depiction of a three-stage scenario tree | 59 |
| 3.2 | Three-stage scenario tree under interstage independent model | 64 |
| 3.3 | Three-stage scenario tree under interstage dependent model . . | 67 |
| 3.4 | The figure depicts forward and backward passes of the SDDP algorithm | 71 |
| 3.5 | General SDDP Scheme | 94 |
| 3.6 | General SDDP Parallelization Scheme | 96 |
| | | |
| 4.1 | Upper Bound Point Estimate and CI Width with Scenarios Trees of Various Size | 112 |
| 4.2 | L_{n_e} in Ascending Order for Each Scenario Tree Size | 115 |
| 4.3 | LBE Point Estimate and Halfwidth for Each Instance | 116 |
| 4.4 | Confidence Interval Errors for Each Instance | 118 |
| 4.5 | Iteration Time in Minutes for Each Instance | 119 |
| 4.6 | Total Time in Minutes for Each Instance | 120 |
| | | |
| 5.1 | Brazilian Power System Regions | 122 |
| 5.2 | Brazil's Total Electricity Production in 2010 | 123 |
| 5.3 | Brazil's Total Generation Capacity in 2010 | 124 |

| | | |
|------|--|-----|
| 5.4 | Percentage of Total Electricity Demand in 2010 for Each Region | 125 |
| 5.5 | Transmission Capacities Between Regions for 2010 | 126 |
| 5.6 | Max Hydro Generation Functions for Each Region - 1 st Month | 130 |
| 5.7 | Thermal Generation Capacity x Cost - Brazilian System 1 st Month | 132 |
| 5.8 | Upper Bound Point Estimate and CI Width for Instances . . . | 140 |
| | | |
| A.1 | SDDP Convergence for Instance 01 | 153 |
| A.2 | Lower Bound for Each Tree on Instance 01 | 154 |
| A.3 | SDDP Convergence for Instance 02 | 157 |
| A.4 | Lower Bound for Each Tree on Instance 02 | 158 |
| A.5 | SDDP Convergence for Instance 03 | 160 |
| A.6 | Lower Bound for Each Tree on Instance 03 | 161 |
| A.7 | SDDP Convergence for Instance 04 | 164 |
| A.8 | Lower Bound for Each Tree on Instance 04 | 165 |
| A.9 | SDDP Convergence for Instance 05 | 167 |
| A.10 | Lower Bound for Each Tree on Instance 05 | 168 |
| A.11 | SDDP Convergence for Instance 06 | 171 |
| A.12 | Lower Bound for Each Tree on Instance 06 | 172 |
| A.13 | SDDP Convergence for Instance 07 | 174 |
| A.14 | Lower Bound for Each Tree on Instance 07 | 175 |
| | | |
| C.1 | Power Plants Cascade Scheme by Basin, source: ONS (2009) . | 192 |
| C.2 | Minimal Outflow Energy Functions of Storage for Each Region - 1 st Month | 194 |
| C.3 | Evaporation Losses Functions of Storage for Each Region - 1 st Month | 195 |
| C.4 | Water Diversion Functions of Storage for Each Region - 1 st Month | 195 |

Chapter 1

Introduction

1.1 Motivation

Renewable energy is a key piece of the supply side of power systems around the world, as sustainable development becomes a goal of our societies. This energy is becoming important to mitigate changes in climate due to global warming, while not depriving humans of access to goods and services to which they have become accustomed. The fossil age of the last century contributes to our current way of living but we are facing the cost of that kind of energy. Therefore, the access to clean energy is part of many energy programs of European countries and others which are engaged in making a better world to live.

Renewable resources should be part of the solution to fulfill the energy demands of seven billion people. There are many types of green energy like wind, solar, geothermal, biomass and tidal generation which can be combined to compose a sustainable portfolio of electricity providers. The main problem with that kind of energy is that we cannot store it and it depends on the natural resources which usually do not match the demand as the times and locations that is necessary.

There are some exceptions to this limitation of renewable energy including hydro-energy resources in which storage reservoirs can be used to control the timing of energy generation. In many countries like Canada and Norway, most of the hydro-electric generation comes from run-of-river power plants which depend basically on the ice cycles. However, in other countries like Brazil and Colombia, the hydro plants have large reservoirs which can be used for optimizing the use of water. In Brazil around 85% of the electricity production comes from hydro plants, and the reservoirs have the capacity of providing energy for several years to satisfy current load levels even if the worst of historical droughts occurs. Given this characteristic, the hydro-thermal scheduling problem becomes very important for minimizing the total operational costs derived from the cost of fuels that feed the thermal plants and the costs of possible demand curtailments. This problem is complicated by the fact that we do not have perfect forecasts for future inflows of water into our reservoirs. This dissertation concerns minimizing the expected value of these operational costs, given a time series forecast for future inflows into the hydrological reservoirs.

1.2 Dissertation Outline

In Chapter 2 we describe a hydro-thermal scheduling problem. In order to help maintain tractability we use an aggregate reservoir representation that is also described in this chapter. A detailed mathematical model known as a multi-stage stochastic program is used to represent this problem. The model

is presented in detail, as we describe the model’s objective function and each of its constraints.

In Chapter 3 we present a sampling-based decomposition algorithm (SBDA) that we use to solve our hydro-thermal scheduling model. A literature survey is presented, and we describe the main concepts related to SBDAs. The first SBDA, known as stochastic dual dynamic programming (SDDP), is described as well as the cut-sharing methodology needed for practical implementations of this algorithm when hydrological inflows exhibit inter-stage dependence. In this chapter we extended previous methodologies related to cut-sharing for SBDAs, establishing new formulas for sharing cuts under a novel interaction between a class of inter-stage dependency models for the hydro inflows, and how they appear in the stochastic program. In Chapter 3 we also describe a parallel implementation of our SBDA.

In Chapter 4 we present a methodology to assess the solution quality of SBDAs. A policy generation procedure, and an associated sampling-based scheme, is described, that we can use to obtain a point estimate for an upper bound on the optimal value of the true problem, along with an estimate of the sampling error. We further describe a procedure for estimating a lower bound on the true problem’s optimal value, along with an estimate of its sampling error. With the lower bound, upper bound, and associated sampling errors, we then construct a confidence interval on the optimality gap of the policy generated by our SBDA. We investigate the results we obtain using this procedure for assessing solution quality, using a problem with 24 monthly

stages. Our investigation studies the nature of the results obtained, using different sample sizes, and provides guidance on the sample sizes necessary to both obtain a high-quality policy, and to prove that it is indeed near-optimal.

In Chapter 5 we present a case study based on the Brazilian interconnected power system. We present optimization results for a family of 24-stage problems in which we consider model variants in which additional hydro capacity becomes available, additional transmission capacity becomes available, and additional load is introduced to the system.

In Chapter 6 we conclude this dissertation with a summary of the main contributions of this work. We finalize our discussion pointing out some possibilities for future work.

Chapter 2

Hydro-Thermal Scheduling Problem

2.1 Problem Overview

In a hydro-thermal scheduling problem we are interested in minimizing the production costs of electricity to supply the system demand considering the operation of both hydro and thermal power generators. In the power sector, the independent system operator (ISO) may decide to use the water available at the hydro plant reservoirs to produce electricity. Doing so avoids the economic expense required to dispatch thermal power plants, but can risk hydro availability in future time periods.

The water available to produce electricity is bounded by the reservoir storage capacities and future water inflows at the river basins of these reservoirs. Most of the time thermal generation must be used to complement the necessary amount of electricity to meet system demand. However, wise use of the hydro and thermal system resources can reduce costs. Figure 2.1 presents the decision process that the ISO faces in operating a hydro-thermal system.

Hydroelectricity is inexpensive to produce, with virtually no associated costs for water usage once hydro turbines are installed. One possibility is to measure indirectly the value of the electricity produced by hydro genera-

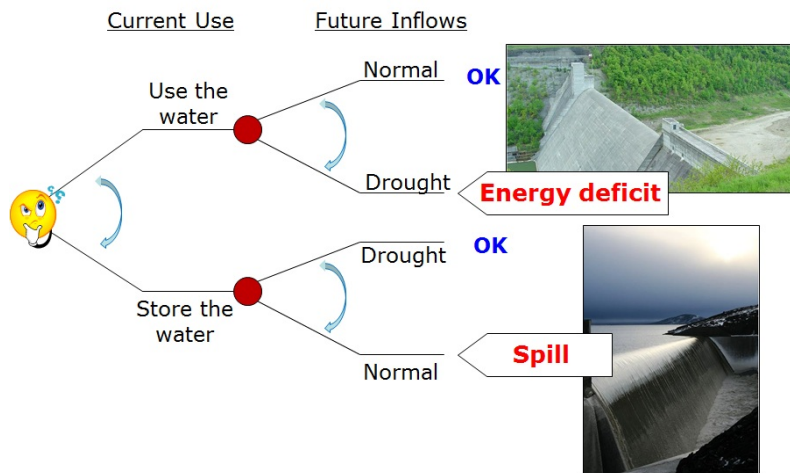


Figure 2.1: Decision process for the hydro thermal scheduling problem

tors by computing the difference between the operational costs in a system containing only thermal generation and the operational costs in that same system containing both hydro and thermal generation. In the former system the thermal generators are usually dispatched in a least cost fashion to meet system demand. In the latter system the thermal generators are dispatched to complement the electricity produced by the hydro generators. By comparing both operational costs we can indirectly measure the value of hydroelectric generation to the system.

2.1.1 Problem Characteristics

The available hydro generation capacity at a particular time period depends on the amount of water stored in the hydro generator's reservoir. If this hydro generator is part of a cascade system (there are generators upstream and/or

downstream in the same river) the amount of stored water is influenced by the operational decisions applied to the generators upstream. This couples the problem in space.

Natural water inflows are responsible for a large part of the future water supply that will be available to generate electricity. These future water inflows and their stochastic nature complicates the resulting hydro thermal scheduling model. Our problem is dynamic because present decisions affect the future. Figure 2.1 presents the intuition behind this idea. On the one hand, if the ISO uses a large quantity of water to produce electricity today and in the future a drought occurs it may be necessary to dispatch more expensive thermal generation (e.g., diesel generators) in order to supply demand or even to curtail some load. This procedure would generate unnecessary expenses to the system. On the other hand, if the ISO stores water to use in the future and large water inflows occur it is possible that the operator must spill some water from the reservoir. This implies a waste of potential energy and hence money. These characteristics couple the problem in time.

In this problem, there are multiple interconnected hydro reservoirs in the system that need to be scheduled over many time periods. This combined with stochastic inflows means that the problem can be defined as a multi-stage stochastic program. The objective is to determine the optimal amount of hydro and thermal electricity to be produced at each time period satisfying the problem constraints such that the expected operating costs of the system are minimized.

2.1.2 Model with Individual Hydro Plants

In formulating a model of the hydro-thermal scheduling problem with individual hydro generators one is interested in determining generation targets for each hydro and thermal plant over multiple time periods with the objective at minimizing the total operational costs. In this setting, the parameters related to water inflows, turbine water outflows, water spillage and water storage are represented by water volumes. In general, such a model captures an interconnected cascade system so that decisions at each generator may affect the whole cascade. Depending on the model's horizon and time discretization, water volumes that are used to produce electricity and water volumes that are spilled from upstream reservoirs are available at the same time period at the next downstream reservoir and can be used to produce electricity. So, besides the water inflows, the amount of water available at each reservoir depends on operational decisions upstream.

Figure 2.2 depicts a hydro plant cascade system. The triangles represent hydro plants with reservoirs and the circles represent run-of-river hydro plants. In this setting, operational decisions of generators 1 through 5 have influence on the water available for generators 6 and 7, and that of generator 7 is also influenced by decisions of generator 6, and so on.

A power system representation with 4 different regions is shown in Figure 2.3. Each region has its individual demand and its individual power generators (hydro and thermal). The hydro generators within a region are coupled by a cascade scheme, and the thermal generators are independent of

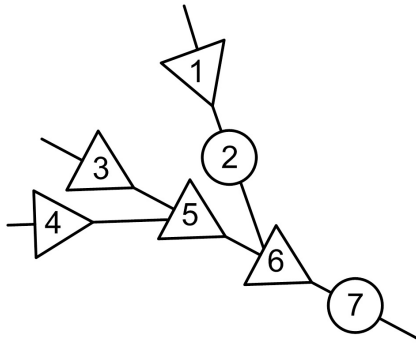


Figure 2.2: Hydro Plants Cascade Representation

each other. Figure 2.3 also shows transmission lines that interconnect the power system, transferring power between regions. With transmission lines in a hydro-thermal power system, the ISO can take advantage of the hydrological diversity between regions in order to operate the system in the best possible way.

The thermal units play an important role in the system reliability. During periods with unfavorable hydrological conditions, the thermal plants can be dispatched to help to satisfy demand. This allows the hydro plants to store water in the reservoirs and produce energy in future time periods as necessary. Thus, one of the main purposes of the thermal units and transmission lines is to optimize the utilization of system resources (water and fuel).

2.1.3 Model with Aggregate Reservoir Representation

The main goal in formulating a model of the hydro-thermal scheduling problem with an aggregate reservoir representation (ARR) is the same as that with

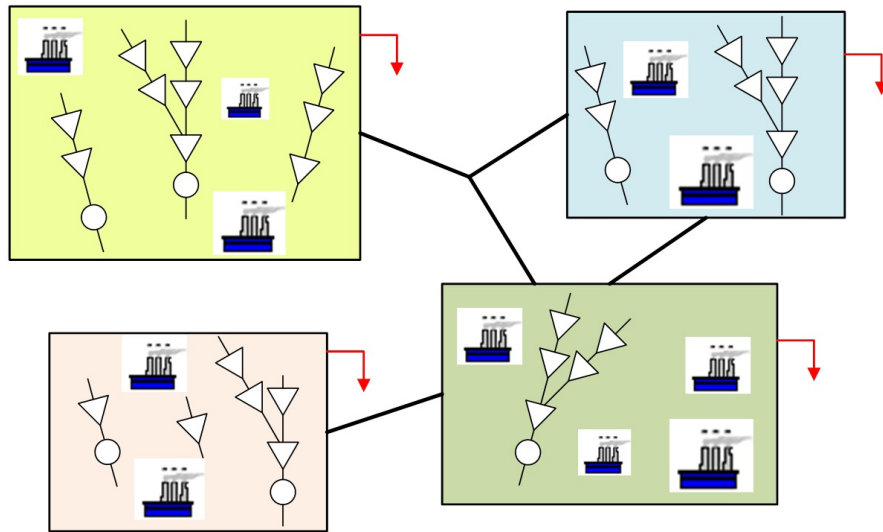


Figure 2.3: Power System Representation with Individual Hydro Plants

individual hydro plants, minimize present and future operational costs subject to a set of constraints. The main difference is that in a model with the ARR, we deal with energy instead of water. Random water inflows and water reservoir volumes are transformed into energy for an aggregate reservoir using the hydro generator productivities along the cascade. Now instead of a solution yielding individual targets for the hydro generators, a solution yields generation targets for the each aggregate reservoir during the planning horizon.

The hydro plants inside a region are aggregated into a single reservoir that has both controllable and uncontrollable energy that can be used to produce electricity. Figure 2.4 depicts some of the parameters of an aggregate reservoir. The energy inflows are divided into controllable and uncontrollable inflows. Both the controllable and the uncontrollable inflows may be used

to generate electricity immediately but only the controllable inflows can be stored for future use. We have energy losses at the aggregate reservoir due to evaporation, diversion of water (e.g., for agricultural use) and water spillage.

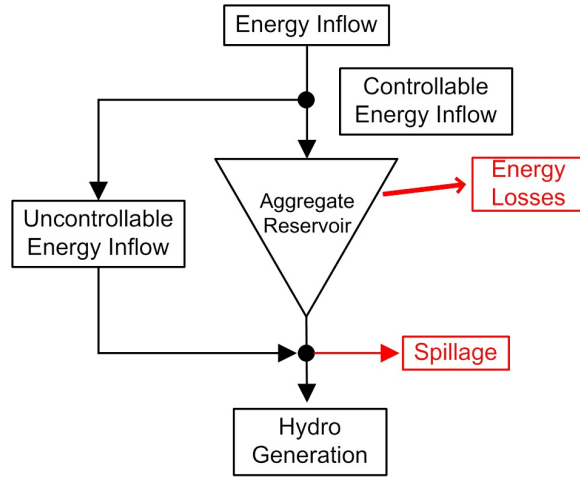


Figure 2.4: Aggregate Reservoir Representation

Figure 2.5 shows the same power system of Figure 2.3 but now under the ARR. Note that all the hydro plants inside each region are replaced by a single aggregate reservoir.

There are a number of parameters required to create the ARR for a system. We detail these parameters later in this chapter.

2.2 Hydro Generation Power Plants

Hydro generation power plants play the main role in the hydro-thermal scheduling problem. In the simplest model of a generation-demand system that contains only thermal generators, the demand is met by the least cost approach,

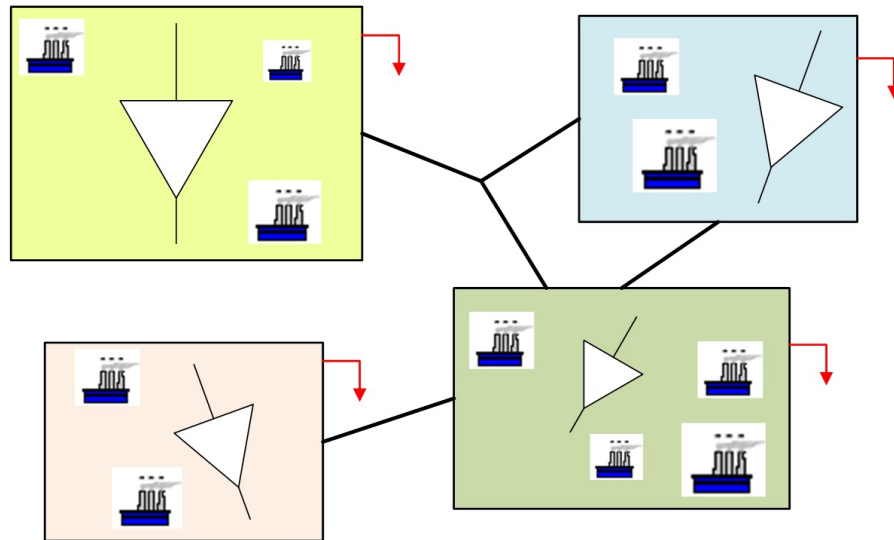


Figure 2.5: Power System Representation with Aggregate Hydro Plants

i.e., thermal generators are dispatched in ascending order of cost until demand is satisfied. In a system with hydro plants, one can produce electricity with water, at no cost, and reduce the operational expenses. But the hydro plants are influenced by the seasons, the amount of hydro generation in a system is directly proportional to the water inflows. Also, the amount of water stored at a reservoir couples the problem over time.

There are a number of important parameters that we have to consider in modeling hydro plants, but first we describe the different types of hydro plants.

2.2.1 The Hydro Plant Schemes

There are three main types of hydro generators. Hydro generators with large reservoirs, run-of-river hydro generators, and small hydro generators.

2.2.1.1 Hydro Plants with Reservoirs

Hydro plants with reservoirs can store water and use this water to produce electricity whenever it is required. Most hydro power comes from hydro plants with reservoirs, where the potential energy of dammed water driving a water turbine and a generator is transformed into electricity. The amount of energy generated is proportional to the difference in height between the top of the reservoir and the water discharge level, which is called the reservoir's head.

Figure 2.6 represents of a hydro generator, a reservoir, and further details including the dam, the penstock, the generator and the turbine. Figure 2.6 also depicts the maximum (\bar{V}) and minimum (\underline{V}) operational volumes of the reservoir and the corresponding maximum (H_{max}) and minimum (H_{min}) heads.

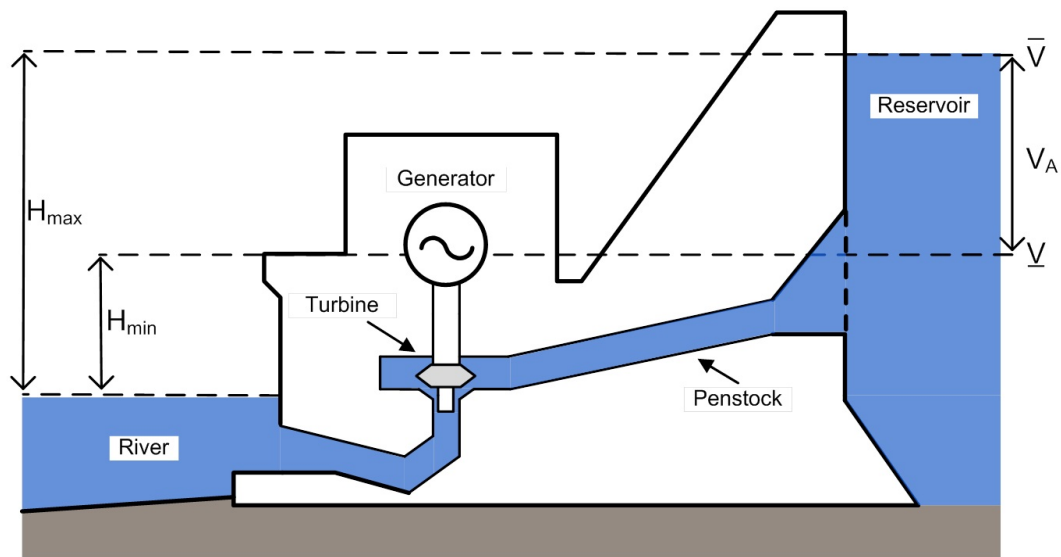


Figure 2.6: Representation of a Hydro Generator with a Reservoir

2.2.1.2 Run-of-River Hydro Plants

Run-of-river hydro generators also have reservoirs, but their capacities are small compared to other hydro plants. In a run-of-river hydro plant, it is essentially impossible to store water, and hence the flow of water either generates electricity or is spilled depending on the hydro plant's generation capacity.

Figure 2.7 depicts a typical run-of-river hydro plant. The water is captured at the intake structure and goes through a buried penstock to the powerhouse.

In this scheme if the water is not captured at the intake it will continue its normal flow in the river. The dam is required to ensure enough water goes to the penstock. The penstock carries the water from an upper elevation

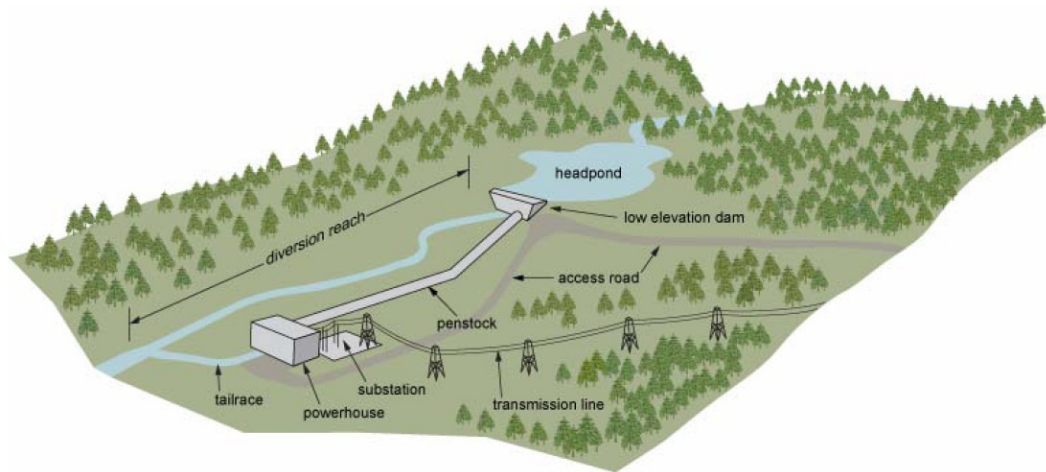


Figure 2.7: Representation of a Run-of-River Plant, source: Douglas (2007)

and delivers it to a lower elevation where the water turbine is located. The difference in height characterizes the head of a run-of-river hydro plant and is responsible for the potential energy that is transformed into electricity by the hydro plant.

2.2.1.3 Small Hydro Plants

Small hydro plants vary in power output. Most commonly installed capacities range from 1 to 30 MW, but there are also plants that have power output less than 1 MW. These hydro power plants are similar to run-of-river plants, with little or no reservoir capacity. That said, we distinguish small and run-of-river plants because we need specific equipment to simultaneously meet the requirements of sufficiently high power output, environmental restrictions and reliability.

Small hydro plants do not have serious environmental impacts. This scheme of power plant does not require large flooding areas and can be installed to produce electricity in remote regions. To a greater degree than run-of-river plants, a drawback of this scheme is that the power output of these plants is highly dependent on the natural flow of the river, making them susceptible to seasonal variations.

Usually, because of small power output and seasonal variation, these power plants are not considered as regular hydro power plants in the hydro scheduling problem. Instead the power produced by small hydro plants at a particular time period is simply subtracted from the system electricity demand.

2.2.2 Common Hydro Plants Parameters

Now we turn to the most common important parameters associated with hydro power plants.

2.2.2.1 Reservoir Head

One of the most important parameters for a reservoir is its head, the difference in height between the reservoir level and the water discharge level. Given a reservoir physical shape, a specific volume corresponds to a specific reservoir height. Each reservoir has its own polynomial that describes the relationship between volume and height.

We have different reservoir's head corresponding to the different volumes. There are four different head's values of interest: minimum, medium,

maximum and equivalent head. The equivalent head ($H_{eq,r}$) of reservoir r is calculated numerically integrating from \underline{V}_r to \overline{V}_r the height with respect to the volume. We are not interested to compute the different heads of the reservoir. We assume that the different head values are given for each reservoir in m .

The minimum ($H_{min,r}$) and maximum ($H_{max,r}$) head corresponds to the minimum and maximum reservoir's volumes respectively. The medium head ($H_{med,r}$) corresponds to 65% of the difference between the minimum and maximum volumes. For run-of-river plants there is just the available head h in m .

2.2.2.2 Hydro Plant Power Output

A hydro plant typically consists of multiple groups of power generators, and the power output represents sum of power output over each group of machines. Equation (2.1) presents the computation of this parameter. If the hydro plant is run-of-river we have to substitute $H_{max,i}$, $H_{med,i}$ and $H_{min,i}$ by h_i . This parameter is used as data for the computation of the maximum hydro generation that is presented further in this chapter.

$$P_{max,r} = \sum_{j \in C_r} n_{r,j} P_{r,j}^N \min \left(1, \left(\frac{H_{max,r}}{Q_{r,j}^N} \right)^{\varphi_r} \right) \quad (2.1a)$$

$$P_{med,r} = \sum_{j \in C_r} n_{r,j} P_{r,j}^N \min \left(1, \left(\frac{H_{med,r}}{Q_{r,j}^N} \right)^{\varphi_r} \right) \quad (2.1b)$$

$$P_{min,r} = \sum_{j \in C_r} n_{r,j} P_{r,j}^N \min \left(1, \left(\frac{H_{min,r}}{Q_{r,j}^N} \right)^{\varphi_r} \right) \quad (2.1c)$$

where,

- $P_{max,r}$ power output of hydro plant r under maximum storage [MW];
- $P_{med,r}$ power output of hydro plant r under medium storage [MW];
- $P_{min,r}$ power output of hydro plant r under minimum storage [MW];
- $P_{r,j}^N$ machine group j 's nominal power output [MW];
- $Q_{r,j}^N$ machine group j 's nominal head [m];
- C_r set of machine groups for hydro plant r ;
- $n_{r,j}$ number of power generators in machine group j ;
- φ_r constant associated with hydro plant r 's water turbine.

If the reservoir head is equal to the nominal head of machine group j then the amount of power that that machine group can produce is equal to the nominal power of that machine group. In this case, the term $\min(1, (\frac{H_r}{Q_{r,j}^N})^{\varphi_r})$ will be 1, otherwise, depending on the reservoir's head the value may be less than 1 and the power output that we can produce with that machine group will be smaller than its nominal power.

2.2.2.3 Hydro Plant Productivity

Other important parameter for a hydro generator is its productivity (ρ_r). The productivity of a hydro plant represents the plant efficiency to transform water into energy. This parameter is given in $MW/m^3/s/m$. We use this parameter to compute the energy parameters associated to each aggregate reservoir further in this chapter.

2.2.2.4 Reservoir Surface Area

A reservoir's surface area is used to compute the amount of water that evaporates during each time period. Similar to the height-volume polynomial each reservoir also has polynomials that characterizes its surface area given its height. There are different surface areas values of interest: minimum, medium and maximum surface areas. We are not interested to compute the different surface areas. We assume that the different surface area values are given for each reservoir in km^2 .

2.3 The Aggregate System Reservoir Description

The ARR, also known as the equivalent reservoir representation, was first mentioned by Pierre Mass in the mid 1940s (Klingerman, 1992). Arvanitidis and Rosing (1970a,b) present the first ARR model with application to the multireservoir hydroelectric power system in the Pacific Northwest. The ARR is an aggregation technique used to reduce the size of the model by aggregating multiple reservoirs inside a specific region to a single aggregate reservoir. The use of ARR consequently reduces the computational effort required to solve a hydro-thermal scheduling model. This type of representation models total hydro generation of a power system or even a specific region inside the system. The main idea of this approach is to deal with everything in terms of energy instead of water.

The ARR has been used since the 1970s in Brazil to model the hydroelectric power system. First the ARR was coupled with a stochastic dynamic

programming approach to solve hydro-thermal scheduling problems (Terry, 1980). Since the mid 1990s the same ARR model started to be used with stochastic dual dynamic programming to solve hydro-thermal scheduling problems for the Brazilian interconnected system (Maceira et al., 1998, 2002, 2008). Marcato (2001) presents a hybrid application of ARR together with individual hydro plants in which the objective is to model more precisely certain generators in the system. A description of the long-term hydrothermal planning problem for the Brazilian system, along with a discussion of the required energy inflow forecasting model and a comparison of the relative merits of aggregating hydro reservoirs via electrical subsystem versus aggregating them via hydrological cascade can be found in de Matos et al. (2008) and de Matos (2008).

We can construct an equivalent reservoir to represent as many or as few hydro generators as we want. Generally the equivalent reservoir, created for each region of the power system, contains a set of hydro generators for a specific river basin, where the characteristics of the random inflows tend to be similar. We follow the work presented in Cepel (2001) to model and describe the parameters of the ARR.

2.3.1 Hydro Power System Configuration

Generally, during a long planning horizon, the configuration of the hydro plants changes. New generators may become available to produce electricity in the future, the number of generators/turbines of a specific hydro plant may in-

crease or even a new reservoir may be formed and filled at a particular time. Each such change creates a different hydro power system configuration.

Three states of a hydro plant allow us to characterize the different hydro power system configurations:

- I. Filling reservoir's minimum volume: This is the state right before the hydro plant is constructed. In this state, the power plant cannot generate electricity, and so the the hydro plant productivity is set to 0. Also, the reservoir cannot be used by the system.
- II. Hydro plant operating with less machines than specified: The reservoir is available to be operated in this state but only a limited subset of the generators/turbines can be used to produce electricity. The hydro plant productivity is set to 0.
- III. Normal reservoir operation: The hydro plant's reservoir is available to be used and also the number of machines operating is equal to the specified number. The hydro plant productivity is set to its normal value and the hydro plant can produce electricity.

If a hydro plant is in state I during time period t and goes to state II or III at time period $t + 1$ (or from state II to III) we have two different hydro power system configurations. Also if there are changes in the hydro generation capacity, from one time period to the next, due to changes in the water discharge levels, we will have different hydro power system configurations.

We can define a hydro power system configuration as a set of hydro generators, each operating according to a certain state, at a particular time period t . The different configurations are important to capture because they have a strong influence in the computation of most parameters of the equivalent reservoir. For each different configuration it is necessary to recalculate the parameters that we explain next.

2.3.2 Maximum Energy Storage Capacity

The maximum energy storage capacity or potential energy of an equivalent reservoir is calculated as the energy that would be obtained from completely depleting the reservoirs of that regions. In order to establish the potential energy of a region it is necessary to adopt an operational rule of reservoir depletion. da Cruz Jr. and Soares (1999) present a nonlinear operational rule that can prioritize storage upstream or downstream in the cascade. In our work we follow the linear operational rule presented in Cepel (2001) that requires that all the reservoirs inside an ARR have to operate at the same percentage of storage capacity.

We can then obtain the maximum energy storage capacity by the sum of the available volume of each reservoir multiplied by its equivalent productivity plus the productivities of all downstream hydro plants of that reservoir. The intuition behind this calculation is that the water used to produce electricity in one hydro generator will be available downstream in the cascade for the other generators of that same cascade to produce electricity as well. We can

obtain the maximum energy storage capacity of an equivalent reservoir by:

$$EA_{max} = \frac{1}{\nu} \sum_{r \in R} \left(\bar{V}_r - \underline{V}_r \right) \sum_{j \in D_r} \rho_j H_j \quad (2.2)$$

where,

- EA_{max} maximum energy storage capacity of the equivalent reservoir
[MW-month];
- ν constant to transform $[m^3/s]$ in $[hm^3/month]$ for period t . It is
function of the number of days in a particular month;
- ρ_j productivity of the group turbine/generator of the hydro
plant j $[MW/m^3/s/m]$;
- H_j equivalent head $H_{eq,j}$ if hydro plant j has a reservoir or the
available head h_j if j is run-of-river plant $[m]$;
- R set of reservoirs in the equivalent reservoir. We could write R_i to
indicate precisely but we will suppress the subscript i for simplicity;
- D_r set of downstream hydro generators of reservoir r (including r).

The maximum energy storage capacity depends on the hydro system configuration. If a new hydro plant starts to operate in the future, the maximum energy storage capacity has to be recalculated at that time period.

2.3.3 Controllable Energy

Controllable energy represents the amount of energy that we can store or use to meet demand, and can be calculated for each time period of the planning

horizon by:

$$EC^t = \sum_{r \in R} Q_r^t \left(\rho_r H_{eq,r} + \sum_{j \in DR_r} \rho_j h_j \right) \quad (2.3)$$

where,

EC^t controllable energy available at equivalent reservoir i in time period t [MW-month];

Q_r^t natural water inflow to reservoir r in time period t [(m³/s)-month];

h_j head of the run-of-river plant j [m];

DR_r set of downstream run-of-river plants from hydro plant r until the next reservoir.

2.3.4 Uncontrollable Energy

Uncontrollable energy is the amount of energy produced by the run-of-river hydro plants in an equivalent reservoir. A share of this amount is accounted for at the upstream reservoirs of these plants, and because of that, for the calculation of uncontrollable energy we use the incremental inflow at these plants. The incremental inflow is obtained from the natural inflow discounted by the natural inflow of the reservoirs immediately upstream of that hydro plant. The uncontrollable energy can be obtained by:

$$EU^t = \sum_{j \in RR} \left(Q_j^t - \sum_{u \in U_j} Q_u^t \right) \rho_j h_j \quad (2.4)$$

where,

EU^t uncontrollable energy available at the equivalent reservoir

in time period t [MW -month];

RR set of run-of-river plants in the equivalent reservoir;

U_j set of immediately upstream reservoirs of the run-of-river plant j .

2.3.5 Natural Inflow Energy

The natural inflow energy is the sum of the controllable and the uncontrollable energy of an equivalent reservoir. We can compute the natural inflow energy by:

$$EN^t = EC^t + EU^t \quad (2.5)$$

where,

EN^t natural inflow energy in the equivalent reservoir in time period t [MW -month].

In Cepel (2001) the natural energy inflow is modeled as the random parameter for the optimization problem. They create a model to forecast energy inflows instead of water. Using the historical series of natural water inflows the historical series of energy inflows is computed using equation (2.5) and then used to calculate parameters of stochastic time series models. Because of that they do not have explicitly the share of controllable and uncontrollable energy that composes the natural inflow energy at each stage. In order to overcome this problem it is created a linear function that maps the percentage of controllable energy in natural energy inflow based on the historical data available.

We prefer to use natural water inflows forecasts at each hydro plant. We assume that the water inflows at each hydro plant given by b_r^t , $t = 2, \dots, T$ for all $r \in R \cup RR$ are random. Then we use equations (2.3) and (2.4), in each time period, to compute exactly the controllable and the uncontrollable energy parameters in each time period.

2.3.6 Minimal Outflow Energy

The minimal outflow energy represents the amount of energy produced by the minimal water outflow requirements from the set of reservoirs that are members of the ARR. The minimal outflow energy does not depend on the natural inflow of water; rather, it depends exclusively on the hydro system configuration at a particular time.

We calculate three different values of minimal outflow energy corresponding to the three different head values of the reservoirs. We can calculate the minimal outflow energy of an equivalent reservoir by:

$$EM_{max}^t = \sum_{r \in R} Q_r^t \left(\rho_r H_{max,r} + \sum_{j \in DR_r} \rho_j h_j \right) \quad (2.6a)$$

$$EM_{med}^t = \sum_{r \in R} Q_r^t \left(\rho_r H_{med,r} + \sum_{j \in DR_r} \rho_j h_j \right) \quad (2.6b)$$

$$EM_{min}^t = \sum_{r \in R} Q_r^t \left(\rho_r H_{min,r} + \sum_{j \in DR_r} \rho_j h_j \right) \quad (2.6c)$$

where, EM_{max}^t , EM_{med}^t and EM_{min}^t represent the minimal outflow energy considering the maximum, medium and minimum reservoir head respectively in time period t in MW -month.

We construct a linear function to represent the minimal outflow energy as a function of the energy stored, $EM^t(EA)$. This type of energy is uncontrollable, which means if we do not use it at a particular time period t we cannot store it for a future time period.

2.3.7 Maximum Hydro Generation

The maximum hydro generation available to be used by an equivalent reservoir has to take in account the energy storage changes during the operational process. The smaller the storage the smaller the maximum hydro generation available. The maximum hydro generation at a particular time period is independent of the natural water inflow, it only depends on the hydro system configuration.

For each time period (when the hydro system configuration has changed) we calculate three parameters for the maximum hydro generation, corresponding to the maximum, medium and minimum reservoir storage levels. Equation (2.7) shows how these parameters are computed.

$$GH_{max}^t = \sum_{r \in R \cup RR} (1 - \lambda_{C,r}^t)(1 - \lambda_{P,r}^t)P_{max,r} \quad (2.7a)$$

$$GH_{med}^t = \sum_{r \in R \cup RR} (1 - \lambda_{C,r}^t)(1 - \lambda_{P,r}^t)P_{med,r} \quad (2.7b)$$

$$GH_{min}^t = \sum_{r \in R \cup RR} (1 - \lambda_{C,r}^t)(1 - \lambda_{P,r}^t)P_{min,r} \quad (2.7c)$$

where,

- GH_{max}^t maximum hydro generation at time t considering maximum energy storage [MW -month];
- GH_{med}^t maximum hydro generation at time t considering medium energy storage [MW -month];
- GH_{min}^t maximum hydro generation at time t considering minimum energy storage [MW -month];
- $\lambda_{C,r}^t$ hydro plant r corrective maintenance rate at time t ($0 \leq \lambda_{C,r}^t \leq 1$);
- $\lambda_{P,r}^t$ hydro plant r preventive maintenance rate at time t ($0 \leq \lambda_{P,r}^t \leq 1$).

We can calculate the hydro plant power output using equation (2.1). Then, we construct a linear function to represent the maximum hydro generation as a function of the energy stored, $GH^t(EA)$.

2.3.8 Evaporation Energy Losses

The evaporation losses represent the amount of energy lost by an equivalent reservoir due to the evaporation of water in a time period t . Each reservoir has a different evaporation coefficient (according to its region). This coefficient is influenced by each season and may even be different for each month of the year.

We can calculate the losses due to water evaporation at the equivalent

reservoir using Equation (2.8):

$$EV_{max}^t = \frac{1}{10^3 \nu} \sum_{r \in R} evp_r^t A_{max,r}^t \sum_{j \in D_r} \rho_j H_{max,j} \quad (2.8a)$$

$$EV_{med}^t = \frac{1}{10^3 \nu} \sum_{r \in R} evp_r^t A_{med,r}^t \sum_{j \in D_r} \rho_j H_{med,j} \quad (2.8b)$$

$$EV_{min}^t = \frac{1}{10^3 \nu} \sum_{r \in R} evp_r^t A_{min,r}^t \sum_{j \in D_r} \rho_j H_{min,j} \quad (2.8c)$$

where,

EV_{max}^t maximum evaporation energy losses in period t [MW -month];

EV_{med}^t medium evaporation energy losses in period t [MW -month];

EV_{min}^t minimum evaporation energy losses in period t [MW -month];

evp_i^t evaporation coefficient of reservoir r in period t [mm];

A_{max}^t maximum reservoir's i surface area in period t [km^2];

A_{med}^t medium reservoir's i surface area in period t [km^2];

A_{min}^t minimum reservoir's i surface area in period t [km^2].

It is important to mention that the different surface areas correspond to the different storage levels. Also, if a downstream hydro plant j is a run-of-river plant we replace $H_{max,j}$ and $H_{min,j}$ by h_j . We construct a linear function to represent the evaporation energy losses as a function of the energy stored, $EV^t(EA)$.

2.3.9 Energy Losses to Fill Minimal Reservoir Volumes

Due to the addition of a new reservoir and/or hydro plant in the power system, new hydro generators will begin operation at a particular time period. After

a new hydro plant is built, it may be necessary to fill the minimum volume of the reservoir in order to be able to operate the system. The water that is used to fill the reservoir represents a loss for the cascade because this volume cannot be used to produce electricity anymore at downstream hydro plants. This leads to the energy losses required to fill the minimal reservoir volume.

The energy losses due to filling reservoirs volumes does not depend on the natural inflow of water. It only depends on the hydro system configuration. Equation (2.9) presents how this loss of energy should be computed:

$$EL^t = \frac{1}{\nu} \sum_{r \in R_M} \frac{V_r}{\Delta t_r} \sum_{j \in D_r^*} \rho_j H_j \quad (2.9)$$

where,

EL^t energy losses to fill reservoirs volumes in period t [MW -month];

Δt_r number of time periods that the reservoir r takes to fill minimal volume;

R_M set of reservoirs that must have their minimal volume filled;

D_r^* set of hydro plants downstream from reservoir r (without i).

If the downstream hydro plant j has reservoir then $H_j = H_{eq,j}$ otherwise $H_j = h_j$ for a run-of-river plant.

2.3.10 Water Diversion Energy Losses

Energy losses due to the diversion of water represent the amount of energy that could have been generated with the amount of water diverted immediately upstream of a hydro plant. Generally, it is necessary to divert water upstream of

a hydro plant to attend certain requirements of specific areas; e.g., sometimes in areas near the river basins water is needed for agriculture and industry purposes, because of that water has to be diverted from its normal course what represent energy losses for the aggregate reservoir. These losses depend on the hydro system configuration and on the water volumes being diverted. The energy losses are independent of natural water inflow.

There are two types of such water diversion losses, one corresponds to water diverted immediately upstream of a reservoir, which is then discounted from the controllable energy. The other corresponds to water diverted immediately upstream of a run-of-river plant, which is then discounted from the uncontrollable energy. We can compute the controllable portion of these losses using equation (2.10):

$$EDC_{max}^t = \sum_{r \in R} Q_{D,r}^t \sum_{j \in D_r} \rho_j H_{max,j} + \sum_{r \in RR} Q_{D,r}^t \sum_{j \in D_{F,r}} \rho_j H_{max,j} \quad (2.10a)$$

$$EDC_{med}^t = \sum_{r \in R} Q_{D,r}^t \sum_{j \in D_r} \rho_j H_{med,j} + \sum_{r \in RR} Q_{D,r}^t \sum_{j \in D_{F,r}} \rho_j H_{med,j} \quad (2.10b)$$

$$EDC_{min}^t = \sum_{r \in R} Q_{D,r}^t \sum_{j \in D_r} \rho_j H_{min,j} + \sum_{r \in RR} Q_{D,r}^t \sum_{j \in D_{F,r}} \rho_j H_{min,j} \quad (2.10c)$$

where,

EDC_{max}^t maximum controllable portion of energy losses from water diversion in period t [MW-month];

EDC_{med}^t medium controllable portion of energy losses from water diversion in period t [MW-month];

EDC_{min}^t minimum controllable portion of energy losses from water diversion

| | |
|-------------|---|
| | in period t [MW -month]; |
| $Q_{D,r}^t$ | water volume diverted immediately upstream of hydro plant r in period t [(m^3/s) -month]; |
| $D_{F,r}$ | set of downstream hydro plants (starting on the first reservoir) of run-of-river plant r . |

If the hydro plant j is run-of-river plant we substitute $H_{max,j}$, $H_{med,j}$ and $H_{min,j}$ by h_j . We adjust a linear function to represent the controllable portion of energy losses due to water diversion as a function of the energy stored, $EDC^t(EA)$.

The uncontrollable portion of energy losses due to water diversion does not depend on the storage, and this makes its computation easier than the controllable portion. We can compute these losses using equation (2.11).

$$EDU^t = \sum_{r \in R} Q_{D,r}^t \sum_{j \in DR_r} \rho_j h_j \quad (2.11)$$

where,

EDU^t uncontrollable portion of energy losses due to water diversion
in period t [MW -month];

2.3.11 Energy from Hydro Plants Operating with Less Machines

Before a new hydro plant with a reservoir can be used, the reservoir has to be filled with a certain volume of water. After this volume has been reached, the new hydro plant goes to state II (described earlier) and the turbine/generator

groups, that belong to the generator, will be installed during the following time periods. Each hydro plant is projected with a different number of groups, and there is a minimum number of operating groups necessary for the hydro generator to be able to produce energy.

While in state II a power generator can produce energy, but this energy is not accounted for in the controllable or uncontrollable energy parameters of the equivalent reservoir. Usually the amount of energy produced by hydro generators in state II is simply discounted from the system demand. Same way we do for small hydro plants (see Section 2.2.1.3).

2.4 Thermal Generation Power Plants

In contrast to hydro plants which must be constructed on river-basins, we have the flexibility to locate thermal generators near the load centers to reduce transmission losses and costs. A thermal plant is a power generator that uses fuel to transform energy from heat into electricity. The most common type of fuel are: natural gas, coal, oil and nuclear.

There is a cost associated with energy production from thermal plants, with one portion being proportional to fuel costs and the other involving operational and maintenance costs. Each thermal plant has its own function that relates power output and cost. Usually these cost function are nonlinear, but in order to simplify the hydro-thermal scheduling problem many models assume that thermal costs are linear functions of the power output. In our model, we assume for each thermal plant a fixed cost where c_g^t is the unit cost

to produce electricity at thermal plant g during period t .

Later on this chapter we present a model formulation for our hydro-thermal scheduling problem. We consider the thermal generation of each thermal plant as a decision variable that is restricted by the thermal power plant minimum and maximum thermal generation at each time period. Note that, these values are fixed parameters for our optimization model.

The maximum thermal generation of a thermal plant g , at a particular time period t can be computed using equation (2.12):

$$\overline{GT}_g^t = \eta_g P_g^t (1 - \lambda_{C,g}^t) (1 - \lambda_{P,g}^t) \quad (2.12)$$

where,

\overline{GT}_g^t maximum thermal generation of thermal plant g available in time period t [MW -month];

η_g thermal plant g turbine/generator efficiency;

P_g^t is the nominal power of thermal plant g at time period t [MW];

$\lambda_{C,g}^t$ is the thermal plant g corrective maintenance rate at time t ;

$\lambda_{P,g}^t$ is the thermal plant g preventive maintenance rate at time t .

During the time horizon of the hydro-thermal scheduling model, we can have generation expansion due to new thermal plants that enter the system. The thermal generation expansion differs from the one with hydro plants because they do not affect other power plant operational decisions. Besides

the addition of new thermal plants, turbines/generators can be modified during the planning horizon, which implies a fluctuation of the available thermal generation at such time periods.

2.5 Power System Demand

We model demand for energy in units of MW -month, and we do so at the resolution of demand for each region, including three levels of load, representing high (H), medium (M) and low (L). Each load level captures a period of the day for some duration. Figure (2.8) represents three load levels during a specific month, with μ representing a multiplier relative to average demand and τ representing load level durations.

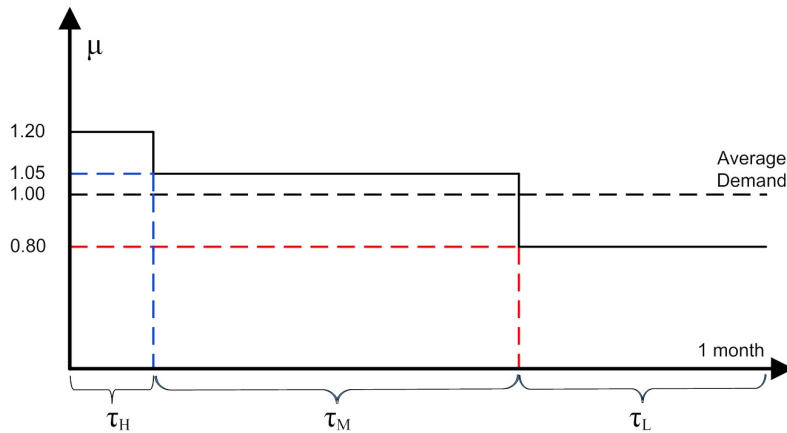


Figure 2.8: Demand at each load level

In our hydro-thermal scheduling model we discount from the demand some energy generation in a preprocessing step. As mentioned before, gener-

ation from small hydro plants is discounted from the demand. Similarly, the sum of the minimal thermal generation is discounted from the demand. Also we subtract from the demand energy produced by hydro plants operating in state II.

Consider a set of regions I that have demand and generation. The electricity demand in region i , at load level k during time period t is defined by equation (2.13):

$$d_{i,k}^t = D_i^t \mu_{i,k}^t - gsh_i^t - \sum_{g \in G_i} \underline{GT}_g^t - EF_i^t \quad (2.13)$$

where,

$d_{i,k}^t$ net demand in region i , at load level k in period t [MW -month];

D_i^t monthly average demand in region i in period t [MW -month];

$\mu_{i,k}^t$ demand multiplier for region i , load level k in period t ;

gsh_i^t energy produced by small hydro plants in region i in period t ;

\underline{GT}_g^t minimum thermal generation from plant g in period t [MW -month];

EF_i^t electricity produced by hydro plants operating in state II in region i in period t .

2.6 Energy Exchanges in the System

Transmission lines allow regions with excess supply to ship energy to regions with a deficit. Using the available hydro generation and the transmission system we can meet demand at regions far away from the river basins. Transmis-

sion capacity between regions is limited and these restrictions play an important role in the operational costs of each region. Lack of transmission capacity can lead to the need to dispatch expensive generators close to the demand or even the need to curtail load.

Figure (2.9) represents a small system with four regions and one virtual region. We can exchange energy between regions that have direct connections. For example, we can exchange energy between regions 1 and 2 directly, but in order to transfer energy between regions 2 and 3 we have to pass through region 1 first. The virtual region 5 is a point where transmission lines have a connection. In this case, to transfer power from region 3 to 4 we have to pass at least through the virtual region 5 first. A virtual region has no demand, so the energy that comes in must be equal the energy that comes out of the region. Inside the other regions the sum of electricity generation, the load curtailment and the power transfers (“-” signal if going out from the region and “+” signal if going into the region) has to equal demand.

2.7 Mathematical Model

We present a formulation of a T -stage stochastic linear program with recourse for our hydro-thermal scheduling problem with ARR.

Indices / Sets

- $i, j \in I$ regions that define each aggregate reservoir;
- $i \in I^+$ regions, including virtual regions;

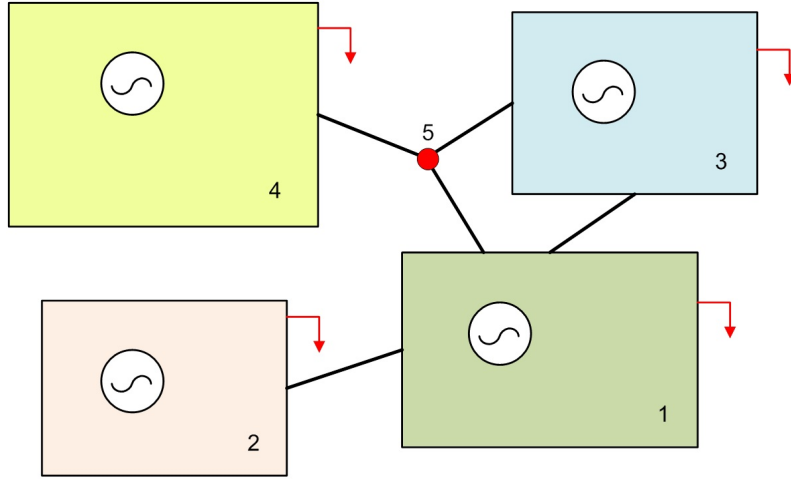


Figure 2.9: Power system with 4 regions and 1 virtual region

$(i, j) \in E$ energy-exchange pairs;

$g \in G$ thermal generators;

G_i subset of thermal generators in region i ;

$k \in K$ load levels;

$\ell \in L$ curtailment levels;

L_i subset of curtailment levels in region i ;

$t \in T$ time stages;

$\omega_t \in \Omega_t$ energy inflow scenarios at stage t ;

Data

β discount rate;

c_g^t thermal energy generation cost at plant g , at stage t [\$/MW-month];

ρ_ℓ^t load curtailment cost at curtailment level ℓ , at stage t [\$/MW-month];

$d_{i,k}^t$ demand in region i , in load level k , at stage t [MW-month];

| | |
|---------------------------|--|
| $\overline{GT}_{g,k}^t$ | upper bound on thermal generation for thermal plant g during load level k [MW -month]; |
| $\underline{p}_{i,j,k}^t$ | lower bound on energy transfers from region i to region j , during load level k at stage t [MW -month]; |
| $\overline{p}_{i,j,k}^t$ | upper bound on energy transfers from region i to region j , during load level k at stage t [MW -month]; |
| \underline{x}_i^t | lower bound on energy storage for region i at stage t [MW -month]; |
| \overline{x}_i^t | upper bound on energy storage for region i at stage t [MW -month]; |
| $\overline{u}_{\ell,k}^t$ | upper bound on unmet demand at curtailment level ℓ , during load level k at stage t [MW -month]; |
| $f_1^t(\cdot)$ | right-hand side (RHS) of the energy-balance constraints. It is a function of storage from stage $t - 1$ and energy inflow available at current stage t [MW -month]; |
| $f_{2,k}^t(\cdot)$ | RHS of demand satisfaction constraints, for load level k . It is a function of storage from stage $t - 1$ and energy inflow available at current stage t [MW -month]; |
| $f_{3,k}^t(\cdot)$ | RHS of the hydro-generation constraint, for load level k . It is a function of storage from stage $t - 1$ and energy inflow available at current stage t [MW -month]; |

Random Variables

| | |
|------------------|--|
| $b_{i,t}^\omega$ | energy inflow in region i , at stage t , under energy inflow scenario ω [MW -month]; |
|------------------|--|

Decision Variables

| | |
|----------------|---|
| x_i^t | energy storage in aggregate reservoir i , at stage t [MW -month]; |
| $gh_{i,k}^t$ | hydro generation at aggregate reservoir i , during load level k , at stage t [MW -month]; |
| s_i^t | water spilled from the aggregate reservoir i , at stage t [MW -month]; |
| $gt_{g,k}^t$ | thermal generation at generator g , during load level k , at stage t [MW -month]; |
| $u_{\ell,k}^t$ | unmet demand at curtailment level ℓ , during load level k , at stage t [MW -month]; |
| $p_{i,j,k}^t$ | energy transfers from region i to region j , during load level k , at stage t [MW -month]; |
| $y_{i,k}^t$ | surplus variable to represent the energy generation exceeding the demand in region i , during load level k , at stage t [MW -month]. |

In this model we assume that the energy inflows are known at the beginning of the current stage t .

Formulation

$$z^* = \min \sum_{i \in I} \sum_{k \in K} \left[\sum_{g \in G_i} c_g^1 g t_{g,k}^1 + \sum_{\ell \in L_i} \rho_\ell^1 u_{\ell,k}^1 \right] + (1 + \beta)^{-1} \mathbb{E}_{b_2 | b_1} h_2(x^1, b_2) \quad (2.14a)$$

$$\text{s.t. } x_i^1 + \sum_{k \in K} g h_{i,k}^1 + s_i^1 = f_1^1(x_i^0, b_{i,1}) \quad \forall i \in I \quad (2.14b)$$

$$g h_{i,k}^1 + \sum_{g \in G_i} g t_{g,k}^1 + \sum_{\ell \in L_i} u_{\ell,k}^1 - \sum_{j: (i,j) \in E} p_{i,j,k}^1 + \quad (2.14c)$$

$$+ \sum_{j: (j,i) \in E} p_{j,i,k}^1 - y_{i,k}^1 = f_{2,k}^1(x_i^0, b_{i,1}) \quad \forall k \in K, \forall i \in I$$

$$g h_{i,k}^1 - y_{i,k}^1 \leq f_{3,k}^1(x_i^0, b_{i,1}) \quad \forall k \in K, \forall i \in I \quad (2.14d)$$

$$\sum_{i: (i,j) \in E} (p_{i,j,k}^1 - p_{j,i,k}^1) = 0 \quad \forall k \in K, \forall j \in I^+ \setminus I \quad (2.14e)$$

$$g h_{i,k}^1, s_i^1, y_{i,k}^1 \geq 0 \quad \forall k \in K, \forall i \in I \quad (2.14f)$$

$$0 \leq g t_{g,k}^1 \leq \overline{G T}_{g,k}^1 \quad \forall k \in K, \forall g \in G \quad (2.14g)$$

$$\underline{p}_{i,j,k}^1 \leq p_{i,j,k}^1 \leq \overline{p}_{i,j,k}^1 \quad \forall k \in K, \forall (i,j) \in E \quad (2.14h)$$

$$0 \leq u_{\ell,k}^1 \leq \overline{u}_{\ell,k}^1 \quad \forall k \in K, \forall \ell \in L \quad (2.14i)$$

$$\underline{x}_i^1 \leq x_i^1 \leq \overline{x}_i^1 \quad \forall i \in I. \quad (2.14j)$$

Note that x_i^0 is the given energy storage amount available to be used at the first stage. For stages $t = 2, \dots, T$ we have:

$$h_t(x^{t-1}, b_t^\omega) = \min \sum_{i \in I} \sum_{k \in K} \left[\sum_{g \in G_i} c_g^t g_{g,k}^t + \sum_{\ell \in L_i} \rho_\ell^t u_{\ell,k}^t \right] + (1 + \beta)^{-1} \mathbb{E}_{b_{t+1}|b_1, \dots, b_t} h_{t+1}(x^t, b_{t+1}) \quad (2.15a)$$

$$\text{s.t. } x_i^t + \sum_{k \in K} g h_{i,k}^t + s_i^t = f_1^t(x_i^{t-1}, b_{i,t}^\omega) \quad \forall i \in I \quad (2.15b)$$

$$g h_{i,k}^t + \sum_{g \in G_i} g t_{g,k}^t + \sum_{\ell \in L_i} u_{\ell,k}^t - \sum_{j:(i,j) \in E} p_{i,j,k}^t + \quad (2.15c)$$

$$+ \sum_{j:(i,j) \in E} p_{j,i,k}^t - y_{i,k}^t = f_{2,k}^t(x_i^{t-1}, b_{i,t}^\omega) \quad \forall k \in K, \forall i \in I$$

$$g h_{i,k}^t - y_{i,k}^t \leq f_{3,k}^t(x_i^{t-1}, b_{i,t}^\omega) \quad \forall k \in K, \forall i \in I \quad (2.15d)$$

$$\sum_{i:(i,j) \in E} (p_{i,j,k}^t - p_{j,i,k}^t) = 0 \quad \forall k \in K, \forall i \in I^+ \setminus I \quad (2.15e)$$

$$g h_{i,k}^t, s_i^t, y_{i,k}^t \geq 0 \quad \forall k \in K, \forall i \in I \quad (2.15f)$$

$$0 \leq g t_{g,k}^t \leq \overline{G T}_{g,k}^t \quad \forall k \in K, \forall g \in G \quad (2.15g)$$

$$\underline{p}_{i,j,k}^t \leq p_{i,j,k}^t \leq \overline{p}_{i,j,k}^t \quad \forall k \in K, \forall (i,j) \in E \quad (2.15h)$$

$$0 \leq u_{\ell,k}^t \leq \overline{u}_{\ell,k}^t \quad \forall k \in K, \forall \ell \in L \quad (2.15i)$$

$$\underline{x}_i^t \leq x_i^t \leq \overline{x}_i^t \quad \forall i \in I. \quad (2.15j)$$

We use $f_1^t(\cdot)$, $f_{2,k}^t(\cdot)$ and $f_{3,k}^t(\cdot)$ on the RHS of the constraints for simplicity. We can represent these functions in more detail as:

$$f_1^t(x_i^{t-1}, b_{i,t}^\omega) = \delta_i^t x_i^{t-1} + EC_i^t - EM_i^t - EV_i^t - EDC_i^t - EL_i^t \quad (2.16)$$

$$f_{2,k}^t(x_i^{t-1}, b_{i,t}^\omega) = d_{i,k}^t - (EU_i^t + EM_i^t - EDU_i^t) \tau_{i,k}^t \quad (2.17)$$

$$f_{3,k}^t(x_i^{t-1}, b_{i,t}^\omega) = (GH_i^t - EU_i^t - EM_i^t + EDU_i^t) \tau_{i,k}^t \quad (2.18)$$

where,

- EC_i^t controllable energy in aggregate reservoir i at stage t [MW -month].
 $EC_i^t = \gamma_i b_{i,t}^\omega$;
- EM_i^t minimal outflow energy at aggregate reservoir i at stage t [MW -month].
This energy is uncontrollable and cannot be stored. $EM_i^t = EM_i^t(\delta_i^t x_i^{t-1})$;
- EV_i^t energy loss due to water evaporation at aggregate reservoir i at stage t
[MW -month]. $EV_i^t = EV_i^t(\delta_i^t x_i^{t-1})$;
- EDC_i^t energy losses from controllable water diversion at aggregate reservoir i at
stage t [MW -month]. $EDC_i^t = EDC_i^t(\delta_i^t x_i^{t-1})$;
- EL_i^t energy losses to fill new hydro reservoirs in aggregate reservoir i
at stage t [MW -month]. This term does not depend on the previous
stored energy from stage $t - 1$;
- EU_i^t uncontrollable energy produced by run-of-river hydro plants
in aggregate reservoir i , at stage t [MW -month]. $EU_i^t = (1 - \gamma_i) b_{i,t}^\omega$;
- EDU_i^t energy losses from uncontrollable water diversion at aggregate reservoir i
at stage t [MW -month];
- GH_i^t maximum hydro generation available at aggregate reservoir i at stage t
[MW -month]. $GH_i^t = GH_i^t(\delta_i^t x_i^{t-1})$;
- δ_i^t correction factor used to incorporate changes in the hydro system
configuration in aggregate reservoir i at stage t ;
- $\tau_{i,k}^t$ load level k duration fraction at region i at stage t ;
- γ_i fraction of the energy inflow that represents controllable energy
at aggregate reservoir i ;

The objective function in equation (2.14a) represents the sum of the first stage costs and the expected discounted future costs. Similarly, equation (2.15a) contains the objective function of the problem at stage t . We have four sets of structural constraints at each stage. Equations (2.14b) and (2.15b) represent the set of energy balance constraints at stage 1 and t respectively. We have a total of $|I|$ constraints of this type at each stage, the purpose of this constraint is to balance the aggregate reservoir energy storage levels. It considers the inventory the aggregate reservoir i will carry on to the next stage plus the amount of energy production by hydro plants and the energy spilled from the aggregate reservoir at that stage. The associated sum must equal the energy storage from the previous stage minus the losses. Equations (2.14c) and (2.15c) represent the set of demand satisfaction constraints. We have a total of $|I| \cdot |K|$ constraints of this type at each stage. The demand satisfaction constraints require that at each stage, and each region i and each load level k , the amount of energy produced plus the demand curtailment and energy exchanges must equal to the system demand for region i during load level k . The third set of structural constraints in (2.14d) and (2.15d) restricts the amount of hydro generation that the model can use from the available energy storage at aggregate reservoir i , during load level k at stage t . We have a total of $|I| \cdot |K|$ constraints of this type at each stage. The last set of structural constraints in (2.14e) and (2.15e) enforces energy balance through virtual regions and we have a total of $|K| \cdot (|I^+| - |I|)$ constraints of this type at each stage. The virtual regions do not have demand, which means that the

energy that comes in the virtual region has to go out. The other constraints represent simple bounds for the problem at each stage.

Chapter 3

A Sampling-Based Decomposition Algorithm

3.1 Literature Survey

We focus on multi-stage stochastic linear programs with recourse. According to Chen and Powell (1999), this class of problems has the following characteristics:

1. the term stage represents a time period;
2. the beginning of the first stage is viewed as here and now;
3. at the beginning of each stage we know deterministically all the parameters in that stage; the random parameters in future time stages is only known probabilistically;
4. we must make decisions at each time stage that can depend only on the realizations observed up to that stage and conditional distribution for parameters in future stages;
5. after the random parameters are realized recourse actions can be taken to compensate for decisions made prior to this particular stage; and,
6. the total cost measures the quality of the decisions made.

Until the mid 1980s, most research on hydro-thermal scheduling under uncertainty used stochastic dynamic programming (SDP) techniques as the solution method. The main drawback of SDP is the “curse of dimensionality” of dynamic programming (DP) that makes the problem intractable when the dimension of the state vector is medium or large. DP algorithm constructs the future cost function by discretizing the state variables at each stage into a set of finite values. Then the algorithm proceeds backward in time using Bellman’s recursion; see Bellman (1957) and Bertsekas (2005). Methods to overcome DP’s “curse of dimensionality” and solve real-size instances of this type of problem were necessary. Benders’ decomposition algorithm (Benders, 1962) lead the way.

Benders’ decomposition algorithm is at the core of the solution methods for multi-stage stochastic linear programs. In the literature, there are two different types of algorithms used to address this class of problems: scenario-based decomposition algorithms and sampling-based decomposition algorithms.

Scenario-based methods first select a modest number of scenarios to represent the probability distribution. The problem, after the scenarios are chosen, is considered as a large deterministic linear program. The optimal solution obtained for this problem is exact, but it is only an approximation of the true original problem, assuming the probability distribution was approximated. Two of the most well known scenario-based decomposition algorithms are the L-shaped method of Van Slyke and Wets (1969) for two-stage stochastic linear programs, and the nested Benders’ decomposition algorithm

(Birge, 1985) for problems with more than two stages. According to Rebenack (2010), one advantage of this method is that more uncertainties can be modeled and represented at the same time. In a hydro-thermal scheduling problem, for example, one can consider randomness not just on the water inflows, but also on electricity demand, fuel prices and other parameters, as long as a modest number of overall scenarios are considered.

An important application of a Benders-style algorithm to a hydro-thermal scheduling problem is presented by Pereira and Pinto (1985). Using a scenario-based method, the authors attempt to solve three- and five-stage hydro-thermal scheduling problems with two possible random realizations at each stage. They presented the dynamic dual programming (DDP) algorithm that later was revised in Velasquez et al. (1999) to be valid for problems with more than two stages. Morton (1996) presents enhancements to the nested Benders' decomposition algorithm, and an application of this algorithm to hydro-thermal scheduling at Pacific Gas and Electric Company can be found in Jacobs et al. (1995).

We focus on a sampling-based decomposition algorithm (SBDA) for multi-stage stochastic linear programs. An SBDA differs from a scenario-based decomposition algorithm, in that it considers scenario trees whose size is too large for a scenario-based algorithm. Sample observations of the random parameters are drawn, at each time stage. An SBDA proceeds, pursuing convergence in some probabilistic sense, until it finally reaches a stopping criterion (Chen and Powell, 1999). The idea to introduce sampling methods

into the nested Benders' decomposition algorithm gave origin to SBDA. The first SBDA to appear in the literature is called stochastic dual dynamic programming (SDDP), presented in Pereira (1989) and Pereira and Pinto (1991). SDDP is one of the most used and well known SBDA, and the motivation for its development was hydro-thermal scheduling.

Since the early 1990s SBDAs have received considerable attention from the stochastic programming community. SDDP-related algorithms such as abridged nested decomposition (AND) by Donohue (1996) and Donohue and Birge (2006), the convergent cutting-plane and partial-sampling (CUPPS) algorithm of Chen and Powell (1999) and the dynamic outer approximation sampling algorithms (DOASAs) by Philpott and Guan (2008) and Philpott and de Matos (2010) were developed to improve SDDP's computational efficiency. In Donohue and Birge (2006) the authors present a different sampling scheme and computational results for AND and SDDP applied to a dynamic vehicle allocation problem with uncertain demand. Their AND algorithm also was applied to a hydro-thermal scheduling problem in the Colombian Power System (Supatgiat, 2001). The DOASA, first presented in Philpott and Guan (2008) to deal with a hydro-thermal scheduling problem, was also used to solve a production planning problem for the dairy industry in New Zealand (Guan and Philpott, 2009).

Since the appearance of SDDP, convergence properties of these algorithms have been studied in the literature. Convergence analyses and statements are shown in Donohue (1996); Chen and Powell (1999); Linowsky and

Philpott (2005) and, Philpott and Guan (2008). A convergence proof was presented by Chen and Powell (1999) for the CUPPS algorithm. Later, Linowsky and Philpott (2005) extend the proof for SBDA related algorithms in general. But according to Philpott and Guan (2008) the convergence proofs presented until that time were weak in the sense that they require an implicit assumption of independence between sampled random variables and convergent subsequences of algorithm iterates. Philpott and Guan (2008) present a simpler convergence proof for SBDA that does not require the independence assumption and holds in general, based on an idea of Donohue (1996) that states: “*Finite convergence of this algorithm follows from the finite convergence of the Nested Decomposition algorithm, since the scenarios from which the optimality cuts are generated are re-sampled at each iteration*”.

Shapiro (2011) discusses statistical properties of the SDDP algorithm. In his work, considering stage-wise independence, a sample from the distribution of the original problem is taken to create a finite sample average approximation (SAA) of the true problem. SDDP is studied for this SAA problem and an extension for the risk-averse case, using conditional value at risk (see Rockafellar and Uryasev, 2000) is presented. Shapiro states that the stopping criteria proposed by Pereira and Pinto (1991) and used by Donohue and Birge (2006) does not guarantee correct stopping or reasonable solution quality for SBDA. He presents another stopping criterion and argues that it is more meaningful from a statistical point of view.

Stopping criteria for SDDP were also studied by Homem-de-Mello et al. (2011). The authors present the stopping criterion proposed in Pereira and Pinto (1991) as a hypothesis test. They argue that the original stopping criterion allows the algorithm to terminate sooner than it should, depending on the sample size chosen, without achieving good solutions. The authors then suggest a modification of the original criterion to alleviate the premature stopping issue.

Monte Carlo methods usually define the sampling schemes for SBDA (Pereira and Pinto, 1991; Donohue and Birge, 2006; Chen and Powell, 1999; Philpott and Guan, 2008). Homem-de-Mello et al. (2011) present two different sampling schemes for SBDA, randomized quasi-Monte Carlo (QMC) and the Latin hypercube sampling (LHS) schemes. The authors apply SDDP with these alternative sampling schemes to a three-year horizon hydro-thermal scheduling problem and achieve more consistent operational policies than with SDDP with traditional Monte Carlo methods.

SBDA avoids the DP “curse of dimensionality” by constructing an approximation of the future cost function. The algorithm approximates the future cost function with a piecewise linear functions (Benders’ cuts) that are iteratively added as the algorithm proceeds. In this chapter an SDPA is presented and the ideas behind the approximation of the future cost function are described. Then we turn to the algorithm’s ability to share cuts under an inter-stage dependency model (Infanger and Morton, 1996). We conclude the chapter with an extension of the cut-sharing procedure to deal with the ARR

model of Chapter 2, coupled with forecasts of natural inflow of water at each hydro plant.

3.2 A Two Stage Stochastic Linear Program

Towards introducing an SBDA in the multi-stage setting, we begin with the simpler non-sampling based algorithm for two-stage linear programs of the form:

$$\begin{aligned}
 z^* = \min_{x_1, x_2} \quad & c_1x_1 + c_2x_2 \\
 \text{s.t.} \quad & A_1x_1 = B_1x_0 + b_1 \\
 & -B_2x_1 + A_2x_2 = b_2 \\
 & x_1 \geq 0, x_2 \geq 0,
 \end{aligned} \tag{3.1}$$

where, for $t = 1, 2$, matrix A_t has m_t rows and d_t columns, and the remaining matrices and vectors are dimensioned to conform. In order to illustrate Benders' scheme for a two-stage linear program we begin by assuming that matrices A_1 , B_1 , A_2 and B_2 are deterministically known as well as the vectors b_1 , b_2 and x_0 .

Model (3.1) can be partitioned by stage following the idea behind Benders' decomposition. We can re-write the problem as a two-stage model with a recourse function in the following way:

$$\begin{aligned}
 z^* = \min_{x_1} \quad & c_1x_1 + h(x_1, b_2) \\
 \text{s.t.} \quad & A_1x_1 = B_1x_0 + b_1 \\
 & x_1 \geq 0,
 \end{aligned} \tag{3.2}$$

where, we define the recourse function as:

$$\begin{aligned}
 h(x_1, b_2) = \min_{x_2} & \quad c_2 x_2 \\
 \text{s.t.} & \quad A_2 x_2 = B_2 x_1 + b_2 \quad : \quad \pi \\
 & \quad x_2 \geq 0.
 \end{aligned} \tag{3.3}$$

The recourse function $h(x_1, b_2)$ represents the cost incurred in the second stage given the first stage decisions x_1 . One way to view the recourse function is via the dual program of (3.3):

$$\begin{aligned}
 h(x_1, b_2) = \max_{\pi} & \quad \pi(B_2 x_1 + b_2) \\
 \text{s.t.} & \quad \pi A_2 \leq c_2.
 \end{aligned} \tag{3.4}$$

Imagine enumerating all the extreme points of model (3.4)'s feasible region, $\pi^{(1)}, \dots, \pi^{(l)}$, where l is the number of extreme points of the set $\Pi = \{\pi : \pi A_2 \leq c_2\}$. We can then write:

$$h(x_1, b_2) = \max_{1 \leq i \leq l} \pi^{(i)}(B_2 x_1 + b_2). \tag{3.5}$$

Model (3.5) can be re-written, by adding a new variable θ to represent the optimal value:

$$\begin{aligned}
 h(x_1, b_2) = \min_{\theta} & \quad \theta \\
 \text{s.t.} & \quad \theta \geq \pi^{(1)}(B_2 x_1 + b_2) \\
 & \quad \theta \geq \pi^{(2)}(B_2 x_1 + b_2) \\
 & \quad \vdots \\
 & \quad \theta \geq \pi^{(l)}(B_2 x_1 + b_2).
 \end{aligned} \tag{3.6}$$

We can then re-write model (3.1) based on (3.2) and (3.6):

$$\min_{x_1, \theta} \quad c_1 x_1 + \theta \quad (3.7a)$$

$$\text{s.t.} \quad A_1 x_1 = B_1 x_0 + b_1 \quad (3.7b)$$

$$-\vec{G}x_1 + e\theta \geq \vec{g} \quad (3.7c)$$

$$x_1 \geq 0, \quad (3.7d)$$

where, \vec{G} represents the cut gradient matrix with l rows given by $\pi^{(i)}B_2$, $i = 1, 2, \dots, l$, and \vec{g} represents the l -vector of cut intercepts with components given by $\pi^{(i)}b_2$, $i = 1, 2, \dots, l$. Model (3.6), as stated, is called the full master program and has optimal value z^* . If instead (3.7c) only contains a subset of the cut constraints the model (3.6) is called the relaxed master program, and has optimal value $\underline{z} \leq z^*$.

Benders' algorithm decomposes (3.1) into two separate problems. A relaxed master problem that represents the first stage together with a subset of optimality cuts (3.7c), and a subproblem that represents the second stage used to determine the value of $h(x_1, b_2)$ and generates dual extreme points. Model (3.7) represents the master problem and models (3.3), (3.4) and (3.5) represent different formulations for the subproblem. Note that the optimality cuts are added iteratively, to the relaxed master and so at an iteration of the algorithm the set of cuts in the relaxed master is just the collection of the cuts generated so far. Let the optimal solution of the relaxed master, at a particular iteration of the algorithm, be $(\hat{x}_1, \hat{\theta})$. Because the master has only a subset of the cuts, we have $\underline{z} = c_1 \hat{x}_1 + \hat{\theta} \leq z^*$.

Now let b_2 be a random vector with sample space Ω , and a sample point (scenario) in Ω denoted ω . A realization of the random vector b_2 is denoted b_2^ω and when the scenario tree is finite, b_2 's probability mass function is denoted $\mathbb{P}(b_2 = b_2^\omega) = p_2^\omega$.

The first stage decision variables, x_1 , have to be chosen prior to the realization of the second stage random vector b_2 . Note that we are only considering uncertainty in b_2 , although this model could be extended to address uncertainties in other second stage parameters. The resulting two-stage stochastic linear program (SLP-2) with random vector b_2 , can be formulated according to model (3.8):

$$\begin{aligned}
z^* = \min_{x_1, x_2^\omega} \quad & c_1 x_1 + \sum_{\omega \in \Omega} p_2^\omega c_2 x_2^\omega \\
\text{s.t.} \quad & A_1 x_1 = B_1 x_0 + b_1 \\
& -B_2 x_1 + A_2 x_2^1 = b_2^1 \\
& -B_2 x_1 + A_2 x_2^2 = b_2^2 \\
& \vdots \quad \ddots \quad \vdots \\
& -B_2 x_1 + A_2 x_2^{|\Omega|} = b_2^{|\Omega|} \\
& x_1 \geq 0, x_2^\omega \geq 0 \quad \forall \omega \in \Omega.
\end{aligned} \tag{3.8}$$

Model (3.8) contains a block diagonal structure with connecting variables (x_1). This type of problem can be solved by Benders' decomposition or equivalently the L-shape method. We can re-write model (3.8) as:

$$\begin{aligned}
z^* = \min_{x_1} \quad & c_1 x_1 + \mathbb{E}_{b_2|b_1} h(x_1, b_2) \\
\text{s.t.} \quad & A_1 x_1 = B_1 x_0 + b_1 \\
& x_1 \geq 0.
\end{aligned} \tag{3.9}$$

For each realization of b_2 we have a problem similar to (3.3). We can

apply the same idea described before in the stochastic case. We now state the Benders' decomposition algorithm for SLP-2.

Benders' Decomposition for SLP-2:

Input: Instance of model (3.8) with a scenario tree with modest-sized $|\Omega|$.
Bound M , $\mathbb{E}h(x_1, b_2) \geq M$, $\forall x_1$.

Output: Optimal solution x_1^* .

0. let $k = 0$;
append lower bounding cuts to the relaxed master program $\theta \geq -M$;
1. solve the relaxed master program, i.e., (3.7), and obtain (x_1^k, θ^k) ;
let $\underline{z}^k = c_1 x_1^k + \theta^k$;
2. do $\omega \in \Omega^k$
form RHS of model (3.3): $B_2[x_1]^k + b_2^\omega$;
solve and obtain $([x_2^\omega]^k, [\pi^\omega]^k)$;
enddo
let $\bar{z}^k = c_1 x_1^k + \sum_{\omega \in \Omega} p_2^\omega c_2 [x_2^\omega]^k$;
3. if $\bar{z}^k = \underline{z}^k$, then stop and output $[x_1^k]$;
4. Form cut gradient and intercept:
 $G^k = \sum_{\omega \in \Omega} p_2^\omega [\pi^\omega]^k B_2$ and
 $g^k = \sum_{\omega \in \Omega} p_2^\omega [\pi^\omega]^k b_2$;
augment \vec{G} and \vec{g} with G^k and g^k ;
5. let $k = k + 1$; goto step 1; \square

3.3 A Multi-Stage Stochastic Linear Program

A T -stage stochastic linear program (SLP-T) with recourse may be formulated as follows:

$$\begin{aligned} z^* = \min_{x_1} \quad & c_1 x_1 + \mathbb{E}_{b_2|b_1} h_2(x_1, b_2) \\ \text{s.t.} \quad & A_1 x_1 = B_1 x_0 + b_1 \\ & x_1 \geq 0, \end{aligned} \tag{3.10}$$

where for $t = 2, \dots, T$,

$$\begin{aligned} h_t(x_{t-1}, b_t) = \min_{x_t} \quad & c_t x_t + \mathbb{E}_{b_{t+1}|b_1, \dots, b_t} h_{t+1}(x_t, b_{t+1}) \\ \text{s.t.} \quad & A_t x_t = B_t x_{t-1} + b_t \\ & x_t \geq 0. \end{aligned} \tag{3.11}$$

The matrices A_t have m_t rows and d_t columns, $t = 1, \dots, T$, and the remaining matrices and vectors are dimensioned to conform. The sample space for stage t is denoted Ω_t , and a sample point (scenario) in Ω_t is denoted ω_t . A stage $t > 1$ scenario, ω_t , has a unique stage $t - 1$ ancestor denoted $a(\omega_t)$. And, a stage $t < T$ scenario has a set of stage $t + 1$ descendants denoted $\Delta(\omega_t)$. A realization of the random vector b_t is denoted $b_t^{\omega_t}$ and when the scenario tree is finite, b_t 's probability mass function is denoted $\mathbb{P}(b_t = b_t^{\omega_t}) = p_t^{\omega_t}$. When the tree is again finite and $\omega_{t+1} \in \Delta(\omega_t)$, we can define the conditional probability mass function $\mathbb{P}(b_{t+1} = b_{t+1}^{\omega_{t+1}} | b_t = b_t^{\omega_t}) = p_{t+1}^{\omega_{t+1}|\omega_t}$ via

$$p_{t+1}^{\omega_{t+1}|\omega_t} = \frac{p_{t+1}^{\omega_{t+1}}}{\sum_{\omega'_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega'_{t+1}}}.$$

In the setting of a hydro-scheduling model with aggregate reservoir representation (ARR), models (3.10) and (3.11) are general formulations of

models (2.14) and (2.15). The x_t vector represents all stage t decision variables including hydro generation, hydro energy storage, water spills from the reservoirs, thermal generation, energy transfer among regions and load curtailment. An A matrix is the stage t model's structural constraint matrix that captures energy-balance constraints, demand satisfaction constraints, maximum hydro generation constraints and energy transfer constraints. The b_t vector represents the deterministic demand, the random energy inflow that includes the controllable energy inflow (EC_i^t) and the uncontrollable energy inflow (EU_i^t), the energy losses to fill minimal reservoir volumes (EL_i^t) and the uncontrollable portion of energy losses from water diversion (EDU_i^t). The term $B_t x_{t-1}$ includes the energy storage that is carried forward from stage $t - 1$ and is available at stage t . This term further captures the fact that terms like the minimal energy outflow (EM_i^t), the evaporation energy losses (EV_i^t), the controllable portion of energy losses from water diversion (EDC_i^t) and the maximum hydro generation (GH_i^t) are all modeled as linear functions that depend on the storage in aggregate reservoir i in the previous stage, e.g., $EM_i^t(\delta_i^t x_i^{t-1}) = aEM_i^t \delta_i^t x_i^{t-1} + bEM_i^t$, where aEM_i^t and bEM_i^t are the slope and the intercept of the linear function $EM_i^t(\delta_i^t x_i^{t-1})$ of aggregate reservoir i in time period t . The expression $h_{t+1}(x_t, b_{t+1})$ represents the future cost function.

Costs include thermal energy generation costs and penalty costs for energy deficits. To handle end-effects, we can specify the value of ending energy inventories via h_{T+1} . Or, alternatively we can let $h_{T+1} = 0$ and instead specify final-period inventory constraints. We assume that the model has

so-called relatively complete recourse, i.e., for any realization of the random inflows and for any history of feasible decisions, the stage t model defined by (3.11) is feasible.

As indicated above, the random parameters in model (3.10) involve energy inflow into aggregate reservoirs, represented by the random vectors b_t . Figure 3.1 shows a three-stage scenario tree.

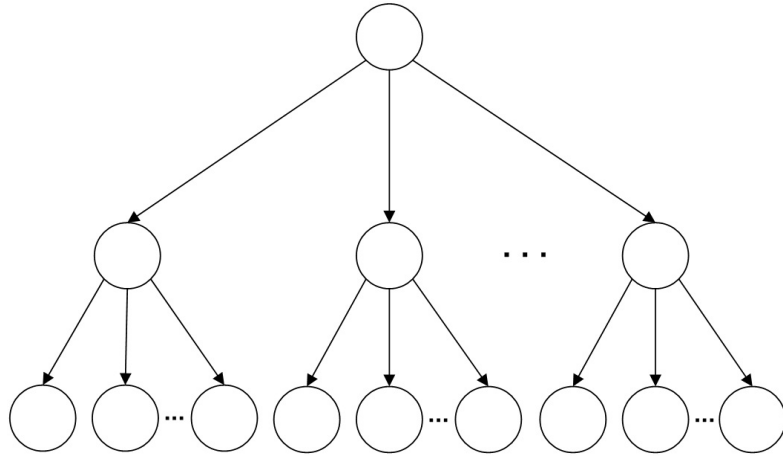


Figure 3.1: Depiction of a three-stage scenario tree

It is important to understand the relative timing of observations of the random inflows and the model's decisions. The energy inflows b_1 in the first time period (month) are assumed known when we make decision x_1 , but only a probability distribution governing future energy inflows (b_2, b_3) is assumed known. The vector realization of b_2 is known when decisions x_2 must be made for the second month. At this time, we have an updated conditional probability distribution governing b_3 given b_2 . In the first month, $B_1 x_0$ represents the

(given) initial energy inventories in the ARR and in subsequent stages $B_t x_{t-1}$ captures the “state” of the system, i.e., the inventories carried forward from one month to the next. We note that while the first stage energy inflows, b_1 , are deterministically known at the beginning of the model, it is notationally convenient to condition on b_1 as we have in (3.10) and (3.11). Moreover, we may wish to solve a *family* of models with different first stage energy inflows, b_1 . The output of an algorithm we describe later in Section 3.6 is a set of cuts in each stage, which approximate expected future cost functions. Solving a family of models for a range of first stage vectors, b_1 , allows us to have a richer approximation of the first stage expected future cost function, across a larger domain of interest.

3.4 Stochastic Process Governing Energy Inflows

The stochastic process that governs the water, and therefore the energy, inflows is one of the most important characteristics of the hydro-thermal scheduling problem. In this section we assume that the stochastic model for b_t is in the same units as that of the constraints. That assumption is implicit in the $B_t x_{t-1} + b_t$ RHS of model (3.11). So, if our flow conservation constraint is in units of water volume then our forecasting for $\{b_t\}$ is also in units of water. Or, if we have formulated the model using an ARR with flow conservation in units of energy then our forecasting of $\{b_t\}$ is also in units of energy. Given our context from Chapter 2, we assume the latter situation here. Later in Section 3.8, we consider the variant in which inflow forecasting is in units of

water volume but we formulate the stochastic program in units of energy.

3.4.1 Interstage Independent Case

The simplest way to represent the random variables representing energy inflows in a scenario tree is to assume that the vectors b_t , $t = 2, \dots, T$, are interstage independent. When we assume independence from one month to the next we mean that the realization of the random variable at a future stage has no relationship with the realization of random variables from previous stages.

3.4.2 Interstage Dependent Case

In the interstage dependent case, we assume the energy-inflow vectors satisfy the following process:

$$b_t = \sum_{j=1}^{t-1} R_j^t b_j + \eta_t, \quad t = 2, \dots, T, \quad (3.12a)$$

$$\eta_t, \quad t = 2, \dots, T, \text{ are independent.} \quad (3.12b)$$

The matrices R_j^t , $j = 1, \dots, t-1$, $t = 2, \dots, T$, are assumed known, presumably because they have been estimated using historical data. Dependency model (3.12) generalizes the periodic autoregressive model (PAR) in which R_j^t exhibits seasonality. For example, referring to de Matos et al. (2008), we can appropriately define the R_j^t matrix and η_t random vectors to have: (i) 12 seasons in a model with monthly time increments, (ii) the length of the lag depend on the month (because some matrices satisfy $R_j^t = 0$), (iii) the PAR model use centered terms $b_j - \mathbb{E}b_j$ in place of b_j (because the associated

deterministic terms can be absorbed in η_t), and (iv) the distribution of η_t can be that of a multivariate shifted lognormal. Note that if we have a lag of a specific order (e.g., six monthly periods) then we can alter (3.12a) so that in the initial periods the inflow vector depends on the “prehistory” of the optimization model (e.g., the inflows in the six months predating the optimization model’s first month).

In the general statement of the stage $t - 1$ problem (i.e., a version of model (3.11) shifted by one stage), the expectation operator is $\mathbb{E}_{b_t|b_1,\dots,b_{t-1}}$. In light of the autoregressive-style dependency process specified above, the first term on the RHS of (3.12a) is deterministic given that we condition on b_1, \dots, b_{t-1} . So, given these values of the inflows in stages $1, \dots, t - 1$, the expectation amounts to integrating with respect to the distribution of η_t . In other words, under (3.12) we can rewrite the conditional future cost function in model (3.11) as

$$\mathbb{E}_{b_{t+1}|b_1,\dots,b_t} h_{t+1}(x_t, b_{t+1}) = \mathbb{E}_{\eta_{t+1}} h_{t+1}(x_t, b_{t+1}(b_1, \dots, b_t, \eta_{t+1})). \quad (3.13)$$

3.5 From an Infinite to a Finite Sampled Scenario Tree

When the probability distribution governing b_t is continuous then the scenario tree in Figure 3.1 is also infinite, with a continuum of descendent nodes at each stage. The sampling-based decomposition algorithm (SBDA) we describe in Section 3.6 is not designed to handle a stochastic program with an infinite scenario tree. So, we first form a finite scenario tree by sampling.

3.5.1 Sampled Scenario Tree under Interstage Independence (SSTII)

First assume that the vectors b_t , $t = 2, \dots, T$, are interstage independent. (Note that even though they are interstage independent, the components of the vector b_t within a time stage can be dependent.) In this case we form a sample scenario tree in the following fashion.

Input: Multivariate distributions $F_t(\cdot)$ governing b_t , $t = 2, \dots, T$, which are assumed to be interstage independent. A procedure for drawing i.i.d. observations from each F_t . Branch size $n(t)$ for each stage, $t = 2, \dots, T$, e.g., $n(t) = 20 \forall t$.

Output: A finite sampled scenario tree with the property that its inflow vectors exhibit interstage independence.

1. Let b_1 denote the known first stage realization.
2. Sample $b_2^1, \dots, b_2^{n(2)}$, i.i.d. observations from F_2 . Let these denote the descendent nodes of b_1 .
3. Sample $b_3^1, \dots, b_3^{n(3)}$, i.i.d. observations from F_3 , independent of those formed in stage 2. Let these denote the *same* set of descendent nodes for each of the $n_2 = n(2)$ leaf nodes $\{b_1\} \times \{b_2^1, \dots, b_2^{n(2)}\}$.
- ⋮
- t. Sample $b_t^1, \dots, b_t^{n(t)}$, i.i.d. observations from F_t , independent of those formed in stages $2, \dots, t - 1$. Let these denote the same set of descendent nodes for each of the $n_{t-1} = \prod_{t=2}^{t-1} n(t)$ leaf nodes given by $\{b_1\} \times \{b_2^1, \dots, b_2^{n(2)}\} \times \dots \times \{b_{t-1}^1, \dots, b_{t-1}^{n(t-1)}\}$.

⋮

T. Sample $b_T^1, \dots, b_T^{n(T)}$, i.i.d. observations from F_T , independent of those formed in stages $2, \dots, T-1$. Let these denote the same set of descendent nodes for each of the $n_{T-1} = \prod_{t=2}^{T-1} n(t)$ leaf nodes given by $\{b_1\} \times \{b_2^1, \dots, b_2^{n(2)}\} \times \dots \times \{b_{T-1}^1, \dots, b_{T-1}^{n(T-1)}\}$. \square

In the scenario tree constructed by this SSTII algorithm, we have the same set of scenarios at each branch of the tree. Figure 3.2 presents an example of a finite three stage scenario tree built by this algorithm with three possible scenarios per stage, i.e., $n(t) = 3$ for all t .

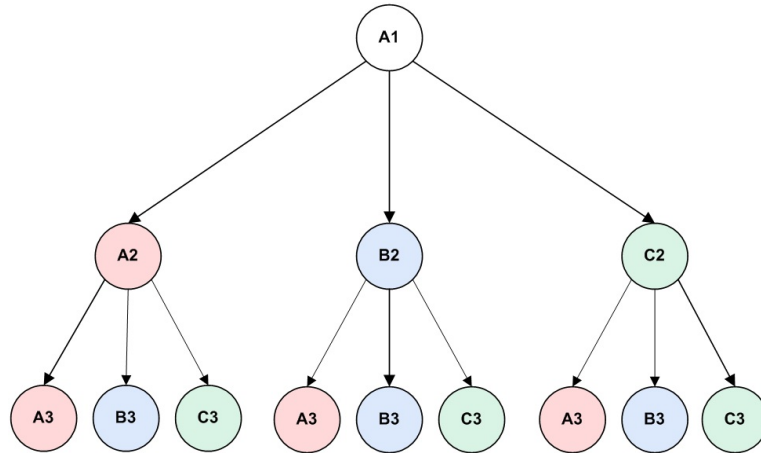


Figure 3.2: Three-stage scenario tree under interstage independent model

Note that in Figure 3.2 the first node represents the first stage where the energy inflows are assumed to be deterministically known. In the second stage we have three scenarios. In the third stage we have for the left branch of the tree from $A2$ the same set of three descendant scenarios $\{A3, B3, C3\}$

that we have for the other branches (*B2* and *C2*). This characterizes the interstage independent setting, i.e., the energy inflows in the third stage are not influenced by what happened before. Here, the original infinite scenario tree and the finite sampled scenario formed by the SSTII algorithm both exhibit interstage independence.

In stage t of the scenario tree created by the SSTII algorithm there are a total of $n_t = \prod_{t=2}^t n(t)$ nodes. These are leaf nodes only temporarily during construction of the tree. In the final scenario tree output by SSTII there are leaf nodes only in stage T . In a more general tree, it will be convenient to denote the n_t realizations on stage t in two distinct ways, b_t^i , $i = 1, \dots, n_t$, and $b_t^{i',j}$, $j = 1, \dots, n(t)$, $i' = 1, \dots, n_{t-1}$. When we use the sampled scenario tree to form an instance of model (3.10), the former representation indexes over $\omega_t \in \Omega_t$ and the latter representation indexes over $\omega_t \in \Delta(\omega_{t-1})$ for each $\omega_{t-1} \in \Omega_{t-1}$. Given i in the former representation, $i' = \lfloor (i-1)/n(t) \rfloor + 1$ and $j = (i-1) \bmod n(t) + 1$. And, given (i', j) in the latter representation, $i = (i'-1)n(t) + j$. We use this notation in describing the procedure to construct a sampled scenario tree under the autoregressive-style interstage dependency model (3.12) in the next section.

3.5.2 Sampled Scenario Tree under Interstage Dependence Model (SSTIDM)

Now assume that the vectors b_t , $t = 2, \dots, T$, are interstage dependent according to model (3.12). In this case we form a sample scenario tree in the

following fashion.

Input: Multivariate distributions $F_t(\cdot)$ governing η_t , $t = 2, \dots, T$, assumed to be interstage independent. A procedure for drawing i.i.d. observations from each F_t . Dependency model (3.12) with known R_j^t matrices. Branch size $n(t)$ for each stage, $t = 2, \dots, T$, e.g., $n(t) = 20$, $\forall t$.

Output: A finite sampled scenario tree with the property that its inflow vectors satisfy the dependency model (3.12), where the independent increments η_t have the empirical distribution the algorithm has formed by sampling.

1. Let b_1 denote the known first stage realization.
2. Sample $\eta_2^1, \dots, \eta_2^{n(2)}$, i.i.d. observations from F_2 . Use equation (3.12a) with $t = 2$ to form $b_2^1, \dots, b_2^{n(2)}$, i.e., $b_2^i = R_1^2 b_1 + \eta_2^i$, $i = 1, \dots, n(2)$. Let these denote the descendent nodes of b_1 .
3. Sample $\eta_3^1, \dots, \eta_3^{n(3)}$, i.i.d. observations from F_3 , independent of those formed in stage 2. For each of the $i = 1, \dots, n_2$ stage 2 leaf nodes use equation (3.12a) with $t = 3$ to form the descendant nodes $b_3^{i,1}, \dots, b_3^{i,n(3)}$ using the *same* set of increments $\{\eta_3^1, \dots, \eta_3^{n(3)}\}$.
- \vdots
- t. Sample $\eta_t^1, \dots, \eta_t^{n(t)}$, i.i.d. observations from F_t , independent of those formed in stages $2, \dots, t-1$. For each of the $i = 1, \dots, n_{t-1}$ stage $t-1$ leaf nodes use equation (3.12a) with t to form the descendant nodes $b_t^{i,1}, \dots, b_t^{i,n(t)}$ using the same set of increments $\{\eta_t^1, \dots, \eta_t^{n(t)}\}$.

⋮

T. Sample $\eta_t^1, \dots, \eta_t^{n(T)}$, i.i.d. observations from F_T , independent of those formed in stages $2, \dots, T - 1$. For each of the $i = 1, \dots, n_{T-1}$ stage $T - 1$ leaf nodes use equation (3.12a) with $t = T$ to form the descendant nodes $b_T^{i,1}, \dots, b_T^{i,n(T)}$ using the same set of increments $\{\eta_T^1, \dots, \eta_T^{n(T)}\}$. \square

Figure 3.3 presents a simple representation of a finite scenario tree with three stages under interstage dependence.

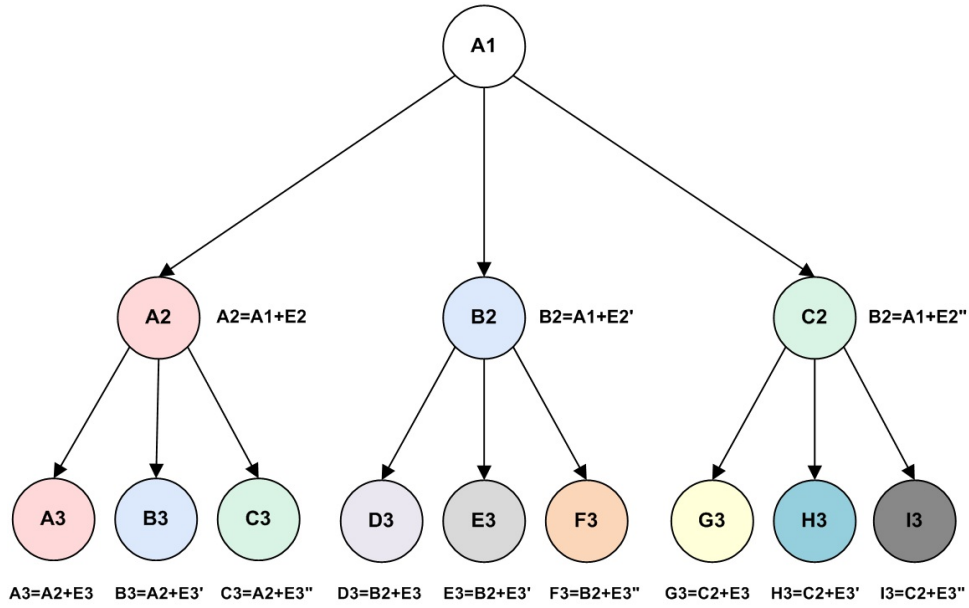


Figure 3.3: Three-stage scenario tree under interstage dependent model

In Figure 3.3 the first node represents the first stage where the energy inflows are assumed to be deterministically known. In the second stage we have three different possible scenarios. In the third stage we have for each branch

of the tree different sets of possible scenarios, this characterizes the interstage dependent setting. The energy inflows on the third stage are influenced by what happened in the second stage.

3.5.3 Sampled Scenario Tree Observations

As indicated in the SSTII and SSTIDM procedures, we emphasize using the *same* set of $n(t)$ observations at stage t to form the descendent nodes of all stage n_{t-1} scenarios. This ensures the resulting sampled tree is interstage independent under SSTII, and satisfies the dependency model (3.12) under SSTIDM. Note that if we instead used a separate, independent set of i.i.d. observations $b_t^1, \dots, b_t^{n(t)}$ for each of the stage n_{t-1} scenarios in SSTII then the resulting sampled tree would *not* be interstage independent. An analogous fact holds for the similarly modified SSTIDM procedure. We wish to maintain interstage independence, or more generally the dependency model (3.12), in our sampled scenario trees so that we can apply an SBDA to solve the stochastic program defined on those trees. The SBDA idea does not apply, for example, to the scenario tree in which we instead use a separate, independent set of i.i.d. observations $b_t^1, \dots, b_t^{n(t)}$ for each of the stage n_{t-1} scenarios in SSTII.

As indicated in model (3.10), let z^* denote the optimal value to model (3.10) defined on the original, infinite scenario tree. Let \hat{z}^* denote the optimal value to model (3.10) instead defined on the simple scenario tree formed using the procedure SSTII or SSTIDM. Of course, \hat{z}^* is a random variable. And, for problems with more than a few stages we cannot compute a realization of \hat{z}^*

because of the exponentially-sized scenario tree. That said, we do know that $\mathbb{E}\hat{z}^* \leq z^*$ (Chiralaksanakul and Morton (2004) and Mak et al. (1999)), and this fact is useful in establishing the quality of a feasible policy.

We say above that the b_t in SSTII, and the η_t in SSTIDM, are drawn i.i.d. at each stage. However, we can use other sampling schemes. For example, we could use Latin hypercube sampling (LHS) or randomized quasi Monte Carlo (RQMC) sampling schemes to generate these observations (Homem-de-Mello et al., 2011). Our only requirement of such an alternative sampling scheme is that it produce an unbiased estimator if we remove the optimization operator, i.e., that η_{t+1} be drawn from F_{t+1} so that the samples satisfy the following unbiasedness condition

$$\mathbb{E}_{\eta_{t+1}} h_{t+1}(x_t, b_{t+1}(\eta_{t+1})) = \mathbb{E} \left[\frac{1}{n(t+1)} \sum_{j=1}^{n(t+1)} h_{t+1}(x_t, b_{t+1}(\eta_{t+1}^j)) \right], \quad (3.14)$$

for all x_t and $t = 1, \dots, T-1$. Here, we have used the representation of the conditional expected future cost function given by (3.13) and have suppressed the dependence of b_{t+1} on b_1, \dots, b_t only for notional simplicity.

3.6 Sampling-based Decomposition Algorithm

The first SBDA-type implementation to appear in the literature was the SDDP algorithm developed by Pereira and Pinto (1991) in order to approximately solve problems of the type described in Section 3.3. While we formalize the SDDP algorithm below, it is easy to visualize as shown in Figure 3.4. To simplify this description, for the moment we regard the finite sampled tree

obtained from the SSTII or SSTIDM procedure of Section 3.5 as defining the problem we wish to solve. So, when we reference the expectation operator $\mathbb{E}_{b_{t+1}|b_1,\dots,b_t}$, this will be with respect to the sampled scenario tree, whose sample space we denote $\hat{\Omega}_T$ in place of the original sample space Ω_T .

During a typical iteration of the SDDP algorithm, cuts have been accumulated at each stage. These represent a piecewise linear outer approximation of the expected future cost function, i.e., $\mathbb{E}_{b_{t+1}|b_1,\dots,b_t} h_{t+1}(x_t, b_{t+1})$, at each stage. On a forward pass we sample a number of linear paths through the tree as depicted in Figure 3.4a. As we solve a sequence of problems at each time stage along a single forward path, the cuts that have been accumulated so far are used to form decisions at each stage. These collections of cuts yield a *policy* that indicates the decisions we will make on any such path through the tree. That policy does not anticipate the future. In fact, the decisions can be made at a node on a sample path at stage t , even before we sample the random energy inflow at stage $t + 1$. In this way, the sample mean of the costs incurred along all the forward sampled paths through the tree form an estimator of the expected cost we incur by following the policy specified by the current set of cuts. Figure 3.4a shows three explicit forward paths in a four-stage tree, while about 200 forward paths in a 120-stage tree are used in de Matos et al. (2008). Note that those (say) 200 sampled forward paths should be selected independently in each iteration of the SDDP algorithm.

In the backward pass of the algorithm, we add cuts to the collection defining the current approximation of the expected future cost function at

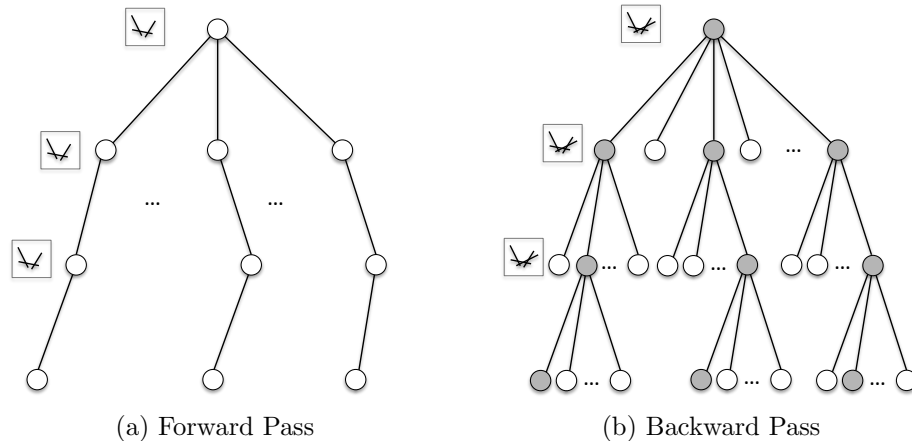


Figure 3.4: The figure depicts forward and backward passes of the SDDP algorithm

each stage. We do this by solving the descendent nodes of each node in the linear paths from the forward pass, except in the final stage, T . In Figure 3.4b the gray nodes correspond to the nodes that were selected in this iteration's forward pass. The white nodes are the additional nodes we solve as part of the backward pass in order to construct optimality cuts. Figure 3.4 shows a single set of cuts corresponding to all the nodes on each stage. This is a correct depiction when the b_t , $t = 2, \dots, T$, are interstage independent. When they are instead dependent according to the autoregressive-style model (3.12) then an intercept-correction formula is applied to each node, which requires keeping track of, and then using, the expected value of the dual variables associated with the cuts and the lag term $(\sum_{j=1}^{t-1} R_j^t b_j)$ as detailed in Infanger and Morton (1996) and later in this section.

Let \vec{G}_t and \vec{g}_t denote the cut-gradient matrix and cut-intercept vector

for stage t that have been generated so far when running the SDDP algorithm. Each backward pass of SDDP along a sample path (there are three in Figure 3.4b) augments \vec{G}_t and \vec{g}_t with one additional row, in a manner we make precise below. We define a stage t optimization problem that we view as a master program with respect to its stage $t + 1$ descendants and as a subproblem with respect to its stage $t - 1$ ancestor:

$$z_t = \min_{x_t, \theta_t} \quad c_t x_t + \theta_t \quad (3.15a)$$

$$\text{s.t.} \quad A_t x_t = B_t x_{t-1} + b_t \quad : \quad \pi_t \quad (3.15b)$$

$$-\vec{G}_t x_t + e \theta_t \geq \vec{g}_t \quad : \quad \alpha_t \quad (3.15c)$$

$$x_t \geq 0. \quad (3.15d)$$

Here, θ_t in the objective function (3.15a), coupled with cut constraints (3.15c), forms an outer linearization of the recourse function $\mathbb{E}_{b_{t+1}|b_1, \dots, b_t} h_{t+1}(x_t, b_{t+1})$ from model (3.11). The structural and nonnegativity constraints (3.15b) and (3.15d) simply repeat the same constraints from model (3.11). The column vector e in constraint (3.15c) is the appropriately-dimensioned vector of all 1s, i.e., e 's dimension is the number of cuts accumulated in that stage. The master problem (3.15) holds for $t = 1, \dots, T$, except that for $t = T$ the cut constraints (3.15c) are absent. The π_t and α_t represent dual vectors associated with constraints (3.15b) and (3.15c), respectively. It is model (3.15) that is solved at each node in the forward sample paths depicted in Figure 3.4a. Recalling notation from Section 3.3, when we let $b_t = b_t^{\omega_t}$ and $x_{t-1} = x_{t-1}^{\omega_t}$, we refer to model (3.15) as $\text{sub}(\omega_t)$. When we specialize model (3.15) in this way,

we similarly append ω_t superscripts to the optimal value, $z_t^{\omega_t}$, as well as the the primal and dual solutions, $(x_t^{\omega_t}, \theta_t^{\omega_t})$ and $(\pi_t^{\omega_t}, \alpha^{\omega_t})$. The SDDP algorithm is formalized below.

Stochastic Dual Dynamic Programming (SDDP) Algorithm:

Input: Instance of model (3.10) with finite scenario tree, interstage independence, and a modest number of immediate descendants at each stage. M_t used to form lower bound on future cost function at each stage.

Output: Set of cuts, \vec{G}_t^{sddp} and \vec{g}_t^{sddp} , $t = 1, \dots, T$, that yield policy for model (3.10), first stage solution x_1 , and lower bound estimator \underline{z} .

0. let $k = 0$;
append lower bounding cuts $\theta_t \geq -M_t$, $t = 1, \dots, T - 1$;
1. solve the stage 1 master program, i.e., (3.15) with $t = 1$, and obtain (x_1^k, θ_1^k) ;
let $\underline{z}^k = c_1 x_1^k + \theta_1^k$;
2. sample i.i.d. paths from $\hat{\Omega}_t$ and index them by \mathcal{S}^k ;
do $\omega \in \mathcal{S}^k$
do $t = 2$ to T
form RHS of sub(ω_t): $B_t[x_{t-1}^{\omega_t}]^k + b_t^{\omega_t}$;
solve and obtain $[x_t^{\omega_t}]^k$;
enddo
enddo
let $\bar{z}^k = c_1 x_1^k + \frac{1}{|\mathcal{S}^k|} \sum_{\omega \in \mathcal{S}^k} \sum_{t=2}^T c_t [x_t^{\omega_t}]^k$;
3. if stopping criterion, given \bar{z}^k and \underline{z}^k , is satisfied then stop and output: (i) set of cuts, \vec{G}_t^{sddp} and \vec{g}_t^{sddp} , $t = 1, \dots, T$, (ii) first stage solution $x_1 = x_1^k$, and (iii) lower bound $\underline{z} = \underline{z}^k$;
4. do $t = T - 1$ downto 1

```

do  $\omega \in \mathcal{S}^k$ 
  do  $\omega_{t+1} \in \Delta(\omega_t)$ 
    form RHS of sub( $\omega_{t+1}$ ):  $B_t[x_t^{\omega_t}]^k + b_{t+1}^{\omega_{t+1}}$ ;
    solve to obtain dual vectors  $\pi_{t+1}^{\omega_{t+1}}$ ,  $\alpha_{t+1}^{\omega_{t+1}}$  and optimal
    value  $z_{t+1}^{\omega_{t+1}}$ ;
  enddo
  Form cut gradient and intercept:
   $G_t = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \pi_{t+1}^{\omega_{t+1}} B_{t+1}$  and
   $g_t = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} z_{t+1}^{\omega_{t+1}} - G_t[x_t^{\omega_t}]^k$ ;
  augment the set of stage  $t$  cuts with  $\theta_t - G_t x_t \geq g_t$ ;
enddo
enddo
5. let  $k = k + 1$ ; goto step 1;  $\square$ 

```

When costs are positive, we can take $M_t = 0$, and enforce nonnegativity for θ_t . Steps 2 and 4 of the SDDP algorithm carryout the forward and backward passes depicted in Figure 3.4a and 3.4b, respectively. In step 2 at the k -th iteration, we select a set of random sample paths from the root node to the stage T leaf nodes denoted \mathcal{S}^k , and an element $\omega \in \mathcal{S}^k$ has the form $\omega = (\omega_2, \dots, \omega_T)$. We could, e.g., have $|\mathcal{S}^k| = 200, \forall k$. In step 2, we form the RHS of sub(ω_t) as $B_t[x_{t-1}^{a(\omega_t)}]^k + b_t^{\omega_t}$. When we do this, we are within a loop with $\omega = (\omega_2, \dots, \omega_T)$ fixed and so we could arguably simplify the notation and replace the $a(\omega_t)$ superscript on x_{t-1} with ω_{t-1} . In the analogous part of step 4, there is no chance for ambiguity, i.e., we are assured that $[x_t^{\omega_t}] = [x_t^{a(\omega_{t+1})}]^k$ since we explicitly loop over $\omega_{t+1} \in \Delta(\omega_t)$. In step 4, and in the remainder of this chapter, we use $p_{t+1}^{\omega_{t+1}|\omega_t}$ when we calculate the cut gradient and cut intercepts. However, given how the sampled scenario tree is constructed we

simply have $p_{t+1}^{\omega_{t+1}|\omega_t} = 1/n(t)$

In step 4, when we form G_t and g_t and augment the set of stage t cuts, this amounts to adding one row to the matrix of cut gradients, \vec{G}_t , and the vector of cut intercepts, \vec{g}_t , that define the stage t subproblem (3.15). Step 4 calculates the cut intercept as

$$g_t = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} z_{t+1}^{\omega_{t+1}} - G_t[x_t^{\omega_t}]^k. \quad (3.16)$$

An equivalent way to calculate the cut intercept is

$$g_t = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \pi_{t+1}^{\omega_{t+1}} b_{t+1}^{\omega_{t+1}} + \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \alpha_{t+1}^{\omega_{t+1}} \vec{g}_{t+1}, \quad (3.17)$$

although when $t = T - 1$ the second term in (3.17) that involves the cut intercept of the descendant nodes, is absent. The equivalence follows by taking the dual of the subproblem (3.15) for stage $t+1$. This can be useful for verifying an implementation. When using the form given in (3.17), we must include in the intercept all the terms associated with simple upper bounds on the decision variables. Note that in step 4 of the SDDP procedure, we obtain $\alpha_{t+1}^{\omega_{t+1}}$. That vector is not used in the above statement of the algorithm, but it would be used if we calculated the cut intercept via (3.17) and it will be required in the extension to handle the autoregressive-style dependency model we discuss next.

3.7 Cut-Sharing under Interstage Dependency Model

When b_t , $t = 2, \dots, T$, are not interstage independent but rather follow the autoregressive-style dependency model (3.12), the SDDP algorithm requires

modification (Infanger and Morton, 1996). First, we have not indexed G_t and g_t above with an ω_t because the same set of cuts is valid for each scenario on stage t , i.e., we store a single set of cuts on each stage. This remains true for the cut gradients under dependency model (3.12) but the cut intercepts differ by scenario and require an ω_t index. It can be shown (see Infanger and Morton, 1996) that we can express $g_t^{\omega_t}$ in the following additive form

$$g_t^{\omega_t} = g_t^{ind} + g_t^{dep}(\omega_t). \quad (3.18)$$

When we first form a cut in step 4 of SDDP, we can compute $g_t^{\omega_t}$ exactly as given in the SDDP procedure, or equivalently via equation (3.17). Then, using equation (3.19) below we can compute $g_t^{dep}(\omega_t)$ and then via (3.18) form $g_t^{ind} = g_t^{\omega_t} - g_t^{dep}(\omega_t)$. The dependent part of the cut intercept for stage t , $t = 2, \dots, T - 1$, is given by

$$g_t^{dep}(\omega_t) = \bar{\pi}_{t+1} \sum_{j=1}^t R_j^{t+1} b_j^{\omega_j} + \bar{\alpha}_{t+1} \sum_{i=t+1}^T D_{t+1}^i \sum_{j=1}^t R_j^i b_j^{\omega_j}, \quad (3.19)$$

where D_t^i , $i \geq t$ is recursively defined as

$$\left. \begin{aligned} D_t^t &= \mathcal{P}_{t+1} R_t^{t+1} + \mathcal{A}_{t+1} \sum_{i=t+1}^T D_{t+1}^i R_t^i \\ D_t^{t+1} &= \mathcal{P}_{t+1} + \mathcal{A}_{t+1} D_{t+1}^{t+1} \\ D_t^i &= \mathcal{A}_{t+1} D_{t+1}^i, \quad i \geq t+2 \end{aligned} \right\} \quad (3.20)$$

with $D_T^T = 0$ and $\mathcal{A}_T = 0$. Here, when we compute a cut in step 4 of the

SDDP algorithm, we form:

$$\bar{\pi}_{t+1} \equiv \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \pi_{t+1}^{\omega_{t+1}} \quad (3.21a)$$

$$\bar{\alpha}_{t+1} \equiv \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \alpha_{t+1}^{\omega_{t+1}}. \quad (3.21b)$$

Recall that the stage t subproblem (3.15) has m_t rows of structural constraints in (3.15b) and we denote by ℓ_t the number of cuts in constraints (3.15c). We define \mathcal{P}_t to be the $\ell_{t-1} \times m_t$ matrix whose rows contain the expected value of the structural constraint dual variables, $\bar{\pi}_t$. Similarly, \mathcal{A}_t is defined as the $\ell_{t-1} \times \ell_t$ matrix whose rows contain the expected value of the cut constraint dual variables, $\bar{\alpha}_t$. For each cut, the SDDP algorithm must then store: (i) the cut gradient, G_t , (ii) the scenario independent cut intercept term, g_t^{ind} , and (iii) the expected dual vectors $\bar{\pi}_{t+1}$ and $\bar{\alpha}_{t+1}$. Then, when we form a subproblem to solve, either in step 2 or in step 4 of the SDDP, we must form the correct set of cut intercepts via (3.18), (3.19) and (3.20).

There is another way to handle interstage dependency that is useful for verifying an implementation. Specifically, we can use the model with interstage independence, and the SDDP algorithm for that model, by expanding the definition (and dimension) of x_t to capture the history. For example, at stage 3, we have $b_3 = R_1^3 b_1 + R_2^3 b_2 + \eta_3$ with the RHS of subproblem (3.15) being $B_3 x_2 + b_3 = B_3 x_2 + R_1^3 b_1 + R_2^3 b_2 + \eta_3$. By redefining B_3 and x_2 to include additional components for R_1^3 , R_2^3 and b_1 , b_2 , respectively, we can capture the interstage dependency model. More specifically, in stage 1 we can introduce a new decision variable $y_1 = b_1$, and in stage 2 we can introduce $y_2 = \begin{pmatrix} y_1 \\ b_2 \end{pmatrix}$.

Then in stage 3 we can use the y_2 carried forward to form $R_1^3 b_1 + R_2^3 b_2$ as part of the “ Bx ” term, and then add the independent component η_3 . We suspect that there is a computational advantage to *not* expanding the dimension of the stage variable in this way and instead using the ideas explained above in (3.18), (3.19) and (3.20), but this hypothesis has not been computationally tested, to our knowledge.

3.8 A Modified SLP- T Formulation

The cut-sharing procedure described in Section 3.7 is designed to handle cases where b_t , $t = 2, \dots, T$, satisfy dependency model (3.12) and the underlying SLP- T has the form of model (3.10). This is valid in the context of a hydro-thermal scheduling model with hydro-thermal scheduling with individual generators, because the parameter b_t represents the water inflow at a certain reservoir at time period t . In the case of the model (3.10) under the ARR, b_t instead represents the energy inflow at a specific equivalent reservoir. So, in the literature, applications of the hydro-thermal scheduling with ARR are coupled with time series forecasts of energy inflows.

That said, there is a compelling argument to be made for forecasting water inflows at the level of individual reservoirs rather than energy inflow into aggregate reservoirs. Forecasting water inflows at individual reservoirs allows the forecasting model to exploit local predictors like precipitation and soil-type in local run-off models in hydrology. Forecasting water inflows in individual reservoirs, rather than energy inflow in an aggregate reservoir, better

lends itself to validation because individual reservoir inflows are measurable. Moreover, when the configuration of the hydro power system evolves over time (as described in Section 2.3.1) this complicates an energy-inflow forecast because a different forecasting model must be constructed for each configuration. However, a forecasting model with individual reservoir resolution of water inflows is unaffected by the hydro system configuration. Forecasting energy inflows, rather than measurable water inflows, arguably unnecessarily ties a forecasting model of a *natural process* to the *decision process* associated with the hydro-thermal system.

Thus, in order to best couple the hydro-thermal scheduling model with ARR, with a forecasting model, we prefer as a model primitive to have the time series model forecast water inflows. This, in turn, requires that we extend existing methods to compute valid cuts for the decomposition method under the resulting form of interstage dependence.

A modified SLP-T, that considers water inflows at individual reservoirs rather energy inflows at aggregate reservoirs, with recourse may be formulated as follows:

$$\begin{aligned}
z^* &= \min_{x_1} c_1 x_1 + \mathbb{E}_{b_2|b_1} h_2(x_1, b_2) \\
\text{s.t.} \quad & A_1 x_1 = B_1 x_0 + \rho_1 b_1 + k_1 \\
& x_1 \geq 0,
\end{aligned} \tag{3.22}$$

where for $t = 2, \dots, T$,

$$\begin{aligned}
h_t(x_{t-1}, b_t) &= \min_{x_t} c_t x_t + \mathbb{E}_{b_{t+1}|b_1, \dots, b_t} h_{t+1}(x_t, b_{t+1}) \\
\text{s.t.} \quad & A_t x_t = B_t x_{t-1} + \rho_t b_t + k_t \\
& x_t \geq 0.
\end{aligned} \tag{3.23}$$

The matrices A_t have m_t rows and d_t columns, $t = 1, \dots, T$. The column-vectors b_t have dimension q_t , in our hydro-thermal model q_t represents the number of individual hydro generators. The matrices ρ_t have m_t rows and q_t columns, in our hydro-thermal model ρ_t represents vectors that transform water inflows at the individual reservoirs into energy inflows (controllable and uncontrollable) at the aggregate reservoirs. The other vectors and matrices are dimensioned to conform.

The term $B_t x_{t-1}$ includes the energy storage that is carried forward from stage $t - 1$ and is available at stage t . This term further captures the parameters of the linear functions (e.g., aEM_i^t) that depend on x_{t-1} . Note that, these model also differs from models (3.10) and (3.11) by the column-vector k_t . The column-vector k_t is used to separate the constant terms of the model, i.e., we consider in our hydro-scheduling model the demand d_t , the intercept terms of the energy functions (e.g., bEM_t) and the other constant terms of each stage into the column-vector k_t . We need this distinction in order to construct the cut-sharing procedure that considers water inflows.

Suppose we are at stage t under scenario ω_t . Assume that we have a simple lag-one linear dependency model for random water inflows given by:

$$b_t = R_{t-1} b_{t-1} + \eta_t, \quad t = 2, \dots, T, \quad (3.24a)$$

$$\eta_t, \quad t = 2, \dots, T, \text{ are independent.} \quad (3.24b)$$

We assume in our model that:

$$\text{vec}(c_t, B_t, A_t, k_t), \quad t = 2, \dots, T \text{ are independent.} \quad (3.25)$$

3.8.1 Basic Cut Computation

An analogue of model (3.15) for our modified SLP- T can be defined as:

$$z_t = \min_{x_t, \theta_t} \quad c_t x_t + \theta_t \quad (3.26a)$$

$$\text{s.t.} \quad A_t x_t = B_t x_{t-1} + \rho_t (R_{t-1} b_{t-1} + \eta_t) + k_t : \pi_t \quad (3.26b)$$

$$-\vec{G}_t x_t + e \theta_t \geq \vec{g}_t \quad : \alpha_t \quad (3.26c)$$

$$x_t \geq 0. \quad (3.26d)$$

The dual of problem (3.26), is:

$$z_t = \max_{\pi_t, \alpha_t} \quad \pi_t \left[B_t x_{t-1} + \rho_t (R_{t-1} b_{t-1} + \eta_t) + k_t \right] + \alpha_t \vec{g}_t \quad (3.27a)$$

$$\text{s.t.} \quad \pi_t A_t - \alpha_t \vec{G}_t \leq c_t : x_t \quad (3.27b)$$

$$e^T \alpha_t = 1 \quad : \theta_t \quad (3.27c)$$

$$\alpha_t \geq 0 \quad (3.27d)$$

For model (3.26) the cut gradient is computed as stated on the SDDP algorithm. The cut intercept is computed as in step 4 of the SDDP algorithm or using the alternative formula given by:

$$g_t^{\omega_t} = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1} | \omega_t} \pi_{t+1}^{\omega_{t+1}} (\rho_{t+1} b_{t+1}^{\omega_{t+1}} + k_{t+1}) + \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1} | \omega_t} \alpha_{t+1}^{\omega_{t+1}} \vec{g}_{t+1}^{\omega_{t+1}}. \quad (3.28)$$

We have an interstage dependency model in b_{t+1} and \vec{g}_{t+1} that appears when we compute the cut intercept for this problem. Because of that, we need to formulate a procedure, similar to the one presented in Section 3.7, to enable cut-sharing so that we can facilitate an SBDA for our modified model (3.22).

3.8.2 Expanding the State

As mentioned earlier one way to handle dependency models is to use an inter-stage independent model with an increase in the dimension of the x_t vector in order to capture the history of the realizations of the random parameters. We now show how this works for model (3.22).

As in Section 3.7 suppose we add an auxiliary variable y_t to model (3.22) in order to capture the history. We set these variables equal to the water inflows from that period, i.e., $y_t = b_t = R_{t-1}b_{t-1} + \eta_t$. Consider a reformulation of model (3.22) in which we introduce an additional set of decision variables and constraints in each stage:

$$\begin{aligned}
 z^* = \min_{x_1, y_1} & \quad c_1 x_1 + \mathbb{E}_{b_2|b_1} h_2(x_1, y_1, b_2) \\
 \text{s.t.} & \quad A_1 x_1 = B_1 x_0 + \rho_1 y_1 + k_1 \\
 & \quad y_1 = b_1 \\
 & \quad x_1 \geq 0,
 \end{aligned} \tag{3.29}$$

where for $t = 2, \dots, T$,

$$\begin{aligned}
 h_t(x_{t-1}, y_{t-1}, b_t) = \min_{x_t, y_t} & \quad c_t x_t + \mathbb{E}_{b_{t+1}|b_1, \dots, b_t} h_{t+1}(x_t, y_t, b_{t+1}) \\
 \text{s.t.} & \quad A_t x_t = B_t x_{t-1} + \rho_t y_t + k_t \\
 & \quad y_t = R_{t-1} y_{t-1} + \eta_t \\
 & \quad x_t \geq 0.
 \end{aligned} \tag{3.30}$$

Applying the decomposition algorithm to model (3.29) we arrive at the

full master program with the following form:

$$z_t = \min_{x_t, y_t, \theta_t} \quad c_t x_t + \theta_t \quad (3.31a)$$

$$\text{s.t.} \quad A_t x_t - \rho_t y_t = B_t x_{t-1} + k_t \quad : \pi_t^B \quad (3.31b)$$

$$-\vec{G}_t^x x_t - \vec{G}_t^y y_t + e \theta_t \geq \vec{g}_t \quad : \alpha_t \quad (3.31c)$$

$$y_t = R_{t-1} y_{t-1} + \eta_t \quad : \pi_t^S \quad (3.31d)$$

$$x_t \geq 0, \quad (3.31e)$$

where \vec{G}_t^x and \vec{G}_t^y are the cut gradient matrix related to the decision variables x_t and y_t , respectively. Now we have π_t^B and π_t^S as the dual variables associated with the set of constraints (3.31b) and (3.31d), respectively.

The dual of (3.31) can be written as:

$$z_t = \max_{\pi_t^B, \pi_t^S, \alpha_t} \quad \pi_t^B (B_t x_{t-1} + k_t) + \alpha_t \vec{g}_t + \pi_t^S (R_{t-1} y_{t-1} + \eta_t) \quad (3.32a)$$

$$\text{s.t.} \quad \pi_t^B A_t - \alpha_t \vec{G}_t^x \leq c_t \quad : x_t \quad (3.32b)$$

$$-\pi_t^B \rho_t - \alpha_t \vec{G}_t^y + e^T \pi_t^S = 0 \quad : y_t \quad (3.32c)$$

$$e^T \alpha_t = 1 \quad : \theta_t \quad (3.32d)$$

$$\alpha_t \geq 0. \quad (3.32e)$$

The purpose of decision variable y_t is to capture the history of the random parameters. The cut gradient is computed using:

$$G_t^x = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1} | \omega_t} \pi_{t+1}^{B, \omega_{t+1}} B_{t+1}, \text{ and} \quad (3.33)$$

$$G_t^y = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1} | \omega_t} \pi_{t+1}^{S, \omega_{t+1}} R_t. \quad (3.34)$$

The cut intercept for the new formulation is computed using:

$$g_t^{\omega_t} = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} z_{t+1}^{\omega_{t+1}} - G_t^x x_t^{\omega_{t+1}} - G_t^y y_t^{\omega_{t+1}}. \quad (3.35)$$

An equivalent way to compute cut intercepts in this case is given by:

$$g_t^{\omega_t} = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} (\pi_{t+1}^{B,\omega_{t+1}} k_{t+1} + \pi_{t+1}^{S,\omega_{t+1}} \eta_{t+1}) + \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \alpha_{t+1}^{\omega_{t+1}} \vec{g}_{t+1}^{\omega_{t+1}}. \quad (3.36)$$

Note that we do not have parameters with interstage dependency models showing up on the computation of the cut intercept. In this case, we can apply the SBDA as we do for problems with interstage independence models. The storage requirements in this case would be: (i) the cut gradients, G_t^x and G_t^y , (ii) the cut intercept term, g_t .

3.8.3 An Equivalence Result

We argue that models (3.31) and (3.32) are equivalent to models (3.26) and (3.27) respectively. In order to show that the models are equivalent we argue via the optimality conditions specified by primal feasibility, dual feasibility and strong duality for the pair of models.

Primal feasibility, dual feasibility and strong duality of the model (3.26) are given by:

$$\begin{aligned} A_t x_t &= B_t x_{t-1} + \rho_t (R_{t-1} b_{t-1} + \eta_t) + k_t \\ -\vec{G}_t x_t + e \theta_t &\geq \vec{g}_t \\ x_t &\geq 0, \end{aligned} \quad (3.37)$$

$$\begin{aligned}
\pi_t A_t - \alpha_t \vec{G}_t &\leq c_t \\
e^T \alpha_t &= 1 \\
\alpha_t &\geq 0,
\end{aligned} \tag{3.38}$$

$$c_t x_t + \theta_t = \pi_t \left[B_t x_{t-1} + \rho_t (R_{t-1} b_{t-1} + \eta_t) + k_t \right] + \alpha_t \vec{g}_t. \tag{3.39}$$

Primal feasibility, dual feasibility and strong duality for model (3.31) are given by:

$$\begin{aligned}
A_t x_t &= B_t x_{t-1} + \rho_t y_t + k_t \\
-\vec{G}_t^x x_t - \vec{G}_t^y y_t + e \theta_t &\geq \vec{g}_t \\
y_t &= R_{t-1} y_{t-1} + \eta_t \\
x_t &\geq 0,
\end{aligned} \tag{3.40}$$

$$\begin{aligned}
\pi_t^B A_t - \alpha_t \vec{G}_t^x &\leq c_t \\
-\pi_t^B \rho_t - \alpha_t \vec{G}_t^y + e^T \pi_t^S &= 0 \\
e^T \alpha_t &= 1 \\
\alpha_t &\geq 0,
\end{aligned} \tag{3.41}$$

$$c_t x_t + \theta_t = \pi_t^B (B_t x_{t-1} + k_t) + \alpha_t \vec{g}_t + \pi_t^S (R_{t-1} y_{t-1} + \eta_t). \tag{3.42}$$

Let $(x_t^*, \theta_t^*, \pi_t^*, \alpha_t^*)$ satisfying (3.37) - (3.39). Thhen, we must argue that we can use this solution to form $(x_t, y_t, \theta_t, \pi_t^B, \pi_t^S, \alpha_t)$ that satisfy (3.40) - (3.42). By using this assumption and strong duality for both models we can state the following:

$$c_t x_t + \theta_t = \pi_t^B (B_t x_{t-1} + k_t) + \alpha_t \vec{g}_t + \pi_t^S (R_{t-1} y_{t-1} + \eta_t)$$

$$c_t x_t^* + \theta_t^* = \pi_t^B (B_t x_{t-1} + k_t) + \alpha_t \vec{g}_t + \pi_t^S (R_{t-1} y_{t-1} + \eta_t)$$

$$\pi_t^* \left[B_t x_{t-1} + \rho_t (R_{t-1} b_{t-1} + \eta_t) + k_t \right] + \alpha_t^* \vec{g}_t = \pi_t^B (B_t x_{t-1} + k_t) + \alpha_t \vec{g}_t + \pi_t^S (R_{t-1} y_{t-1} + \eta_t)$$

By doing some manipulation on the expression above we obtain the dual variables for the modified problem in (3.43)

$$x_t = x_t^* \tag{3.43a}$$

$$y_t = b_t \tag{3.43b}$$

$$\pi_t^B = \pi_t^* \tag{3.43c}$$

$$\pi_t^S = \pi_t^* \rho_t \tag{3.43d}$$

$$\alpha_t = \alpha_t^* \tag{3.43e}$$

Assume we are at the last stage, $t = T$. There are no optimality cuts in model (3.26), constraint set (3.26c), as well as in model (3.31), constraint set (3.31c) associated to this stage. Proceeding by induction, we assume $y_t = b_T$ in model (3.26c), then, it is easy to see that it lends itself into model (3.26).

Now assume we are at stage $t = T - 1$, we need to show that the cuts that are present in model (3.26) are equivalent to the cuts in model (3.31). Consider the cut matrix for (3.26), as defined in the SDDP algorithm, given by:

$$\vec{G}_{T-1} = \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T} \omega_{T-1}^{\omega_T} \pi_T^{\omega_T} B_T. \tag{3.44}$$

We form this matrix using all the dual vertices of stage T .

Now, consider the cut matrix \vec{G}_t^x of model (3.31), that is computed

using (3.33). At stage $t = T - 1$ we have:

$$\vec{G}_{T-1}^x = \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T | \omega_{T-1}} \pi_T^{B, \omega_T} B_T. \quad (3.45)$$

Note that, for equations (3.45) and (3.44) are the same when $\pi_t^B = \pi_t$. So, we can write $\vec{G}_{T-1}^x = \vec{G}_{T-1}$, that holds for any other stage t . For the same stage we can re-write equation (3.31c) in terms of \vec{G}_{T-1} and obtain:

$$-\vec{G}_{T-1} x_{T-1} - \vec{G}_{T-1}^y y_{T-1} + e \theta_{T-1} \geq \vec{g}_{T-1}. \quad (3.46)$$

At this point, we turn to model (3.26). For the cut intercept vector, using equation (3.28), at stage $t = T - 1$ we obtain:

$$g_{T-1}^{\omega_t} = \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T | \omega_{T-1}} \pi_T^{\omega_T} (\rho_T b_T^{\omega_T} + k_T). \quad (3.47)$$

Now, for model (3.31), using equation (3.36), we can write the cut intercept vector as:

$$g_{T-1}^{\omega_{T-1}} = \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T | \omega_{T-1}} (\pi_T^{B, \omega_T} k_T + \pi_T^{S, \omega_T} \eta_T). \quad (3.48)$$

By using the result of (3.48) and (3.34) into (3.46) we can write:

$$\begin{aligned} -\vec{G}_{T-1} x_{T-1} + e \theta_{T-1} &\geq \vec{G}_{T-1}^y y_{T-1} + \vec{g}_{T-1} \\ -\vec{G}_{T-1} x_{T-1} + e \theta_{T-1} &\geq \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T | \omega_{T-1}} \pi_T^{S, \omega_T} R_{T-1} y_{T-1} + \\ &\quad + \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T | \omega_{T-1}} (\pi_T^{B, \omega_T} k_T + \pi_T^{S, \omega_T} \eta_T) \\ -\vec{G}_{T-1} x_{T-1} + e \theta_{T-1} &\geq \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T | \omega_{T-1}} \pi_T^{S, \omega_T} (R_{T-1} y_{T-1} + \eta_T) + \\ &\quad + \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T | \omega_{T-1}} \pi_T^{B, \omega_T} k_T. \end{aligned}$$

Proceeding by induction, we assume that $y_{T-1} = b_{T-1}$ and we can write:

$$\begin{aligned}
-\vec{G}_{T-1}x_{T-1} + e\theta_{T-1} &\geq \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T|\omega_{T-1}} \pi_T^{S,\omega_T} b_T + \\
&\quad + \sum_{\omega_T \in \Delta(\omega_{T-1})} p_T^{\omega_T|\omega_{T-1}} \pi_T^{B,\omega_T} k_T.
\end{aligned} \tag{3.49}$$

It is now possible to see that for $\pi_T^B = \pi_T$ and $\pi_T^S = \pi_T \rho_T$ we have in the RHS of (3.49) \vec{g}_{T-1} as stated in equation (3.47). So, we define \vec{g}_{T-1} of model (3.26) equal to $\vec{g}_{T-1} + \vec{G}_{T-1}^y y_{T-1}$ of model (3.31), this result holds for any other stage t .

With the (primal, dual) pair obtained from (3.43) we can say that we should get the same solutions for both models when we have the established relationship between the cut gradient matrices and the cut intercept vectors. But as mentioned earlier we conjecture that by increasing the dimension of the problem the computational time required to solve the problem may increase, and so we do not expect to have an efficient model using the modified formulation. So instead in the next section we describe the cut-sharing procedure under the aggregate dependency model (3.24) that does not require increasing the dimension of the problem.

3.8.4 Cut-Sharing for SLP-3

We start by illustrating how cut-sharing works with the simple case of a three stage stochastic linear program (SLP-3) with a linear lag-one model given by (3.24). Later we present the general form of the cut formation for a T -stage stochastic linear program (SPL- T).

We have $T = 3$ and we compute cuts for stage $t = 2$, the only non-

trivial case for SLP-3. From model (3.27) for $T = 3$ we have that the second stage cut gradient and intercept are:

$$G_2 = \sum_{\omega_3 \in \Delta(\omega_2)} p_3^{\omega_3|\omega_2} \pi_3^{\omega_3} B_3 \quad (3.50)$$

$$g_2^{\omega_2} = \sum_{\omega_3 \in \Delta(\omega_2)} p_3^{\omega_3|\omega_2} \pi_3^{\omega_3} (\rho_3 b_3^{\omega_3} + k_3). \quad (3.51)$$

From model (3.27) with $T = 3$ we see that $\alpha_3 \vec{G}_3$ in (3.27b), $\alpha_3 \vec{g}_3$ in (3.27a) and constraints (3.27c) - (3.27d) are all absent. This coupled with the fact that c_3 and A_3 are interstage independent (see equation 3.25) means that the collection of dual variables $\{\pi_3^{\omega_3} : \omega_3 \in \Delta(\omega_2)\}$ from the solution of one set of stage 2 descendants is feasible for the descendants of any second stage scenario. Therefore these dual variables generate a valid cut that can be used for any second stage scenario.

It is necessary to show how these cuts can be recomputed for each second stage scenario so that they can be used during future SBDA iterations. We follow the notation presented in Infanger and Morton (1996). Let $\sigma_3 \in \Sigma_3$ index realizations for the third stage. A full indexing of realizations in the third stage would be $\omega_3(\omega_2, \sigma_3)$, but the only parameter that requires ω_3 resolution is $b_3^{\omega_3} = R_2 b_2^{\omega_2} + \eta_3^{\sigma_3}$. All other parameters can be indexed by simply σ_3 , including the conditional probability $p_3^{\omega_3|\omega_2} = p_3^{\sigma_3}$. Also, because B_3 is independent of the second stage random parameter the cut gradient matrix G_2 is written in (3.50) without index ω_2 . But, note that the cut intercept on expression (3.51) presents the random parameter $b_3^{\omega_3}$ that has interstage

dependency model (3.24). By using the dependency model in (3.24) we can separate the cut intercept from (3.51) in two parts as:

$$g_2^{\omega_2} = g_2^{ind} + g_2^{dep}(\omega_2), \quad (3.52)$$

where,

$$g_2^{ind} = \sum_{\sigma_3 \in \Sigma_3} p_3^{\sigma_3} \pi_3^{\sigma_3} (\rho_3 \eta_3^{\sigma_3} + k_3) \quad (3.53)$$

$$g_2^{dep} = \underbrace{\left[\sum_{\sigma_3 \in \Sigma_3} p_3^{\sigma_3} \pi_3^{\sigma_3} \right]}_{\bar{\pi}_3} \rho_3 R_2 b_2^{\omega_2}. \quad (3.54)$$

For SLP-3, the dependent part of the cut intercept (3.54) does not depend on the dual variables associated with the cuts, because there are no cuts in the third stage. In order to compute the dependent part of the cut intercept for a particular second stage scenario it is necessary to know a priori the realization of the random parameter $b_2^{\omega_2}$ in the second stage and the expected dual vector $\bar{\pi}_3$ from stage 3 used to compute a cut. In this case, it is necessary to store for each computed cut: (i) the cut gradient G_2 , (ii) the scenario independent cut intercept g_2^{ind} , and (iii) the expected dual vector $\bar{\pi}_3$. We can compute valid cuts for the second stage with this stored information by calculating the dependent part of the cut intercept using the closed-form scenario-dependent correction (3.54) and then the cut intercept using (3.52).

3.8.5 Cut-Sharing for SLP- T

Model SLP-3 does not show all the difficulties behind cut-sharing in a full SLP- T . In a three-stage model, when we are computing cuts for the second stage we do not have to deal with dual variables associated with cuts constraints of the next stage (because there are no cuts at the last stage).

Consider now a problem with more stages denoted SLP- T . In Theorem 1 we show that the dependent portion of the cut intercept can be computed using:

$$g_t^{dep}(\omega_t) = [\bar{\pi}_{t+1}\rho_{t+1} + \bar{\alpha}_{t+1}D_{t+1}]R_t b_t^{\omega_t}, \quad (3.55)$$

where, for $t = 2, \dots, T$ the matrix D_t is defined recursively as:

$$D_t = [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1}D_{t+1}]R_t, \quad D_T = 0. \quad (3.56)$$

The cut intercept $g_t^{\omega_t}$ is then computed using (3.18), (3.55) and (3.56). Note that if $T = 3$ then (3.55) for $t = 2$ reduces to (3.54). More generally, \mathcal{P}_t is defined to be the $\ell_{t-1} \times m_t$ matrix whose rows contain the expected value of the structural constraint dual variables, $\bar{\pi}_t$. Similarly, \mathcal{A}_t is defined as the $\ell_{t-1} \times \ell_t$ matrix whose rows contain the expected value of the cut constraint dual variables, $\bar{\alpha}_t$.

Theorem 1. *Consider an SLP- T as formulated in (3.22). Assume a linear lag-one model given by (3.24). The cut intercepts for stage t , $t = 2, \dots, T - 1$ are given by (3.18), (3.55) and (3.56).*

Proof. We proceed by induction. First, assume that we are at stage $t = T - 1$. The cut intercept $g_{T-1}^{\omega_{T-1}}$ can be computed using the method described by SLP-3, and in this case we have that (3.55) holds because $D_T = 0$.

The inductive hypothesis is (3.18), (3.55) and (3.56) and we verify the same expressions with t decremented by 1. For notational convenience, the vector analog of (3.55) is:

$$\vec{g}_t^{dep}(\omega_t) = [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1}D_{t+1}]R_t b_t^{\omega_t}. \quad (3.57)$$

From model (3.27) a stage $t - 1$ cut intercept for scenario ω_{t-1} is:

$$g_{t-1}^{\omega_{t-1}} = \underbrace{\sum_{\omega_t \in \Delta(\omega_{t-1})} p_t^{\omega_t | \omega_{t-1}} \pi_t^{\omega_t} (\rho_t b_t + k_t)}_{1^{\text{st}} \text{ term}} + \underbrace{\sum_{\omega_t \in \Delta(\omega_{t-1})} p_t^{\omega_t | \omega_{t-1}} \alpha_t^{\omega_t} \vec{g}_t^{\omega_t}}_{2^{\text{nd}} \text{ term}}. \quad (3.58)$$

By substituting the lag-one model (3.24) into the 1st term of (3.58) we obtain:

$$\bar{\pi}_t \rho_t R_{t-1} b_{t-1}^{\omega_{t-1}} + \mathbb{E}_{\sigma_t} \pi_t^{\sigma_t} (\rho_t \eta_t^{\sigma_t} + k_t). \quad (3.59)$$

By using the inductive hypothesis (3.18) and (3.55) and the lag-one model (3.24) in the 2nd term of (3.58) we obtain:

$$\bar{\alpha}_t \vec{g}_t^{ind} + \bar{\alpha}_t [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1}D_{t+1}]R_t R_{t-1} b_{t-1}^{\omega_{t-1}} + \mathbb{E}_{\sigma_t} \alpha_t^{\sigma_t} [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1}D_{t+1}]R_t \eta_t^{\sigma_t}. \quad (3.60)$$

Now using the definition of matrix D_t (3.56) and summing the two terms we obtain:

$$\bar{\pi}_t \rho_t R_{t-1} b_{t-1}^{\omega_{t-1}} + \mathbb{E}_{\sigma_t} \pi_t^{\sigma_t} (\rho_t \eta_t^{\sigma_t} + k_t) + \bar{\alpha}_t \vec{g}_t^{ind} + \bar{\alpha}_t [D_t] R_{t-1} b_{t-1}^{\omega_{t-1}} + \mathbb{E}_{\sigma_t} \alpha_t^{\sigma_t} [D_t] \eta_t^{\sigma_t}. \quad (3.61)$$

Finally, it is possible to separate expression (3.61) in dependent and independent parts as:

$$g_{t-1}^{dep}(\omega_{t-1}) = [\bar{\pi}_t \rho_t + \bar{\alpha}_t D_t] R_{t-1} b_{t-1}^{\omega_{t-1}} \quad (3.62)$$

$$g_{t-1}^{ind} = \mathbb{E}_{\sigma_t} [\pi_t^{\sigma_t} \rho_t + \alpha_t^{\sigma_t} D_t] \eta_t^{\sigma_t} + \mathbb{E}_{\sigma_t} \pi_t^{\sigma_t} k_t + \bar{\alpha}_t g_t^{ind}, \quad (3.63)$$

which completes the proof. \square

In this case, when running an SBDA it is necessary to store for each computed cut: (i) the cut gradient G_t , (ii) the scenario independent cut intercept g_t^{ind} and (iii) the cumulative expected dual vector $[\bar{\pi}_{t+1} \rho_{t+1} + \bar{\alpha}_{t+1} D_{t+1}] R_t$. At a particular stage t , we can compute valid cuts for $\text{sub}(\omega_t)$ with the stored information by calculating the dependent part of the cut intercept using the closed-form scenario-dependent correction (3.55) and then the cut intercept using (3.18). The first time we compute a cut, the cumulative expected dual vector associated with this cut can be created from the set of cumulative expected dual vectors that from of the descendant scenarios. In order to perform such computation we use (3.55) and (3.56). Relative to the case of interstage independence we need additional storage of $[\bar{\pi}_{t+1} \rho_{t+1} + \bar{\alpha}_{t+1} D_{t+1}] R_t$ for each cut.

3.9 Parallel Implementation

A simplified idea of the SDDP algorithm can be seen in Figure 3.5. We have a small three stage problem with three scenarios per stage (usually we apply the

SDDP to much larger instances). There are a total of 9 forward paths in the scenario tree presented in Figure 3.5. The SDDP algorithm chooses a number of forward paths in the tree and then starts the optimization process. The problem can be decomposed for each forward path because they are completely independent from each other, except for the information of the first node, which is carried on to the second stage nodes. Because of that it is possible to parallelize the algorithm in order to solve large instances faster.

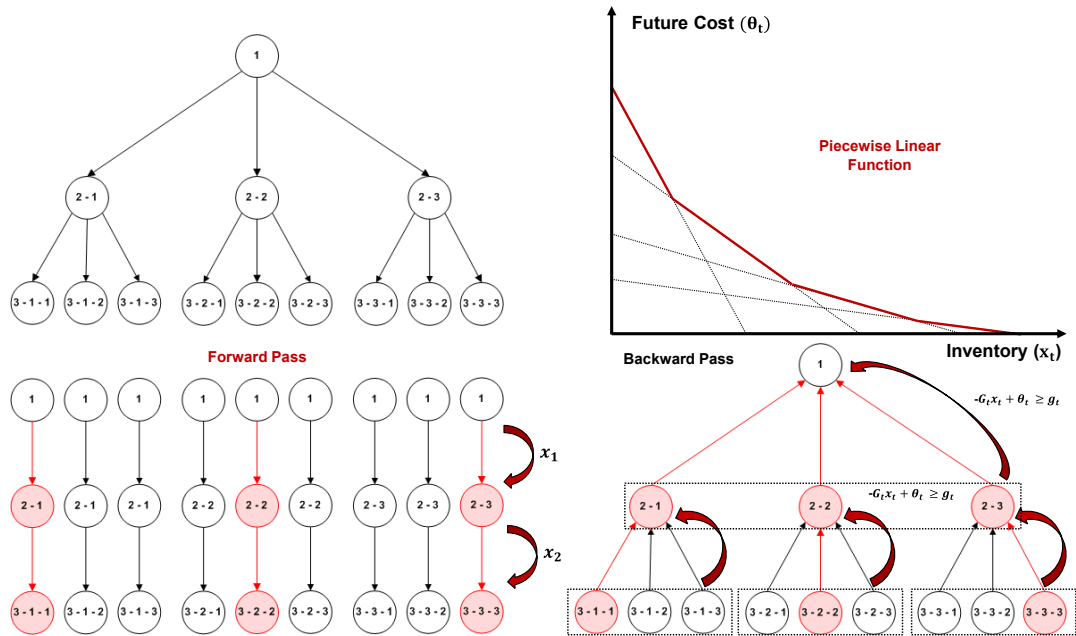


Figure 3.5: General SDDP Scheme

The SDDP algorithm parallelization can be performed in both the forward and backward steps depicted in Figure 3.5. We use Message Passing Interface (MPI) function calls in order to create the parallel implementation

of SDDP. Figure 3.6 presents a general parallelization scheme for the algorithm. The algorithm starts by selecting a number of forward paths on a scenario tree, in which the first node of all the paths is the same and represents the present stage. The problem related to this node is solved by the root processor, or root core, and the information about the connecting variables is broadcast to all other cores. Also, at this point the root core designates the forward paths that each core will have to deal with. The MPI function call used in this case is MPI Scatterv. MPI Scatterv takes a vector of information from one core, in this case the root, and splits that information up, sending a portion of it to each of the other cores. Each core solves all its designated forward paths and computes their costs. If for a specific forward path the algorithm has to compute cuts then the core that was assigned that path also performs the backward step for that path.

The optimal solution of the first stage gives a lower bound on the optimal value, which is computed at the root. After all the forward paths of a specific iteration are solved the algorithm calls MPI Reduce to pass the information about the costs of each path to the root core. After that it is possible to compute the upper bound associated with the sample tree and to decide whether to stop the algorithm. If the algorithm does not stop it is necessary to call MPI Gatherv in order to gather all the computed cuts in the root core. After that there is one more call of MPI Bcast to broadcast all the cuts computed during that iteration to the other cores.

In our simulations we decide to use 128 forward paths, 32 backward

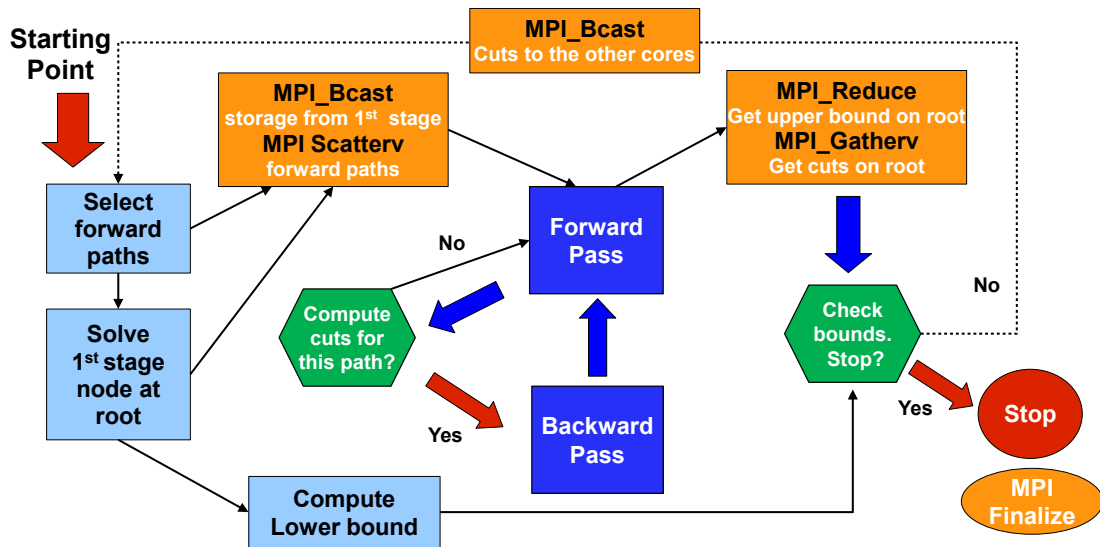


Figure 3.6: General SDDP Parallelization Scheme

paths and 128 cores. This means we assign at each iteration only one forward path for each core and only 32 of these cores have to perform backward steps while the others are idle. Note that the backward step is the one that requires more computational effort. If we decide to reduce the number of cores to 64, at each iteration there are 2 forward paths to be solved by each core and again only 32 cores have to perform the backward step. The computational time will increase a little because now each core has solve two forward paths. The computational time would more than double if instead of one, two backward paths were assigned to be performed by one core. So in order to get faster results we decided to give at most one forward path for each core and one or zero backward paths.

Chapter 4

Solution Quality Evaluation in a Multistage Stochastic Program

4.1 Policy Generation Procedure and Cost Estimator

A solution to a multistage stochastic program is defined by a policy, which specifies what decision to take at each stage, given the history of the stochastic process up to, and including, that stage. Mathematically, a policy for model (3.22) is a sequence of mappings, $x_t(b_1, \dots, b_t)$, $t = 1, \dots, T$, whose domain is the support of the random vector (b_1, \dots, b_t) and whose range is in \mathbb{R}^{d_t} , $t = 1, \dots, T$. To be feasible the sequence of decisions $x_1(b_1)$, $x_2(b_1, b_2), \dots, x_T(b_1, \dots, b_T)$ must be nonnegative, satisfy the sequence of structural constraints in (3.22) and (3.23), and be nonanticipative. Nonanticipativity means that as indicated, $x_t(\cdot)$ can depend on the random inflows through stage t . It can depend on stage $t + 1, \dots, T$ inflows only through the *distribution* of those inflows. In words, a policy is a rule which specifies what decision to take at each stage for each possible realization of the random inflows up to that point in time.

We propose a method for generating a policy that applies to a multistage stochastic linear program with relatively complete recourse whose stochas-

tic parameters exhibit interstage independence or satisfy the dependency model (3.24). This approach uses the fact that SBDA's iteratively form a cut-based policy generation scheme. The forward pass in step 2 of the SDDP algorithm presented in Chapter 3 computes a sample-mean estimator, \bar{z}^k , of the expected cost of the policy at iteration k , evaluated on the sample paths indexed by \mathcal{S}^k .

It is necessary to be clear about the sample space from which we simulate inflow vectors when we assess the performance of our cut-based policy. We let Ω_T denote the sample space of the original stochastic program, i.e., under the stochastic process described in Chapter 3, where Ω_T is uncountably infinite. And, we let $\hat{\Omega}_T$ denote the sample space of the finite scenario tree generated by either the SSTII or SSTIDM procedure of Chapter 3. Recall that z^* denotes the optimal value of the stochastic program defined on the tree associated with Ω_T and \hat{z}^* denotes that associated with $\hat{\Omega}_T$. Suppose we have run SDDP until the termination criteria is satisfied and we have obtained SDDP's output, including \vec{G}_t^{sddp} and \vec{g}_t^{sddp} , $t = 1, \dots, T$, which denote the cuts collected at each stage.

The forward pass in step 2 of SDDP gives information about the “in-sample” performance of the policy at each iteration, i.e., with respect to $\hat{\Omega}_T$'s scenario tree formed by the SSTII or SSTIDM procedure. However, we wish to know the out-of-sample performance of the policy specified by the collection of cuts $(\vec{G}_t^{sddp}, \vec{g}_t^{sddp})$, $t = 1, \dots, T$. In order to do that we sample n_u i.i.d. paths from Ω_T , b_1^i, \dots, b_T^i , $i = 1, \dots, n_u$, and compute $x_t(b_1^i, \dots, b_t^i)$, $t = 1, \dots, T$, by solving the n_u associated sequences of stage t problems (3.26) with $b_t = b_t^i$,

$\vec{G}_t = \vec{G}_t^{sddp}$ and $\vec{g}_t = \vec{g}_t^{sddp}$, and with $x_{t-1} = x_{t-1}(b_1^i, \dots, b_{t-1}^i)$. Then, we can form the estimator

$$U_{n_u} = \frac{1}{n_u} \sum_{i=1}^{n_u} \sum_{t=1}^T c_t x_t(b_1^i, \dots, b_t^i). \quad (4.1)$$

We know that $\mathbb{E}U_{n_u} \geq z^*$. Here, \mathbb{E} denotes the expectation with respect to Ω_T . Of course, $x_1(b_1^i)$ is identical for all $i = 1, \dots, n_u$ samples and it is not necessary to solve (3.26) with $t = 1$ to compute this first stage decision. This is why x_1 is an output of SDDP. That said, we express the estimator (4.1) in this form for notational convenience. Note that here we are talking about model (3.26) and its associated equations, but the following procedures that we describe also can be applied to models (3.15) and (3.31) and their associated equations.

4.1.1 Policy Generation Procedure (PGP)

The following procedure states in algorithmic form the procedure for forming an out-of-sample (i.e., sampling from Ω_T , outside of the finite scenario tree) statistical estimator of the form (4.1) for the policy's cost. For simplicity of describing the procedure, for the moment we assume that $R_{t-1} = 0$ in (3.24), i.e., that the process is inter-stage independent.

- Input:* Model (3.26) with cuts $\vec{G}_t = \vec{G}_t^{sddp}$ and $\vec{g}_t = \vec{g}_t^{sddp}$, $t = 1, \dots, T$, from the output of SDDP. First stage solution x_1 from SDDP.
Sample size n_u .
- Output:* Sample mean estimator, U_{n_u} , and sample variance estimator, S_u^2 , for expected cost of policy.

1. let $x_1^i = x_1, i = 1, \dots, n_u$;
2. sample i.i.d. paths from $\Omega_T, b_1^i, \dots, b_T^i, i = 1, \dots, n_u$;
do $i = 1, \dots, n_u$
do $t = 2$ to T
form RHS of model (3.26b): $B_t x_{t-1}^i + \rho_t b_t^i + k_t$;
solve and obtain x_t^i ;
enddo
let $z^i = \sum_{t=1}^T c_t x_t^i$;
enddo
3. let $U_{n_u} = \frac{1}{n_u} \sum_{i=1}^{n_u} z^i$;
let $S_u^2 = \frac{1}{n_u - 1} \sum_{i=1}^{n_u} (z^i - U_{n_u})^2$; \square

In step 2 of the PGP, we repeatedly solve model (3.26) in a forward pass and obtain $x_t(b_1^i, \dots, b_t^i)$ for each sample path i , which we denote in the algorithm by x_t^i for brevity. Step 3 computes the sample mean estimator of the expected cost of operating under the policy specified by the cuts we obtain as output from SDDP, along with the associated sample variance. We have stated PGP for the case of interstage independence, but the extension to the dependency model (3.24) is not difficult. First, SDDP instead outputs: (i) x_1 , (ii) \vec{G}_t^{sddp} , (iii) the independent components of the cut intercepts, denoted $\vec{g}_t^{sddp, ind}$, (iv) $\bar{\pi}_t$ and $\bar{\alpha}_t$ associated with each cut, i.e., \mathcal{P}_t and \mathcal{A}_t in matrix form, and (v) the cumulative expected dual matrices D_t^i as defined in (3.56). Then, in step 2 of PGP, in addition to forming the RHS of model (3.26) we also form

the cut intercepts using equations (3.18) and (3.55), where b_1^i, \dots, b_t^i replaces $b_1^{\omega_1}, \dots, b_t^{\omega_t}$ in equation (3.55). When we report computational results later in this chapter, we do so with respect to the interstage dependency version of the PGP that we have just sketched.

4.2 Lower Bound Estimation

As when estimating the cost of a policy, we must be clear what the sample space is when we discuss lower bound estimation. As before, $\hat{\Omega}_T$ denotes the sample space associated with the finite sampled scenario tree and Ω_T denotes that of the original stochastic process (3.24). The respective optimal values for the two corresponding stochastic programs are z^* and \hat{z}^* . In Section 4.1 our primary aim was to estimate the cost of a policy with respect to Ω_T , i.e., we obtained a policy with estimated cost U_{n_u} , where $\mathbb{E}U_{n_u} \geq z^*$. Similarly, in this section we seek a lower bound estimator on z^* . That said, \hat{z}^* , or rather its bounds and related estimators, play an important role in achieving this goal.

As is shown in Chiralaksanakul and Morton (2004), we have that $\mathbb{E}\hat{z}^* \leq z^*$. The intuition behind this bound is clear when the branch size $n(t) = 1$ for all t because then the approximate “stochastic” program we form on a finite scenario tree has a single scenario. Within that approximating model we know the future when we making hydro and thermal generation decisions. So, the resulting estimate of the system’s cost is optimistic. More precisely, when $n(t) = 1$ for all t , $\mathbb{E}\hat{z}^*$ is the so-called wait-and-see bound. The intuition behind the result $\mathbb{E}\hat{z}^* \leq z^*$ is similar when $n(t) > 1$. We have sampled a finite

number of inflow scenarios and the approximate stochastic program defined on this finite scenario tree effectively “over-optimizes” by adapting its decisions to the specific scenarios that have been sampled. The approximation’s ability to over-optimize in this manner diminishes as $n(t)$ grows.

In principle one way to form a lower bound estimator on z^* is to: (i) form n_ℓ i.i.d. scenario trees via SSTIDM, (ii) solve the associated stochastic programs to obtain $\hat{z}^{*,i}$, $i = 1, \dots, n_\ell$, and (iii) form the associated sample mean and sample variance estimators of $\mathbb{E}\hat{z}^*$. Unfortunately, we cannot computationally carryout this procedure because we cannot solve the approximate stochastic program exactly for the values of T and $n(t)$ we have in mind. However, the SDDP algorithm produces a lower bound on \hat{z}^* , namely \underline{z} , and we can replace $\hat{z}^{*,i}$ above with \underline{z}^i to yield a computable lower bound estimator. Note that the bound \underline{z} is a deterministically valid lower bound on \hat{z}^* , given that $\hat{\Omega}_T$ is the sample space. We summarize the procedure below.

Input: Instance of model (3.22). Branch size $n'(t)$, $t = 1, \dots, T - 1$.
Sample size n_ℓ .

Output: Sample mean estimator, L_{n_ℓ} , and sample variance estimator, S_ℓ^2 , of lower bound on optimal value of (3.22), z^* .

1. do $i = 1, \dots, n_\ell$
 - run SSTIDM to form a sample tree with $n'(t)$ branches at stage t , a tree independent of those in other iterations;
 - run SDDP and obtain lower bound on optimal value \underline{z}^i ;
- enddo

$$\begin{aligned}
2. \quad & \text{let } L_{n_\ell} = \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \underline{z}^i; \\
& \text{let } S_\ell^2 = \frac{1}{n_\ell - 1} \sum_{i=1}^{n_\ell} (\underline{z}^i - L_{n_\ell})^2; \quad \square
\end{aligned}$$

The LBE procedure forms a lower bound estimator on z^* satisfying $\mathbb{E}L_{n_\ell} \leq z^*$. We run SSTIDM with a branch size $n(t)$ to provide the input to SDDP when we wish to form a high quality policy. We can use a smaller branch size $n'(t)$ when running LBE in order to produce a lower bound to assess the quality of that policy. If the branch size, $n'(t)$, used in LBE is too small the bias $\mathbb{E}L_{n_\ell} - z^*$ can be too large in magnitude in order to produce a reasonable statement regarding the policy's quality. If LBE's branch size, $n'(t)$, is too large then the computational effort needed to carryout the LBE procedure can be excessive. We examine further this trade-off in our computational results of Section 4.5.

4.3 Confidence Interval Construction on Policy Quality

In this section, we build on ideas of Chiralaksanakul and Morton (2004) in forming a confidence interval on the optimality gap. Consider the policy specified by the cuts output by SDDP, $(\vec{G}_t^{sddp}, \vec{g}_t^{sddp})$, $t = 1, \dots, T$. The policy specifies the action we take on an arbitrary sample path from Ω_T . That is, let b_1, \dots, b_T be a random sample path selected from Ω_T . Then we solve the associated sequence of stage t problems (3.26) with b_t , $\vec{G}_t = \vec{G}_t^{sddp}$ and $\vec{g}_t = \vec{g}_t^{sddp}$,

and with $x_{t-1} = x_{t-1}(b_1, \dots, b_{t-1})$. The random variable

$$U = \sum_{t=1}^T c_t x_t(b_1, \dots, b_t) \quad (4.2)$$

is the random cost of the policy on path b_1, \dots, b_T . The policy has expected cost $\mathbb{E}U \geq z^*$. If $\mathbb{E}U - z^*$ is small then the policy is of high quality. Of course, we cannot compute $\mathbb{E}U$ and we do not know z^* . However, the cost estimator for the PGP of Section 4.1 and the lower bound estimators of Section 4.2 allow us to form a one-sided confidence interval that bounds $\mathbb{E}U - z^*$ from above.

Let U_{n_u} denote the point estimate, and S_u^2 be the sample variance estimator, output by PGP. We have $\mathbb{E}U_{n_u} = \mathbb{E}U$ and

$$\mathbb{P} \left\{ \mathbb{E}U \leq U_{n_u} + z_\alpha \frac{S_u}{\sqrt{n_u}} \right\} = \mathbb{P} \left\{ \frac{\sqrt{n_u}(U_{n_u} - \mathbb{E}U_{n_u})}{S_u} \geq -z_\alpha \right\},$$

where z_α denotes the $(1 - \alpha)$ -level quantile of a standard normal random variable. By the central limit theorem for i.i.d. random variables,

$$\lim_{n_u \rightarrow \infty} \mathbb{P} \left\{ \frac{\sqrt{n_u}(U_{n_u} - \mathbb{E}U_{n_u})}{S_u} \geq -z_\alpha \right\} = 1 - \alpha.$$

Hence, for sufficiently large n_u , we infer an approximate one-sided $100 \cdot (1 - \alpha)\%$ confidence interval for $\mathbb{E}U$ of the form $(-\infty, U_{n_u} + z_\alpha S_u / \sqrt{n_u}]$.

Let L_{n_ℓ} and S_ℓ^2 be estimators produced by the LBE procedure. Since $z^* \geq \mathbb{E}L_{n_\ell}$,

$$\begin{aligned} \mathbb{P} \left\{ z^* \geq L_{n_\ell} - t_{n_\ell-1, \alpha} \frac{S_\ell}{\sqrt{n_\ell}} \right\} &\geq \mathbb{P} \left\{ \mathbb{E}L_{n_\ell} \geq L_{n_\ell} - t_{n_\ell-1, \alpha} \frac{S_\ell}{\sqrt{n_\ell}} \right\} \\ &= \mathbb{P} \left\{ \frac{\sqrt{n_\ell}(L_{n_\ell} - \mathbb{E}L_{n_\ell})}{S_\ell} \leq t_{n_\ell-1, \alpha} \right\}. \end{aligned}$$

Again by the central limit theorem for i.i.d. random variables,

$$\lim_{n_\ell \rightarrow \infty} \mathbb{P} \left\{ \frac{\sqrt{n_\ell}(L_{n_\ell} - \mathbb{E}L_{n_\ell})}{S_\ell} \leq t_{n_\ell-1, \alpha} \right\} = 1 - \alpha.$$

So we infer, for sufficiently large n_ℓ , that $[L_{n_\ell} - t_{n_\ell-1, \alpha} S_\ell / \sqrt{n_\ell}, \infty)$ is an approximate one-sided $100 \cdot (1 - \alpha)\%$ confidence interval for z^* . Note that this confidence interval is valid for any value of the branch size, $n'(t)$. Here $t_{n_\ell-1, \alpha}$ denotes the $(1 - \alpha)$ -level quantile of a Student t random variable with $n_\ell - 1$ degrees of freedom.

We can combine the events $\{L_{n_\ell} - t_{n_\ell-1, \alpha} S_\ell / \sqrt{n_\ell} \leq z^*\}$ and $\{\mathbb{E}U \leq U_{n_u} + z_\alpha S_u / \sqrt{n_u}\}$ using the Boole-Bonferroni inequality to obtain

$$\mathbb{P}\{L_{n_\ell} - t_{n_\ell-1, \alpha} S_\ell / \sqrt{n_\ell} \leq z^*, \mathbb{E}U \leq U_{n_u} + z_\alpha S_u / \sqrt{n_u}\} \approx 1 - 2\alpha.$$

This coupled with

$$\begin{aligned} \mathbb{P}\{L_{n_\ell} - t_{n_\ell-1, \alpha} S_\ell / \sqrt{n_\ell} \leq z^*, \mathbb{E}U \leq U_{n_u} + z_\alpha S_u / \sqrt{n_u}\} &\leq \\ \mathbb{P}\{\mathbb{E}U - z^* \leq (U_{n_u} - L_{n_\ell}) + t_{n_\ell-1, \alpha} S_\ell / \sqrt{n_\ell} + z_\alpha S_u / \sqrt{n_u}\} &\leq \\ \mathbb{P}\{\mathbb{E}U - z^* \leq (U_{n_u} - L_{n_\ell})^+ + t_{n_\ell-1, \alpha} S_\ell / \sqrt{n_\ell} + z_\alpha S_u / \sqrt{n_u}\} & \end{aligned}$$

yields validity of the confidence interval formalized in the following procedure.

Here, $(\cdot)^+ = \max(\cdot, 0)$.

Confidence Interval Construction (CIC):

Input: Instance of model (3.22) under dependency model (3.24).
Branch sizes $n(t)$ for policy construction and $n'(t)$ for lower bound estimation (LBE), sample sizes n_ℓ and n_u , and $\alpha \in (0, 1)$.

Output: Approximate $(1 - \alpha)$ -level confidence interval on optimality gap $\mathbb{E}U - z^*$.

1. run SSTIDM to form a sample scenario tree with branch sizes $n(t)$;
2. run SSDP to approximately solve (3.22) defined on the sample scenario tree to obtain $(\vec{G}_t^{sddp}, \vec{g}_t^{sddp})$, $t = 1, \dots, T$, x_1 , and \underline{z} ;
3. run PGP with sample size n_u to obtain U_{n_u} and S_u^2 ;
4. run LBE with branch sizes $n'(t)$ and sample size n_ℓ to obtain L_{n_ℓ} and S_ℓ^2 ;
5. let $\epsilon_\ell = t_{n_\ell-1, \alpha} S_\ell / \sqrt{n_\ell}$ and $\epsilon_u = z_\alpha S_u / \sqrt{n_u}$;
output one-sided CI on $\mathbb{E}U - z^*$, $[0, (U_{n_u} - L_{n_\ell})^+ + \epsilon_\ell + \epsilon_u]$; \square

4.4 Tightening the Confidence Interval

Below we list four reasons that we might not obtain satisfactory results when running the CIC procedure:

- (a) Under LBE: The computational effort to run SDDP on (say) $n_\ell = 15$ -20 instances of model (3.22) is prohibitive;
- (b) the bias of L_{n_ℓ} , from LBE, is large;
- (c) the sampling error, ϵ_ℓ and/or ϵ_u , is large; or,
- (d) the candidate policy is far from optimal to (3.22).

Issues (b), (c) and (d) can produce a confidence interval that is unacceptably large, and issue (a) concerns computational tractability. Before turning to how we can diagnosis which issue is the foremost problem, we discuss the related matter of how the inputs to CIC affect each issue. We do this separately for the LBE and LBE2 variants of CIC.

CIC under LBE: The inputs are $n(t)$, the branch size for producing a policy; $n'(t)$, the branch size for producing the lower bound estimator; n_ℓ , the number of replicates for the lower bound estimator; and, n_u , the sample size for the upper bound estimator. Increasing $n(t)$ primarily affects (d). Increasing $n(t)$ tends to increase the quality of the cut-based policy, and also increases the computational effort required to solve the instance of model (3.22) used to produce that policy. (The variance of U differs under different policies, and hence increasing $n(t)$ can affect ϵ_u in (c) but this cannot be expected to behave monotonically as the policy improves.) Increasing $n'(t)$ affects (a), (b) and (c). Increasing $n'(t)$ decreases the bias of L_{n_ℓ} , tends to decrease the sampling error, ϵ_ℓ , and increases the computational effort discussed in (a). We usually fix $n_\ell = 15-20$, but increasing n_ℓ affects (c), decreasing sampling error ϵ_ℓ . Increasing n_u affects (c), decreasing sampling error, ϵ_u .

The assessment of which issue is of primary importance can be approached as follows. The confidence interval width consists of $(U_{n_u} - L_{n_\ell})^+ + \epsilon_\ell + \epsilon_u$. The first term, $(U_{n_u} - L_{n_\ell})^+$, is the point estimate of the optimality gap, ϵ_ℓ is the sampling error associated with the lower bound estimator and ϵ_u is that associated with the upper bound estimator. If ϵ_ℓ or ϵ_u dominates

then we know the primary issue is (c). If $(U_{n_u} - L_{n_\ell})^+$ dominates it could be because $\mathbb{E}U$ is too large, i.e., the policy is poor—issue (d)—or because $\mathbb{E}L_{n_\ell}$ is too small, i.e., the bias is large—issue (b). With CIC under LBE, we can solve two or three instances of (3.22), starting with a small value of $n(t)$ and allowing it to grow to produce two or three policies and associated upper bound estimators. And, we can solve multiple instances of (3.22) with growing $n'(t)$ to form two or three lower bound estimators. The focus is then which is changing more quickly. Is the lower bound increasing more quickly (i.e., issue (b) is foremost) or is the upper bound decreasing more quickly (i.e., issue (d) is foremost)?

Although we do not pursue it here, another option is to sample in a non-i.i.d. manner when forming the sample scenario tree in SSTIDM. This could be for the scenario tree used to produce the policy and/or the scenario trees used in forming the lower bound estimator. Two promising approaches are to use Latin hypercube sampling (LHS) or randomized quasi Monte Carlo (RQMC). RQMC and LHS can be applied to problems in which the scalar components of a random vector are independent (or are functions of a component-wise independent vector). The idea is to more uniformly spread the $U(0, 1)$ random variates that are used, e.g., through inversion, to form the nonuniform random variables. When forming a multivariate normal, or multivariate lognormal, we do so using a linear transformation of a collection of underlying independent standard normal random variables. Hence, the dependent components η_t of the interstage dependency model (3.24) are amenable to LHS and RQMC.

Application of RQMC in stochastic programming is examined in Diwekar and Kalagnanam (1997) and Pennanen and Koivu (2005). Homem-de-Mello Homem-de-Mello (2008) provides conditions under which pointwise (i.e., without the optimization operator) rates of convergence associated with non-i.i.d. sampling schemes like RQMC and LHS are inherited by the solution and value estimators in the two-stage setting. LHS is also applied in Bailey et al. (1999); Diwekar and Kalagnanam (1997); Freimer et al. (2005); and, Linderoth et al. (2006).

4.5 Solution Quality Analysis

We assess the solution quality for a hydro thermal scheduling problem for the mid-horizon (HTSPM). We optimize over 24 monthly stages considering the Brazilian interconnected power system that we first describe in Chapter 2 and describe further in Chapter 5. We compute the upper bound estimator using a large number ($n_u = 12800$) of independent forward paths and the lower bound estimator LBE using $n_\ell = 15$ independent trees. We consider a dynamic linear model (DLM) from Marangon Lima (2011) to generate the inflow scenarios and create interstage dependent scenario trees according to SSTIDM. The DLM satisfies the additive form (3.24) that we introduce in Chapter 3.

We have carried out the solution quality analysis that we describe in this chapter on several problem instances. These instances differ only in: (a) the same sample sizes, $n(t)$, at each stage in the sample scenario tree that we use to form the cuts that define the solution policy via PGP, which in turn

feed into the upper bound estimator, U_{n_u} , and (b) the sample sizes, $n'(t)$ used in the sample scenario trees that define the lower-bound estimator, L_{n_ℓ} . We solve a total of seven such problem instances with growing values of $n(t)$ and $n'(t)$. Our smallest instances use $n(t) = 10$ and $n'(t) = 8$ and then $n(t) = 20$ and $n'(t) = 15$. Here, these sample sizes denote the number of branches at each stage in the scenario trees, and we use identical values for all of the stages. However, in our larger sampled scenario trees we use a larger number of samples in the early iterations and use fewer as the stage index grows larger. Specifically, for the tree used to generate the policy, we use equation (4.3):

$$n(t) = \max\{\rho^{t-1}n(1), n_{\min}\}. \quad (4.3)$$

And, we use this same equation with $n(1)$ and $n(t)$ replaced by $n'(1)$ and $n'(t)$ for the LBE procedure. We use $\rho = 0.8$ and $n_{\min} = 20$ and range the values of $n(1)$ from 60 up to 2000 and $n'(1)$ from 45 up to 1500.

Appendix A gives detailed results for each of the seven sets of scenario trees we consider. The next four sections report a summary of these results.

4.5.1 Upper Bound Estimator Analysis

In this section we analyze the results of all the simulated instances for the outputs of the PGP. Table 4.1 presents the upper bound estimator, U_{n_u} , and the confidence interval width, ϵ_u , as the number of branches, $n(t)$, used to define the scenario tree that yields the PGP grows. The ‘‘Branches’’ column indicates the value of $n(1)$. When $n(1) = 10$ and $n(1) = 20$, the same branch

size is used in all stages $t = 1, \dots, T - 1$. When $n(1) > 20$, we use formula (4.3). For the results in Table 4.1, we use a confidence level of $1 - \alpha = 0.95$, and each row is based on $n_u = 12800$.

| Branches | U_{n_u} | ϵ_u | $\epsilon_u\%$ |
|----------|-----------|--------------|----------------|
| 10 | 4755388.8 | 62257.5 | 1.3% |
| 20 | 4182477.9 | 63699.1 | 1.5% |
| 60 | 4256700.1 | 58379.2 | 1.4% |
| 100 | 4171942.8 | 63322.1 | 1.5% |
| 200 | 4120735.7 | 61167.5 | 1.5% |
| 1000 | 4090672.5 | 58522.3 | 1.4% |
| 2000 | 4076822.7 | 58511.5 | 1.4% |

Table 4.1: Upper Bound Estimator for PGPs From Scenario Trees of Growing Size

The $n_u = 12800$ forward paths for estimating U_{n_u} in Table 4.1 are all identical; i.e., we form these estimators using common random numbers. Figure 4.1 shows the upper bound estimator and the confidence interval width for each scenario-tree size. Table 4.1 and Figure 4.1 suggest that as the size of the scenario tree used to define the cuts for PGP grows, the estimated cost, U_{n_u} , tends to decrease. The sampling error does not shrink because we estimate the sample variance with a constant sample size, $n_u = 12800$.

We carried out a paired Student- t test in order to assess whether the apparent decrease in cost with larger scenario trees is statistically significant. Table 4.2 shows the results of this test. A positive difference for the point estimate indicates a larger cost when the scenario tree has a smaller number of branches. With one exception (when the number of branches grows from 20

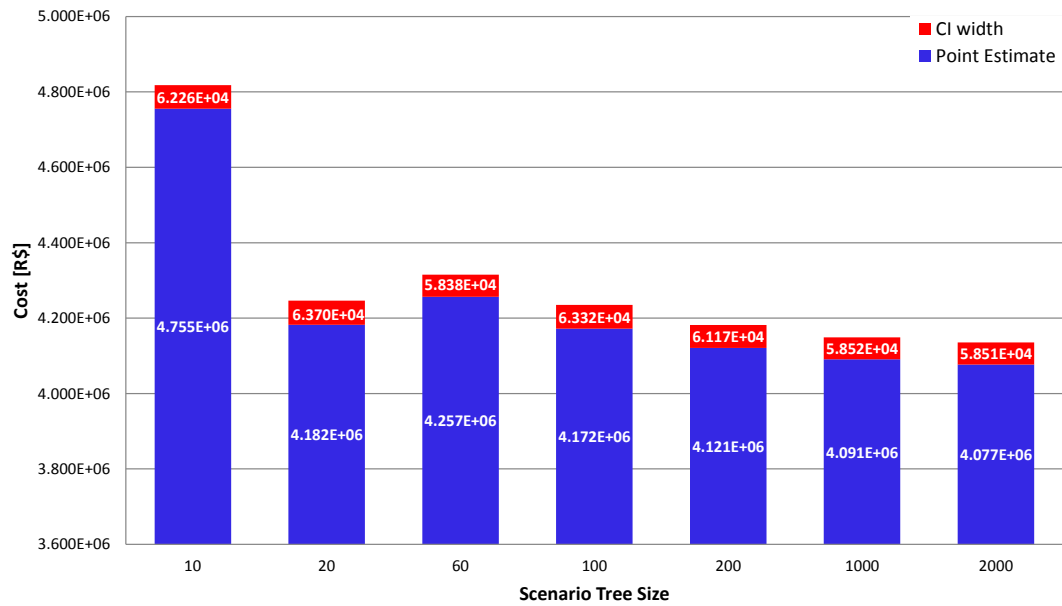


Figure 4.1: Upper Bound Point Estimate and CI Width with Scenarios Trees of Various Size

to 60), larger trees yield smaller estimates of cost. All of the point estimates of those differences are, individually, statistically significant at a level of $1 - \alpha = 0.90$. The significant variance reduction that we obtain by using common random numbers in estimating these differences can be seen by comparing the halfwidth values (HW) in Table 4.2 with the CI widths (ϵ_u) reported in Table 4.1.

| Branches | | 10 | 20 | 60 | 100 | 200 | 1000 |
|-----------------|----|-----------|-----------|-----------|------------|------------|-------------|
| 20 | Pt | 572910.9 | | | | | |
| | HW | 23540.8 | | | | | |
| 60 | Pt | 498688.8 | -74222.1 | | | | |
| | HW | 17604.6 | 13660.4 | | | | |
| 100 | Pt | 583446.1 | 10535.1 | 84757.3 | | | |
| | HW | 23811.1 | 9626.8 | 13067.4 | | | |
| 200 | Pt | 634653.2 | 61742.2 | 135964.4 | 51207.1 | | |
| | HW | 23365.3 | 10191.1 | 10923.7 | 7531.9 | | |
| 1000 | Pt | 664716.3 | 91805.4 | 166027.5 | 81270.2 | 30063.1 | |
| | HW | 23698.3 | 11212.3 | 10373.0 | 9103.9 | 5905.1 | |
| 2000 | Pt | 678566.1 | 105655.2 | 179877.3 | 95120.1 | 43913.0 | 13849.8 |
| | HW | 24116.9 | 11058.8 | 10742.0 | 8959.6 | 6019.8 | 4668.8 |

Table 4.2: Paired Student- t Test for PGP with Different Scenario-Tree Sizes
Paired Student- t tests using common random numbers and a 90% level with a sample size of 12800. The table contains confidence intervals for the column entry less the row entry; e.g., the first entry is 572910.9 ± 23540.8 is a confidence interval for $U_{n_u}(10) - U_{n_u}(20)$, where $U_{n_u}(10)$ and $U_{n_u}(20)$ denote the point estimates from scenario trees with $n(1) = 10$ and $n(1) = 20$ branches at each stage, respectively.

4.5.2 Lower Bound Estimator Analysis

In this section we analyze the results of all the simulated instances for the outputs of the LBE procedure. Table 4.3 presents the lower bound estimator, L_{n_ℓ} , and the confidence interval width ϵ_ℓ , as the number of branches, $n'(t)$, used to define the scenario trees in LBE grows. The “Branches” column indicates $n'(1)$. When $n'(1) = 8$ and $n'(1) = 15$, the same branch size is used in all stages $t = 1, \dots, T - 1$. When $n'(1) > 15$, we use formula (4.3). For the results in Table 4.3, we use a confidence level of $1 - \alpha = 0.95$, and each row is based on $n_\ell = 15$.

| Branches | L_{n_ℓ} | ϵ_ℓ | $\epsilon_\ell\%$ |
|----------|--------------|-----------------|-------------------|
| 8 | 3500672.5 | 595878.0 | 12.5% |
| 15 | 4166823.0 | 599633.0 | 14.3% |
| 45 | 4011643.7 | 246636.0 | 5.8% |
| 75 | 4032745.6 | 135208.5 | 3.2% |
| 150 | 3952628.1 | 110963.3 | 2.7% |
| 750 | 3924632.1 | 90483.6 | 2.2% |
| 1500 | 3922280.6 | 67669.3 | 1.7% |

Table 4.3: Lower Bound Estimator LBEs From Scenario Trees of Growing Size

Consider $\underline{z}^i(n'(1)), i = 1, \dots, n_\ell$ to be the i.i.d. lower bounds obtained for each scenario tree with branch size $n'(1)$. We define the order statistics for the lower bounds obtained for each scenario tree with branch size $n'(1)$ as $\underline{z}^{(1)}(n'(1)) \leq \underline{z}^{(2)}(n'(1)) \leq \dots \leq \underline{z}^{(n_\ell)}(n'(1))$. Now, using the order statistics we can plot the different values of the obtained lower bounds in ascending order. Figure 4.2 shows the lower bound for each of $n_\ell = 15$ trees for each

scenario tree size in ascending order. Notice that the variability reduces and the lines are becoming flatter and flatter around roughly 4.0×10^6 as the scenario-tree size grows.

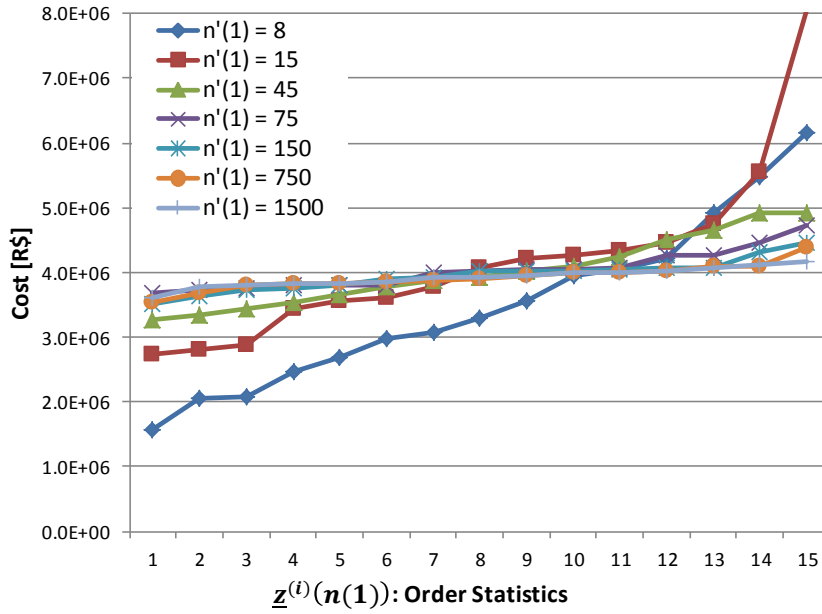


Figure 4.2: L_{n_ℓ} in Ascending Order for Each Scenario Tree Size

Figure 4.3 shows the lower bound estimator L_{n_ℓ} and the confidence interval width for each scenario tree size. It is possible to notice that the lower bound estimator is increasing and the confidence interval width is decreasing in value as the size of the sampled scenario trees grows. The estimator appears to grow until it reaches a certain value and then it stays in the neighborhood of that value as the scenario tree grows. The confidence interval width, on the other hand, is decreasing as the size of the scenario trees grows.

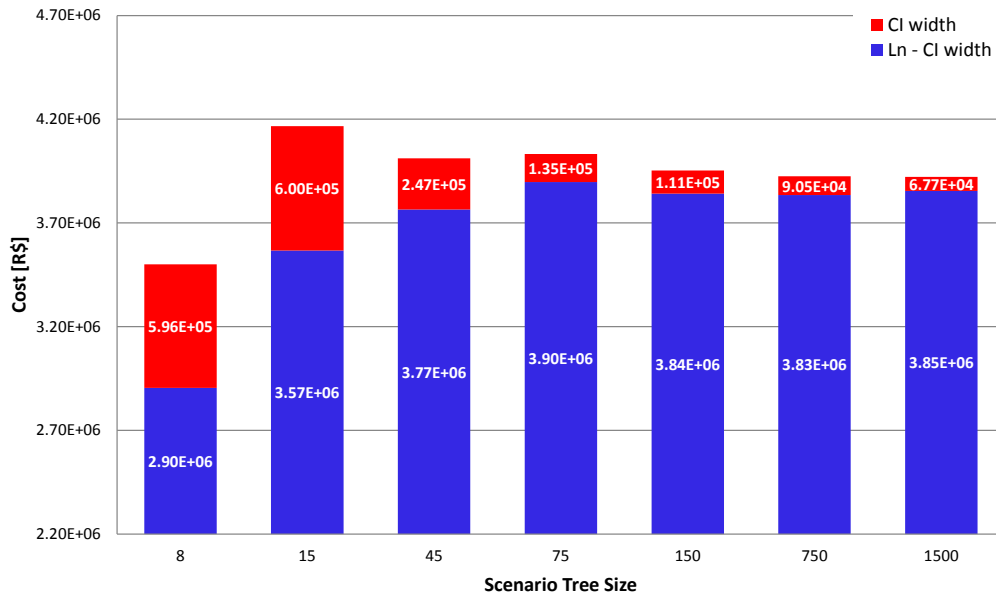


Figure 4.3: LBE Point Estimate and Halfwidth for Each Instance

4.5.3 Confidence Interval Analysis

Now we analyze the results of all the simulated instances for the confidence interval errors. Table 4.4 presents the point estimate of the optimality gap, $(U_{n_u} - L_{n_\ell})^+$, along with the confidence interval width, $(U_{n_u} - L_{n_\ell})^+ + \epsilon_\ell + \epsilon_u$, for scenario trees of increasing size. The first column, labeled “Branches,” shows the sample size at each stage of the scenario trees, as described in the previous two sections.

Figure 4.4 present the errors associated with the upper bound, lower bound and gap between $\mathbb{E}U$ and z^* for each instance. It is possible to notice that the confidence interval error reduces as the sample size becomes bigger. The confidence interval width shrinks as the scenario trees grow in size. As the

| Branches | $gap[R\$]$ | $gap\%$ | CI for $\mathbb{E}U - z^*$ | CI% |
|-------------|------------|---------|----------------------------|-------|
| 10 / 8 | 1254716.3 | 26.4% | 1912851.8 | 40.2% |
| 20 / 15 | 15654.9 | 0.4% | 678987.1 | 16.2% |
| 60 / 45 | 245056.4 | 5.8% | 550071.6 | 12.9% |
| 100 / 75 | 139197.1 | 3.3% | 337727.7 | 8.1% |
| 200 / 150 | 168107.6 | 4.1% | 340238.4 | 8.2% |
| 1000 / 750 | 166040.4 | 4.1% | 315046.3 | 7.7% |
| 2000 / 1500 | 154542.1 | 3.8% | 280722.9 | 6.9% |

Table 4.4: Gap Estimator and CI Width From Scenario Trees of Growing Size

scenario tree size (based on $n(1)$) used for forming the policy grows, the point estimate of upper bound tends to decrease. As the scenario tree size (based on $n'(1)$) used in LBE grows, the lower bound tends to grow. Together this means that the gap estimate (shown in green) tends to shrink. The sampling error associated with the upper bound is relatively constant (shown in blue) because the sample size for out-of-sample testing ($n_u = 12800$) is held constant. Finally as $n'(t)$ grows the sampling error associated with the lower bound (shown in red) tends to decrease.

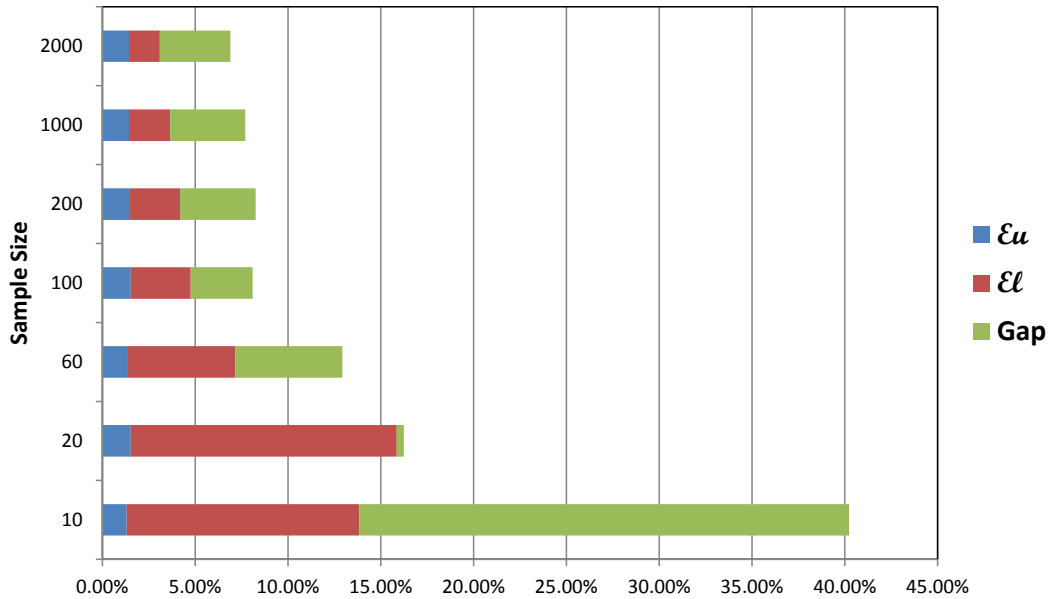


Figure 4.4: Confidence Interval Errors for Each Instance

4.5.4 Computational Time Analysis

Now we analyze the computational time regarding to the sampled scenario tree used to define the collection of cuts for PGP for the different tree sizes. Figure 4.5 presents the computation time for each iteration for different sample sizes. Clearly, we notice that the computational effort grows as the branch size in the scenario trees grows. The computational effort per iteration grows as we add more cuts to the subproblems at each stage because the constraint matrix grows in size.

Figure 4.6 presents the total computation time for different sample sizes. Again, it is clear that the total computational time is growing with

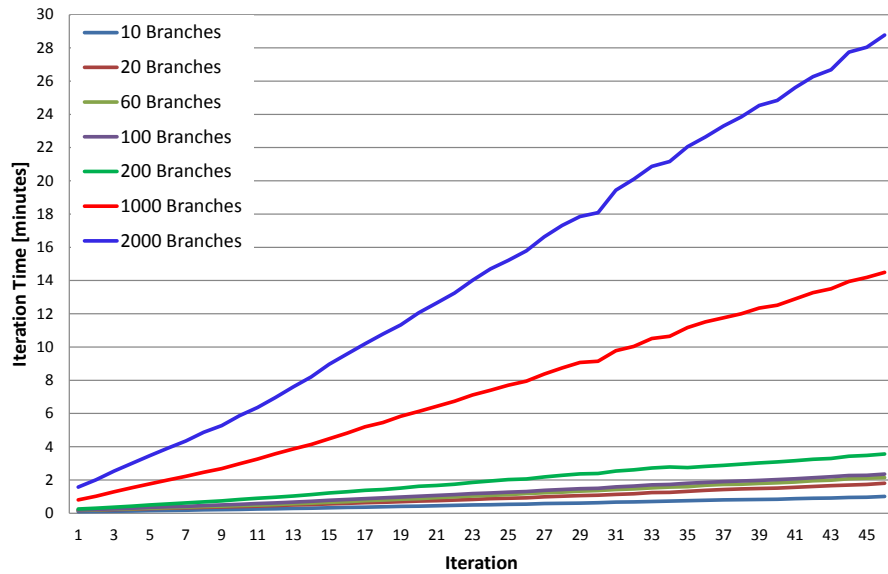


Figure 4.5: Iteration Time in Minutes for Each Instance

larger scenario trees. Note that to run the SDDP for 45 iterations using $n(1) = 2000$ for the first stage and $n(t)$ defined by (4.3) for $t = 2, \dots, T - 1$ takes approximately 660 minutes or 11 hours using 128 cores.

Although the results of the solution quality are better when we have bigger sample sizes, the computational time is also longer. For the remainder of this dissertation, in Chapter 5, we use $n(1) = 200$ for the first stage and $n(t)$ defined as in (4.3) for $t = 2, \dots, T - 1$.

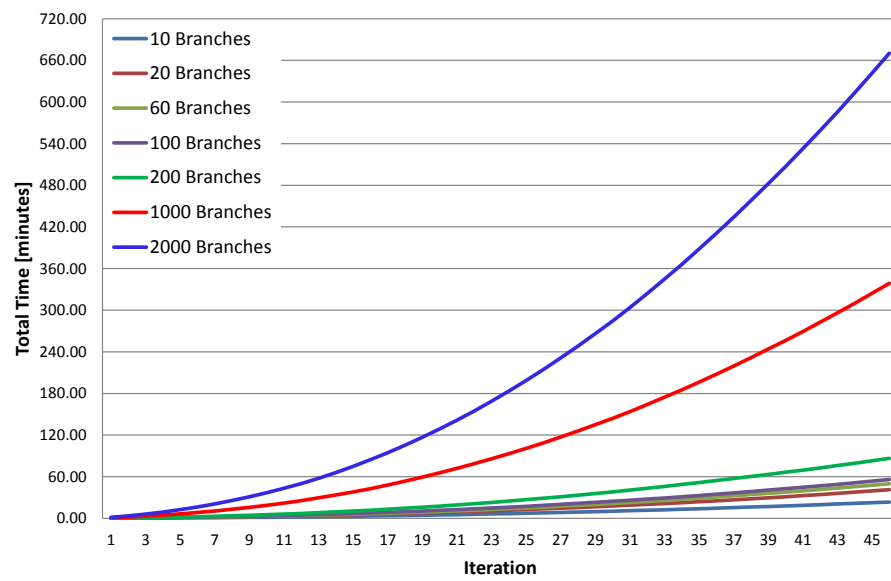


Figure 4.6: Total Time in Minutes for Each Instance

Chapter 5

Case Study

The case study we describe in this chapter is focused on the Brazilian interconnected power system. We describe a base case and consider three excursions from that base case by including new run-of-river hydro generation, additional transmission capacity, and by increasing system demand. We used the model for the hydro-thermal scheduling problem with the aggregate reservoir representation describe in Chapter 2 using water inflows forecasting from a DLM model that provides an interstage dependent structure that satisfies the additive form (3.24). As in Chapter 4, we define 24 monthly stages for this medium horizon problem and we consider the formulation using the cut-sharing methodology developed in Section 3.8.5 to solve the problem. Before we go into more detail about each instance we describe in the next section the main characteristics of the Brazilian interconnected power system.

5.1 The Brazilian Interconnected Power System

The Brazilian interconnected power system is different from most other countries around the world. The major portion of electricity is produced by hydro generators. The system is divided in four regions, they are: Region 1 - South

East / Central, Region 2 - South, Region 3 - North East and Region 4 - North. These four regions represent around 98% of the Brazilian electricity demand, and the other 2% is isolated from the rest of the system. Figure 5.1 presents each of the Brazilian regions considered by the independent system operator when solving the hydro-thermal scheduling problem.

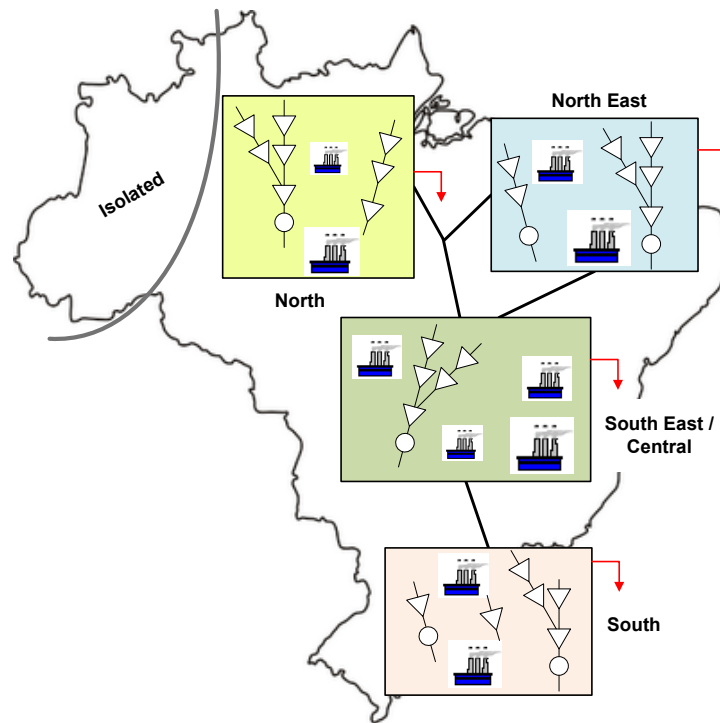


Figure 5.1: Brazilian Power System Regions

5.1.1 Power Generation Production and Capacity

In 2010, around 89% of the total electricity was produced by hydro plants (ONS, 2011). Figure 5.2 presents the percentage of the total energy production

produced by each source during the year 2010.

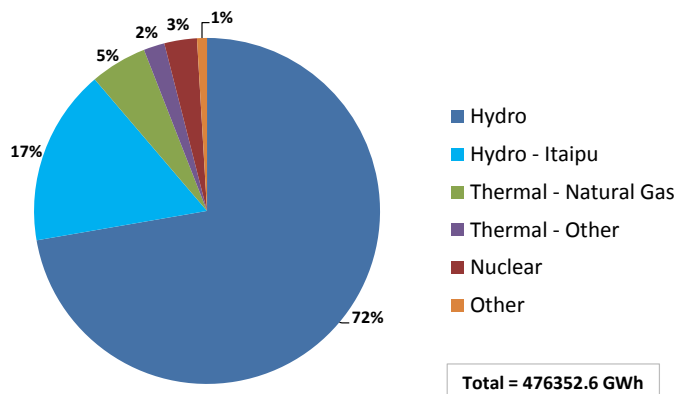


Figure 5.2: Brazil's Total Electricity Production in 2010

Region 1 (that includes the Itaipu hydro generator) is responsible for 63% of the total electricity production, Region 2 produces around 17.7%, Region 3 produces around 11.3% and Region 4 is responsible for 8.0% of the total electricity production. In terms of generation capacity the picture above changes. Figure 5.3 presents the percentage of the total generation capacity for each source.

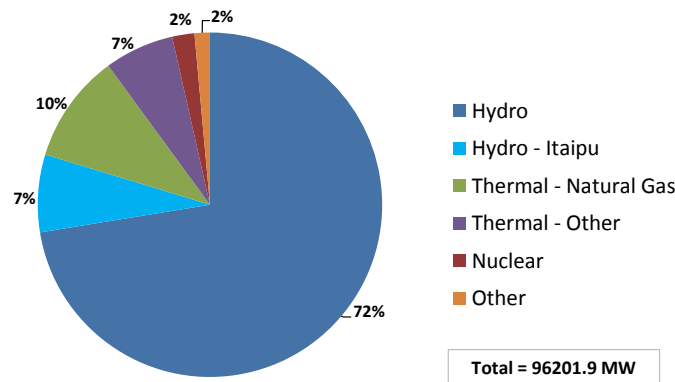


Figure 5.3: Brazil's Total Generation Capacity in 2010

5.1.2 Power System Demand

The average electricity demand for the interconnected power system during the year 2010 was 54,251.6 [MW]. Most of the electricity demand is located in Region 1, which represents the load center. Figure 5.4 presents the percentage of electricity demand required by each Region. The total energy consumption in 2010 was 475095.6 [GWh], and the percentages of each region are almost the same as the ones presented in Figure 5.4.

5.1.3 Interconnections

In our version of the medium horizon hydro-thermal scheduling problem we consider transmission lines that interconnect each region of the Brazilian power system. Transmission lines within each region are not considered in our model because of the aggregate reservoir representation that we assume. This is also the current procedure used by the Brazilian independent system operator.

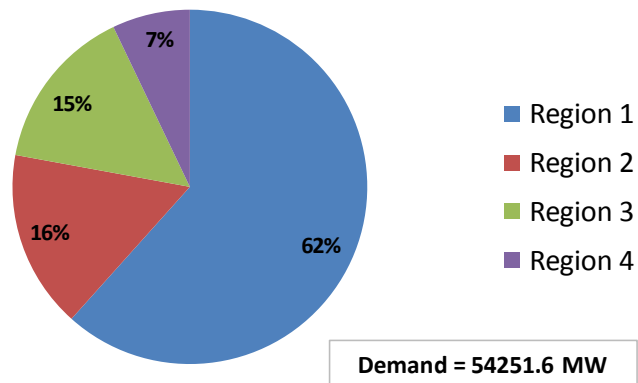


Figure 5.4: Percentage of Total Electricity Demand in 2010 for Each Region

From Figure 5.1 it is possible to see the interconnections considered in our model. Region 1 is connected to Region 2, to Region 3, and to a virtual Region that has connection to Region 4. Region 3 is also connected to the virtual Region. The transmission link capacities for the year 2010 are presented in Figure 5.5. Note that these values change for each month depending on the maintenance schedule for the transmission lines and also because of the addition of new lines into the system.

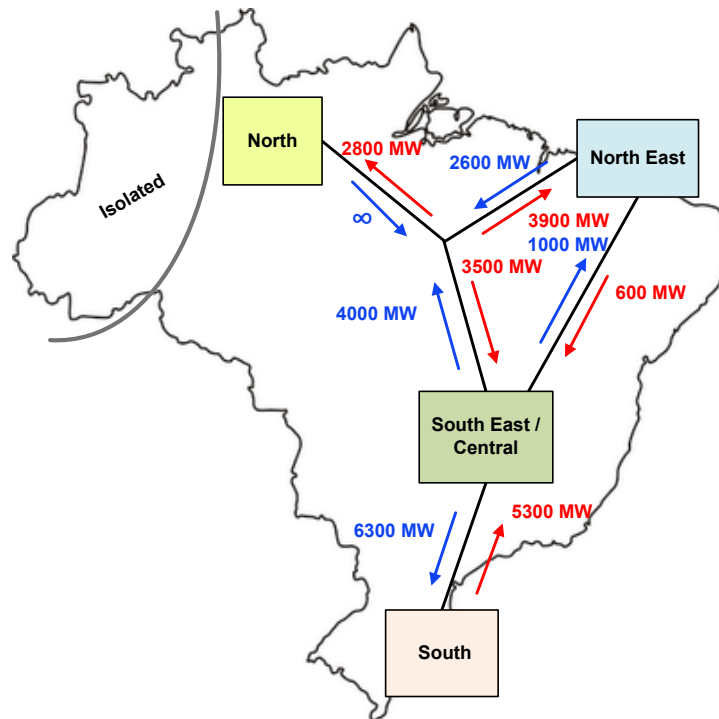


Figure 5.5: Transmission Capacities Between Regions for 2010

5.1.4 Simulation Considerations

As a base point for our case studies we consider the data available from the Brazilian Commercialization Chamber website (CCEE, 2010). The input data used is for the month of February of 2010. Because of the large amount of data (contained in 34 different files), part of the inputs description are suppressed in this document but one can download directly from that website.

5.1.4.1 Power System Demand

The power system demand changes from month to month for each region. Table C.1, in Appendix C, presents the power system demand for each region for the 24-month horizon. From the total demand of each region we subtract the electricity generation produced by small hydro plants, electricity produced by hydro generators operating with fewer machines than their normal capacity, and minimal thermal generation, which is inflexible. The result is the net demand for each region that is used in the optimization model.

There is a considerable amount of electricity generation from small hydro plants in the system which is not considered in the centralized dispatch by the ISO. Because of that, this amount of power is subtracted from the system demand presented in Table C.2, where the values represent the amount of generation produced by small hydro plants for each month.

We consider also the amount of power that is produced by hydro plants that are operating with fewer machines than their normal capacity (which is not counted in the energy parameters computation described in Chapter 2). These amounts are also subtracted from the demand, and they are presented in Table C.3.

The minimal thermal generation required from each thermal plant is also discounted from the demand. Table C.4 presents the minimal thermal generation, or inflexible thermal generation, for each month in each region.

After subtracting all these terms we get the net demand for each region

that is used as input for the optimization model. Table 5.1 presents the net demand for each region in each month.

| Month | Region 01 | Region 02 | Region 03 | Region 04 |
|-------|-----------|-----------|-----------|-----------|
| 1 | 31554.1 | 8894.0 | 7578.5 | 3733.0 |
| 2 | 32225.2 | 9035.3 | 7607.5 | 3755.0 |
| 3 | 29661.9 | 7956.4 | 7667.5 | 3859.0 |
| 4 | 28185.8 | 7646.8 | 7509.5 | 3933.0 |
| 5 | 27874.5 | 7575.8 | 7356.5 | 3939.0 |
| 6 | 28317.5 | 7504.8 | 7329.5 | 3912.0 |
| 7 | 28465.5 | 7335.3 | 7301.5 | 3954.0 |
| 8 | 28488.1 | 6974.4 | 7389.5 | 3967.0 |
| 9 | 29090.3 | 6988.4 | 7485.5 | 3950.0 |
| 10 | 28551.0 | 7055.0 | 7525.5 | 3931.0 |
| 11 | 28002.6 | 7113.0 | 7537.5 | 3888.0 |
| 12 | 30289.6 | 8194.9 | 7810.5 | 4160.0 |
| 13 | 31005.9 | 8462.3 | 7769.5 | 4019.7 |
| 14 | 31640.9 | 8543.1 | 7874.5 | 3886.4 |
| 15 | 30637.6 | 7985.0 | 7826.5 | 3889.4 |
| 16 | 28838.6 | 7560.8 | 7637.5 | 3845.1 |
| 17 | 28483.5 | 7501.8 | 7472.5 | 3852.1 |
| 18 | 28719.6 | 7491.0 | 7409.5 | 3700.8 |
| 19 | 29290.6 | 7347.0 | 7409.5 | 3745.8 |
| 20 | 29377.6 | 7176.4 | 7492.5 | 4248.0 |
| 21 | 29596.6 | 7358.4 | 7609.5 | 4230.0 |
| 22 | 29440.6 | 7609.4 | 7661.5 | 4840.4 |
| 23 | 29926.2 | 7845.1 | 7677.5 | 4706.4 |
| 24 | 32032.6 | 8565.9 | 8189.5 | 4871.4 |

Table 5.1: Net Demand for Each Region [MW-month] - Optimization Input

As we describe in Chapter 3, we consider three load levels (low, medium and high). The net demand from Table 5.1 is then transformed into demand for each load level using the load level durations and load level base multipliers from Tables C.5 and C.6 from Appendix C.

5.1.5 Hydro Generation and Energy Parameters

There are a total of 150 hydro generators and 151 thermal generators that participate in the hydro-thermal scheduling problem that we analyze. Note that the hydro and thermal power system configuration is dynamic, changing for each month depending on the addition of new generators in the system. The generators also have scheduled maintenance that influence the production capacity for each source for different months. The hydro generators that are part of the centralized dispatch on the Brazilian system are presented in each river basin in Appendix C.

| Region 01 | Region 02 | Region 03 | Region 04 |
|-----------|-----------|-----------|-----------|
| 146959.21 | 18396.65 | 37344.75 | 10707.87 |

Table 5.2: Initial Energy Storage for Each Aggregate Reservoir [MW-month]

Table 5.2 presents the initial hydro energy storage on each aggregate reservoir that represent each region. Figure 5.6 presents the maximum hydro generation as functions of the storage for the first month of the problem. We present the other energy functions of the storage in Appendix C. Figures C.2, C.3 and C.4 present the minimal outflow energy, evaporation losses and water diversion functions of the storage for each region in the first month respectively. Note that these functions change from region to region depending on the hydro system configuration.

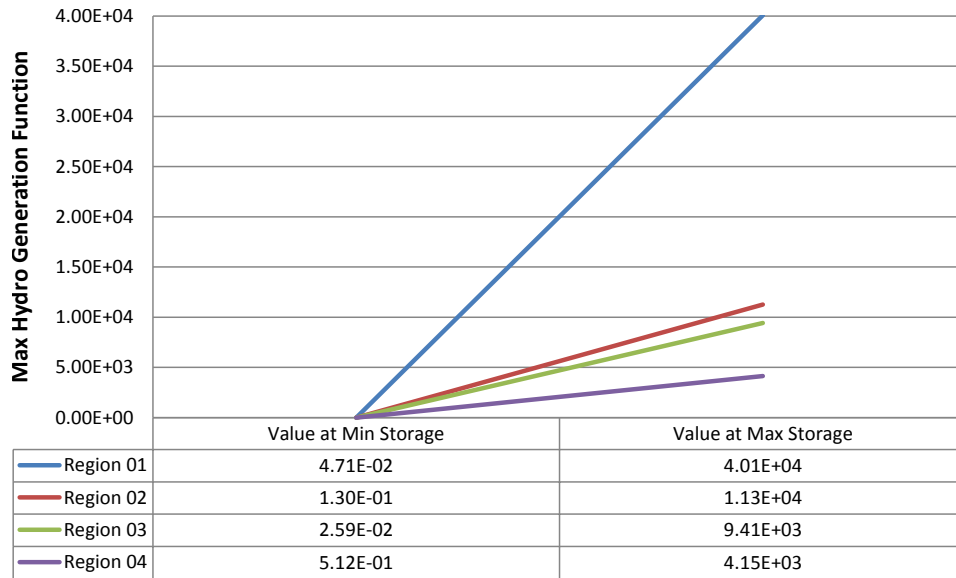


Figure 5.6: Max Hydro Generation Functions for Each Region - 1st Month

5.1.6 Thermal Generation and Costs

The maximum thermal generation capacity, already reduced by the minimal thermal generation from Table C.4, for each region is presented in Table 5.3. Note that we deal with thermal generation using individual thermal plants, but because of the large number of generators we prefer to show the total thermal generation capacity for each region instead. Figure 5.7 show the thermal generation capacity versus cost for the first month for the Brazilian system. The thermal generation versus cost presented in Figure 5.7 changes from month to month, and we just present them for the first month here.

We consider also four different load deficit levels with different cost and different curtailment magnitudes. Table 5.4 presents the different load deficit

| Month | Region 01 | Region 02 | Region 03 | Region 04 |
|-------|-----------|-----------|-----------|-----------|
| 0 | 4794 | 1235 | 2305 | 153 |
| 1 | 4749 | 1057 | 2305 | 306 |
| 2 | 4586 | 680 | 2446 | 306 |
| 3 | 4598 | 780 | 2446 | 306 |
| 4 | 5218 | 1021 | 2446 | 306 |
| 5 | 4758 | 1002 | 2729 | 306 |
| 6 | 4933 | 858 | 3047 | 306 |
| 7 | 5164 | 858 | 3202 | 306 |
| 8 | 4940 | 858 | 3202 | 306 |
| 9 | 4940 | 1004 | 3358 | 306 |
| 10 | 6079 | 1197 | 3358 | 306 |
| 11 | 5158 | 1122 | 3106 | 306 |
| 12 | 5320 | 1210 | 3379 | 306 |
| 13 | 5340 | 1074 | 3379 | 306 |
| 14 | 5340 | 1017 | 3379 | 306 |
| 15 | 5473 | 1014 | 5053 | 306 |
| 16 | 5759 | 1014 | 5053 | 306 |
| 17 | 5473 | 1046 | 5053 | 306 |
| 18 | 5473 | 946 | 5378 | 306 |
| 19 | 5473 | 988 | 5053 | 306 |
| 20 | 5473 | 1112 | 5703 | 306 |
| 21 | 5472 | 1213 | 5703 | 363 |
| 22 | 5546 | 1171 | 5789 | 363 |
| 23 | 5781 | 1146 | 6210 | 705 |

Table 5.3: Max Thermal Generation Capacity for Each Region [MW-month]

costs and the associated levels of curtailment.

| | Level 01 | Level 02 | Level 03 | Level 04 |
|------------------|----------|----------|----------|----------|
| Cost [R\$/MW] | 1031.76 | 2225.84 | 4651.31 | 5285.67 |
| Load Curtailment | 5% | 5% | 10% | 80% |

Table 5.4: Load Deficit Levels and Costs

In order to discount future costs to present values a discount factor of 12% a year is used in the objective function coefficients that represent the

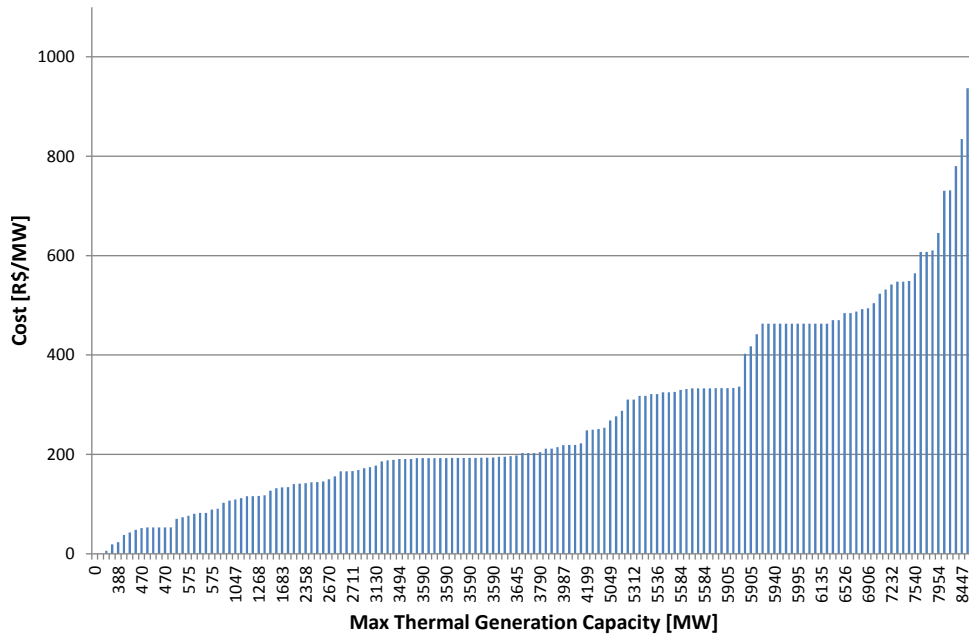


Figure 5.7: Thermal Generation Capacity x Cost - Brazilian System 1st Month

expected future cost.

5.1.7 Water Inflows

We consider the water inflows forecast produced by a dynamic linear model (DLM) from Marangon Lima (2011). Then a scenario tree is created according to SSTIDM procedure (see Chapter 3), and used in the optimization model. Note that once we have the water inflow forecasts and the productivity of each hydro plant the model computes all the necessary parameters, described in Chapter 2, for each stage and each scenario during the optimization process.

5.2 Problem Instances

In our case study, we solve four problem instances, in addition to the instance which we describe in Chapter 4. These problem instances differ from the base case in that they have increased hydro generation capacity, increased transmission capacity, and increased system load. After describing these instances, we examine their affect on key decision variables such as energy storage, controllable energy generation, thermal generation, power exchanges, and total expected cost to operate the system. We also compare some outputs of these new instances with the outputs of Instance 05 of Chapter 4.

5.2.1 Instance A - Base Case

Instance A is the basis of our analysis. It is an instance with 24 monthly stages with $n(t) = 200$ for $t = 1$ and $n(t)$ defined according to (4.3) for $t = 2, \dots, T - 1$, where we consider $\rho = 0.8$ and $n_{min} = 20$. We consider that the aggregate reservoir levels are initially at 50% of the values presented in Table 5.2 for the month of February of 2010. By choosing 50% we expect to create instances where we start from average hydro conditions and we believe that this assumption will represent that some flexible thermal generation has to be dispatched in order to satisfy demand. As a result, we should have positive total expected costs.

Based on our analyses in Chapter 4, we let the SDDP algorithm run for 45 iterations and then output the first stage solution and the collection of cuts to be used by the PGP procedure. We run PGP with $n_u = 12800$ in

order to evaluate the policy by computing the upper bound estimator U_{n_u} and the sampling error, ϵ_u . We consider the input parameters described in Section 5.1.4 to create this instance.

Instance 05 of Chapter 4 is the same as our base case, except that the reservoirs storage starts with 10% of their capacity instead of 50%. We denote this instance as Instance E for future reference.

5.2.2 Instance B - Addition of New Hydro Generation

In the input files obtained from CCEE (2010), there is information about new hydro generators that will start to operate up to five years in the future. One of these new generators is the Jirau run-of-river hydro plant to be installed in the system to supply the Southeast / Central Region in Brazil.

According to the information contained in the input files, the Jirau power generator is supposed to have the first machine in operation by February of 2013. This hydro plant has a total of 44 machines with nominal power of 75 MW each. The minimal number of machines specified for this hydro plant to start to contribute to the total energy production of the aggregate reservoir system is 27.

In instance B we decide to anticipate the startup operation date for this hydro plant in order to see the modifications in the total cost to operate the system. So, in order to do that we defined that the three first machines would start its operation in February of 2010, this means that we consider that the reservoir of this hydro plant is already complete when we start our

simulation. From March 2010 to October 2010 three new machines start to operate in each month, so by the beginning of October we already have the minimal number of machines in operation. This case represents the addition of 2025 MW of capacity that may produce hydroelectricity by October 2010 in comparison with the base case.

5.2.3 Instance C - Addition of New Transmission Lines

From the input files we also have information about new transmission lines that will be constructed in the near future. In January of 2012 new transmission lines will be start to operate and increase the transmission capacity between the Southeast / Central region and the South region. The new lines will represent an increase of 1000 MW in the direction South East to South and 500 MW in the other direction.

In order to create instance C, we decided to add 1000 MW of capacity in each direction beginning in February of 2010 in order to see the effect compared with the current system conditions. We also decided to increase the transmission capacity between the Southeast / Central Region with the North East and North regions by 1000 MW in all directions. With this instance it is possible to see if we have a lack of transmission capacity in the base case.

5.2.4 Instance D - Increase System Demand

For instance D we decide to increase the electricity demand for the regions in the South and in the North East. The demand used in the input files

is a forecast for the whole horizon, updated at each month during the new simulations. The forecast values used for the year 2011 for the two mentioned regions was around 2% below what happened during that month. So in order to create this instance we decided to increase by 2% the demand of both regions from February of 2010 up to the last month that we analyze, which is January 2013. With this instance it is possible to analyze how the total cost to operate the system will behave considering the current generation capacity under increased demand.

5.3 Case Study Results

We present the results obtained for some output parameters of our hydro-thermal scheduling model for each instance. We also compare the expected total cost resulting from the different runs of PGP.

5.3.1 Energy Storage

One of the most important decision variables is the energy storage that is carried forward to the next stage. Table 5.5 presents the energy storage decisions for the first stage for each instance. Note that the storage amount in Instance E is the smallest of all because the reservoirs levels start from 10% and in the other instances the initial levels are 50%.

Table 5.5 also presents the energy storage for the whole system for Instances A through E. Note that the total energy storage for the system has a slight difference, around 200 MW larger, for Instance B in comparison with

| Region / Instance | E | A | B | C | D |
|--------------------------|----------|----------|----------|----------|----------|
| 01 | 49328.6 | 108473.7 | 105180.8 | 109287.5 | 108337.8 |
| 02 | 2824.7 | 9973.4 | 9222.2 | 10734.2 | 9909.3 |
| 03 | 1876.7 | 16046.1 | 20292.6 | 14471.5 | 15846.1 |
| 04 | 6640.1 | 8731.2 | 8731.2 | 8731.2 | 8731.2 |
| Total | 60670.1 | 143224.4 | 143426.9 | 143224.4 | 142824.4 |

Table 5.5: Hydro Energy Storage Decisions [MW-month]

A and C. Note that this difference is because of the new run-of-river capacity that Instance B has, which allows the aggregate reservoirs to store more energy. Instance D has higher demand, so it needs to produce more energy to supply its demand than Instances A, B and C which implies in smaller storage, about 400 MW, at the first stage. Instance E that is about 82500 MW smaller than the others, because of its low initial storage.

5.3.2 Hydro and Thermal Generation

We present the controllable hydro generation for each region and each instance in Table 5.6. Note that the controllable hydro generation is the amount of energy that we decide to produce from our energy storage, so the behavior of these values is the opposite of those presented for the energy storage.

| Region / Instance | E | A | B | C | D |
|--------------------------|----------|----------|----------|----------|----------|
| 01 | 5935.5 | 5237.5 | 8530.4 | 4423.7 | 5373.4 |
| 02 | 8868.2 | 9027.1 | 9778.2 | 8266.3 | 9091.2 |
| 03 | 3766.7 | 4329.0 | 82.4 | 5903.6 | 4529.0 |
| 04 | 3851.7 | 5805.2 | 5805.2 | 5805.2 | 5805.2 |
| Total | 22422.0 | 24398.7 | 24196.2 | 24398.7 | 24798.7 |

Table 5.6: Controllable Hydro Generation Decisions [MW-month]

Table 5.6 also presents the controllable hydro generation for the whole system for all instances. Note that the total controllable hydro generation for the system has a slight difference, around 200 MW smaller, for Instance B in comparison with A and C. This difference is because of the new run-of-river capacity that Instance B has, which allows the system to use less controllable energy. Instance D has higher demand, so it needs to produce more energy to supply its demand than Instances A, B and C which implies in smaller storage, about 400 MW, at the first stage. Instance E produces about 2000 MW less than Instances A and C because of its low initial storage, and also because it prefers to store hydro energy for future use. This difference in the controllable hydro generation from Instance E in comparison with the other instances is supplied by thermal generation, which makes the present cost for Instance E larger than for the others. Table 5.7 presents the thermal generation on the first stage for the whole system for each instance. Note that the thermal generation for instances A through D is very small, around 40 MW, and the thermal generation for Instance E is much larger, around 2700 MW.

| Instance | E | A | B | C | D |
|-----------------|----------|----------|----------|----------|----------|
| Total | 2711.1 | 40.8 | 40.8 | 40.8 | 40.8 |

Table 5.7: Total Thermal Generation Decision for the System [MW-month]

5.3.3 Power Exchanges

We present the results of the power exchanges between regions for Instance A, the base case, and Instance C, where we have additional transmission capac-

ity. Table 5.8 presents the power exchanges between regions. The row entry represents represents the total amount that goes in or out of that Region. If the amount is positive it means that power is flowing into that Region, and if the amount is negative power is flowing out of that Region. Note that for the Virtual Region we have a “±” symbol, which represents that the same amount in going in and out of the Virtual Region. Note that the amounts are different for Regions 01 through 03 for the two instances.

| Instance / Region | 01 | 02 | 03 | 04 | Virtual |
|-------------------|---------|----------|----------|----------|-----------|
| A | 6453.52 | -2807.52 | -491.97 | -3154.03 | ± 3203.04 |
| C | 7267.35 | -2046.72 | -2066.60 | -3154.03 | ± 4792.64 |

Table 5.8: Power Exchanges Between Regions for Instances A and C [MW-month]

5.3.4 Total Cost

Figure 5.8 presents the total expected cost for instances A through D. We carried out a paired Student- t test in order to assess whether the apparent decrease or increase in cost with different instances is statistically significant. Table 5.9 shows the results of this test. A positive difference for the point estimate indicates a larger cost. So we can see that the expected cost for Instance E is the largest one, because we start only with 10% of the reservoir volumes. The cost of Instance A is around three times larger than the cost of Instance B, which is expected because of the addition of new hydro generation capacity in Instance B. The cost of Instance A is slightly larger than the cost in Instance C, meaning that the addition of new transmission capacity did not

have much influence in the expected cost in this case. The cost of Instance A is around 1.5 times smaller than the cost of the higher demand of Instance D, which means that with the current generation installed if the demand has the modifications described earlier the system operational costs are expected to be higher.

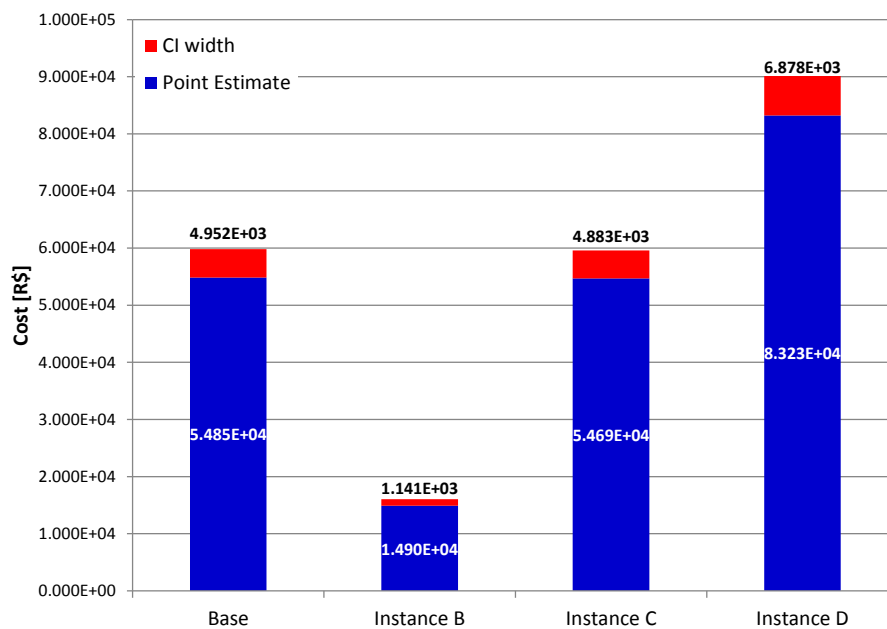


Figure 5.8: Upper Bound Point Estimate and CI Width for Instances

de Carvalho (2008) presents an analysis of the cost per MWh to build the hydro generators of Santo Antonio and Jirau that are located in the same river basin. The total power of both generators is 6450 MW where 3300 MW are from the Jirau hydro plant. de Carvalho considers an amortization period of 30 years with internal rate of return of 10% per year. The considered investment cost for both hydro generators is roughly around R\$ 17.5 billion and

| Instance | | E | A | B | C |
|----------|----|-----------|----------|----------|----------|
| A | Pt | 4065888.9 | | | |
| | HW | 60031.6 | | | |
| B | Pt | 4105836.8 | 39948.0 | | |
| | HW | 61059.6 | 4384.6 | | |
| C | Pt | 4066048.5 | 159.6 | -39788.4 | |
| | HW | 59993.2 | 1436.1 | 4324.0 | |
| D | Pt | 4037508.5 | -28380.4 | -68328.4 | -28540.0 |
| | HW | 59564.5 | 2592.8 | 6388.9 | 2899.8 |

Table 5.9: Paired Student-t Test for Expected Cost of Different Instances
Paired Student- t tests using common random numbers and a 90% level with a sample size of 12800. The table contains confidence intervals for the column entry less the row entry; e.g., the first entry is 4065888.6 ± 60031.6 is a confidence interval for $U_{n_u}(E) - U_{n_u}(A)$, where $U_{n_u}(E)$ and $U_{n_u}(A)$ denote the point estimates from Instances E and A respectively. Note that the test is not an all pairwise test. Rather the results hold individually for each pair.

the estimated generation cost is 77 R\$/MWh (de Carvalho, 2008). Seventy percent of the total generation production from Jirau hydro plant is meant for long-term contracts with 30 years (EPE, 2008). According to EPE (2008), the electricity price for long term contracts for the Jirau hydro generator was settled at 71.37 R\$/MWh. As mentioned before, we consider a two year horizon in our problem instance and the generation amount provided by Jirau hydro plant is 2025 MW per month. In order to compute the annualized cost (to represent the investment) we multiply the long term electricity price by the generation amount required and the number of days in a month and the number of hours in a day to obtain the monthly cost for the hydro generator, then we discount with a discount rate of 12%. So we have:

$$M^G = 71.37 \times 2025 \times 30 \times 24 = R\$ 104.06 \text{ Million}$$

$$T^G = R\$ 2.245 \text{ Billion.}$$

Here, M^G represents the monthly cost for the new generation and T^G represents the total cost for two years for the new generation (two years to match the time horizon of our optimization model). Now, if we consider the difference in total cost of Instance A and B, we notice that the difference is about R\$ 39947.9, and we have to multiply this amount by the number of days in a month and the number of hours in a day in order to get the total cost saved due to additional hydro capacity of 2025MW over the two year horizon. So we have:

$$Pt^G = 39947.9 \times 30 \times 24 = R\$ 28.76 \text{ Million.}$$

Here, Pt^G represents the difference between the point estimates of Instance B and A. Now if we compare the value of Pt^G with T^G it is clear that the investment in new hydro generation is not justified for this case. The investment cost for new hydro capacity has to be analyzed for future planning and expansion conditions of the power system, e.g., under an increase in system demand, under the possibility of shrinking hydro supply, which in combination would correspond to operating at a higher rate on the thermal cost curve. Our analysis indicates that only under such conditions would new hydro capacity become an interesting option.

The addition of new transmission capacity between Region 01 and the others, Instance C, barely changed the total operational cost (if we compare with Instance A). In order to increase these capacities by 1000 MW we consider

the construction of transmission lines at voltage level of 525 kV. It is necessary to construct about 180 km of transmission lines to connect Region 01 and 02, 950 km of lines to connect Region 01 and 04, and 900 km of lines to connect Region 01 and 03. We use the standard costs per km for transmission lines of 500 kV, 314510.0 R\$/km, from (ANEEL, 1999) in order to compute the revenue requirement per year from these transmission links. We also consider the year cost of the buses for the transmission lines from (ANEEL, 1999), R\$ 3751450.0 for each bus. Approximately, for the two years horizon we have for the revenue requirement of the new transmission capacity:

$$T^C = 2 \times (314510 \times (180 + 950 + 900) + 2 \times 3751450) = R\$ 1.292 \text{ Billion}$$

$$Pt^C = 159.6 \times 30 \times 24 = R\$ 0.12 \text{ Million.}$$

Here, T^C represents the total transmission revenue requirement for the new capacity for the two years and Pt^C represents the difference between the point estimates of Instance C and A. Now if we compare the value of Pt^C with T^C it is clear that the investment in new transmission capacity is not justified for this case. As mentioned for the addition of new hydro generation, the investment cost for new transmission capacity has to be analyzed for future planning and expansion conditions of the power system.

For Instance D, the increase in system demand represents the addition of about 400 MW per month if compared to the demand of Instance A. Let Pt^D be the difference between the point estimates of Instance E and A given by:

$$Pt^D = -28380.4 \times 30 \times 24 = -R\$ 20.43 \text{ Million.}$$

Note that the negative sign in Pt^D represents the increase in cost incurred by the additional load. Now, we can obtain the cost to serve the additional load in terms of R\$/MWh by:

$$C^D = \frac{20.43 \times 10^6}{24 \times 30 \times 400} = 70.9 \text{ R\$/MWh.}$$

Here, C^D represents the cost to serve the additional demand in terms of R\$/MWh. Note that, if we compare C^D with the thermal generation costs presented in Figure 5.7, C^D has the cost of a cheap thermal plant.

It is possible to notice from the performed analysis that many different things can be analyzed with our model. Many interesting questions arise: What is the increment in demand that will justify the addition new generation? What is the optimal point that we can obtain if we consider investment and operational costs? How well prepared is the current hydro-thermal system to handle changes in the climate, which may result in drier hydro conditions and reduced hydro supply? Given the current conditions, what is the load curtailment risk for the next stages? The developed model gives the possibility to deal with these and many other questions and also it is a valuable tool to analyze investment opportunities in generation and transmission.

Chapter 6

Conclusions and Future Work

6.1 Research Contributions

Hydro-thermal scheduling is a challenging problem that we modeled as a large-scale multi-stage stochastic program with stochastic monthly inflows. In the literature there are two main representations of such a problem. One representation of the hydro-thermal scheduling problem is with the modeling resolution of individual hydro generators and stochastic monthly inflows of water. The second approach uses an energy-based aggregate reservoir representation with stochastic monthly inflows of energy. The representation with individual hydro generators can be more precise since we can better represent the relationship of the hydro plants in each river basin. Also, this representation uses forecasting models that can exploit local predictors to forecast the stochastic water inflows. But the computational effort to solve such a model grows with the number of hydro generators, the level of representation of the system details, the number of stages and the branches in the scenario tree. On the other hand, the representation of the problem using the aggregate reservoir scheme is more appealing from the computational point of view, since the number of decision variables and constraints shrinks considerably for large systems. However applications of the aggregate reservoir representation, as the one used in

Brazil, are tied to the forecasts of energy inflows that may not represent well the behavior of the water inflows at each hydro generator.

We present in this dissertation a model for the hydro-thermal scheduling problem using the aggregated reservoir representation and water inflow forecasts instead of energy forecasts. In this way, we can use forecast models that better represent the real stochastic parameter and also exploit computational efficiency. Once the stochastic inflow forecasts are given to the model, they are transformed into energy that contributes to hydro energy generation, to energy storage for the aggregate reservoirs or even to spillage.

Our approach to solve this problem is based on the SDDP algorithm, which has been in the literature since 1991 (Pereira and Pinto, 1991). The cut-sharing methodology presented by Infanger and Morton (1996) for the SDDP is very important for practical applications of the algorithm, when inflows exhibit interstage dependency. In order to deal with the hydro-thermal scheduling problem with the aggregate reservoir representation, presented in Chapter 2, and with forecasts of water inflows at each hydro generator we had to extend previous work and create a novel approach for cut-sharing. We present in Chapter 3 this extension of the cut-sharing methodology (and the proof of its validity) for the SDDP algorithm to handle an aggregate energy-based scheme with water inflows represented by an additive interstage dependency model.

In Chapter 4, we assess the solution quality of a policy generated with a sample average approximation with respect to the true problem. We propose a procedure for forming an out-of-sample upper bound estimator for the true

problem using the collection of cuts and first stage optimal solution of the sampled average approximation problem. We also propose a procedure to create a lower bound estimator for the true problem by generating several scenario trees and running the SDDP algorithm on each of them. For the two proposed procedures we also compute the sample variance estimators. Then we construct confidence intervals for the true problem based on these estimators. We investigate how to increase various sample sizes in order to shrink the width of these intervals. From the results obtained, it is possible to notice that the confidence interval width and error reduces as the sample size grows.

In Chapter 5, we used scenario trees and further sample sizes based on our analysis in Chapter 4 on assessing solution quality. We compared a base case problem instance with three modeling excursions based on currently planned improvements to, and demands on, the hydro-thermal system in Brazil. These include expansion in hydro generation capacity, expansion in transmission capacity, and additional system load. Our analysis suggested that the additional transmission capacity provided little, if any, benefit in reducing expected system costs. Additional run-of-river hydro generation capacity had a significant effect in decreasing expected costs. The modeling scenario we considered with additional system load significantly increased expected system cost.

6.2 Final Remarks and Future Research Goals

The theory developed in Chapter 3 proposes that approaching the problem via: (a) a decomposition algorithm that uses the cut-sharing methodology to handle additive interstage dependency, and (b) a model with interstage independence but an extended state variable, i.e., additional decision variables and constraints, are mathematically equivalent. The model with the extend-state formulation does not have to compute the cuts in a complicated way because the extended state allows us to model the problem using interstage independence. On the other hand, the constraint matrix is larger due to an increased number of decision variables and constraints, compared to the formulation that makes use of the cut-sharing methodology. One future research goal is to perform a computational study in order to analyze the computational efficiency of both formulations as the problem size scales large.

The results presented in Chapter 4 shows that the solution quality assessment with respect to the true problem is possible for a multi-stage stochastic program. Our results become better and better as the sample size grows, but also the computational effort required to solve such problems grows. The question that arises is how large should the sample size be because of the trade-off between solution quality and computational effort. In our case, we choose during the case study to use the number of branches $n(1) = 200$ scenarios for the first stage and $n(t)$ defined as in equation (4.3) for $t = 2, \dots, T-1$, because the results of Chapter 4 show that besides the gain in quality of larger samples the computational effort increases considerably. We believe that a promis-

ing research direction would be to assess the solution quality in multi-stage stochastic programs using smart sampling ideas to better select the scenarios to create the sampled scenario trees. If such sampling schemes are embedded in the creation of the scenario tree, we may be able to solve instances and still obtain reasonable results, avoiding unnecessary computational effort.

We solve several instances of the hydro-thermal scheduling problem considering 24-monthly stages. One interesting idea for future work is the assessment of the policy quality as the time horizon grows. It would be possible to analyze how the estimators from PGP and LBE procedures and the optimality gap behave as the number of stages increase.

All of our analysis has been developed considering a risk neutral approach. Another research direction is to employ risk measures such as the conditional value at risk (CVaR) within the SDDP algorithm, as developed in Shapiro (2011). Assessing solution quality in such a setting would require extension of the current techniques.

Appendices

Appendix A

Solution Quality Instances

A.1 Instance 01 - HTSPM: 24STG-10SCN-A

We start our analysis with an instance of the 24-stage problem considering $n(t) = 10$ scenarios for all stages (24STG-10SCN-A). We use the full sampled tree with a total of 10^{23} nodes to generate cuts. The cuts are then used to compute the upper bound estimator during the policy generation procedure (PGP). We consider 128 forward paths in the first tree to estimate the local upper bound (local, in that it applies only within the SDDP algorithm) at each iteration and we use 32 of these paths to compute the backward step of the SDDP algorithm, where we generate the cuts. The lower bound for the sample average approximation is given by the optimal value, $c_1x_1^k + \theta_1^k$, of the first stage problem when the SDDP algorithm stops at iteration k .

At each iteration of the algorithm new forward paths are chosen using Monte Carlo sampling. We run the algorithm for a certain number of iterations until the lower bound obtained by the sampled scenario tree does not show considerable improvement and its value is close enough to the upper bound.

Figure A.1 presents the convergence process of the SDDP algorithm during 45 iterations. Table B.1 in Appendix B, presents the results of the

lower bound, upper bound, confidence interval halfwidth for the upper bound, and the computation time for each iteration for this problem instance. The problem was solved using the supercomputer Ranger from Texas Advanced Computing Center (TACC) using 128 cores. The algorithm parallelization was developed using message passing interface (MPI). While a detailed analysis of the effectiveness of the parallelization is not one of our research goals, in Chapter 3 we describe further the algorithm parallelization that was necessary to solve such large instances.

Once the first instance is solved we use the optimal solution of the first stage and the collection of cuts obtained for each stage to determine the upper bound estimator as described in Section 4.1. As mentioned before a new collection of forward paths is sampled from the distribution of the inflows. Note that this collection of paths is independent of the first tree, because now we are interested in analyzing the out-of-sample performance. We sampled $n_u = 12800$ forward paths and computed the estimator U_{n_u} and the variance S_u^2 described in the PGP procedure. The estimated cost, sample standard deviation and confidence interval width obtained for PGP are presented in Table A.1. Note that here we consider $\alpha = 0.05$ so $z_\alpha = 1.645$, so we can compute $\epsilon_u = z_\alpha \cdot S_u / \sqrt{n_u}$.

Once the PGP estimators have been formed, we form the lower bound estimator, L_{n_ℓ} . We use the procedure described for LBE, using a total of $n_\ell = 15$ independent scenario trees with branch size $n'(t) = 8$ for $t = 1, \dots, T-1$. We have now 15 trees with 8 scenarios per stage and we run the SDDP

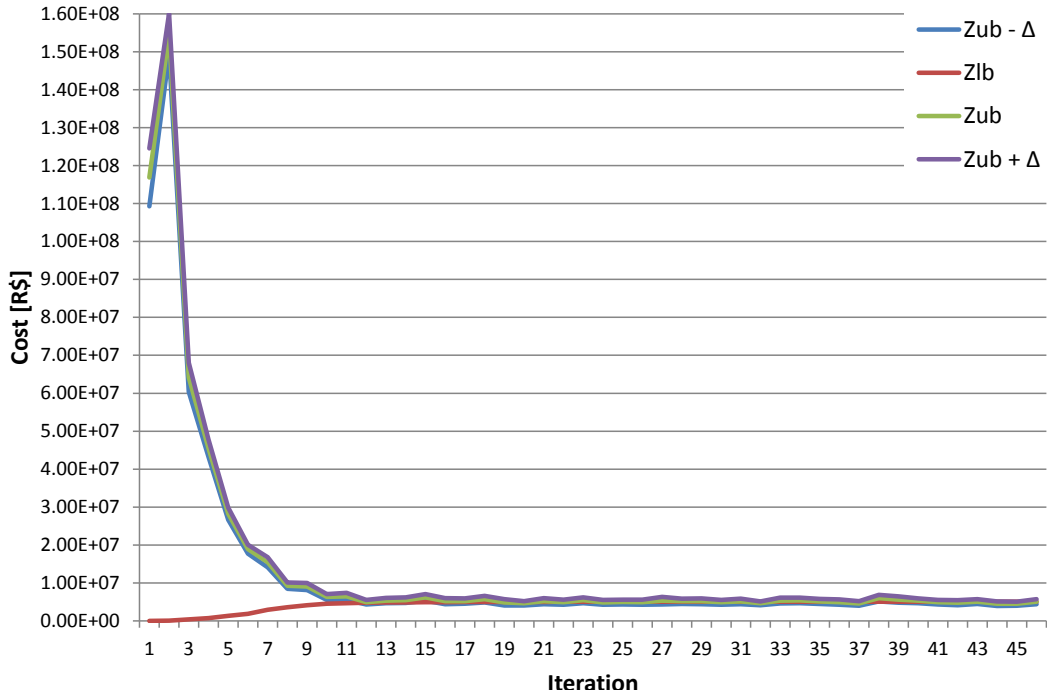


Figure A.1: SDDP Convergence for Instance 01

Referring to the SDDP algorithm presented in Section 3.6, Zlb is the lower bound \underline{z}^k at iteration k , Zub is the sample mean upper bound \bar{z}^k at iteration k , and Δ is a 95% confidence interval halfwidth associated with upper bound estimator.

algorithm on each of those trees to estimate L_{n_ℓ} and S_ℓ^2 . Figure A.2 shows the different values of the lower bound for each one of the 15 trees. We run the SDDP algorithm for 20 iterations for each tree. The values obtained for L_{n_ℓ} , S_ℓ and ϵ_ℓ are presented in Table A.2.

The gap between U_{n_u} and L_{n_ℓ} and the error associated with this gap is presented in Table A.3. The $(1 - \alpha)$ -level confidence interval on the optimality gap $\mathbb{E}U - z^*$ and its associated error with respect to U_{n_u} for this instance are

| $U_{n_u}[R\$]$ | $S_u[R\$]$ | $\epsilon_u[R\$]$ | $\epsilon_u\%$ |
|----------------|------------|-------------------|----------------|
| 4755388.8 | 4294897.6 | 62257.5 | 1.3% |

Table A.1: Output of PGP for Instance 01
 To form the cuts we use $n(t) = 10$ for all stages and run SDDP for 45 iterations. To estimate the upper bound we use $n_u = 12800$.

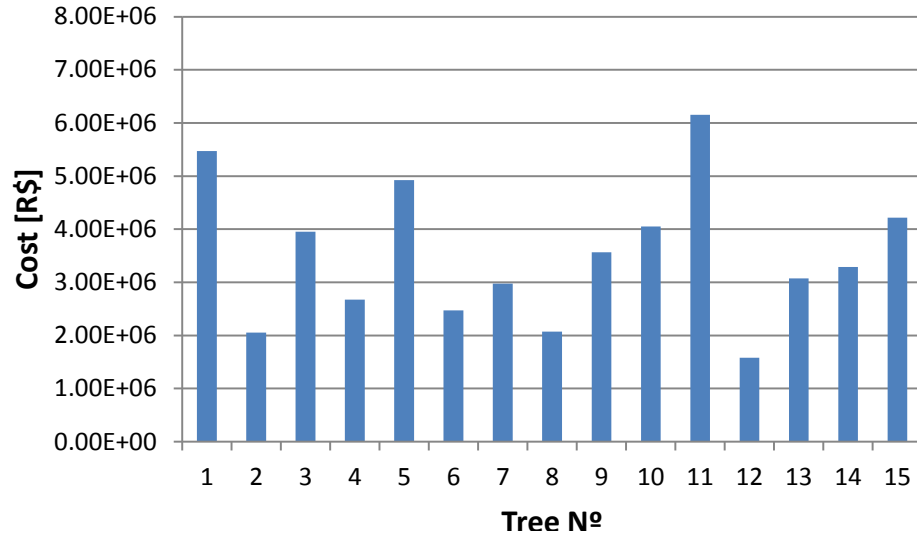


Figure A.2: Lower Bound for Each Tree on Instance 01

presented in Table A.4.

The error associated with the CI for this instance is around 40%, and it can be explained by the large variability of the water inflows. We increase the sample size for the next instances in order to reduce the CI error. We present next other instances with larger sample sizes.

| $L_{n_\ell}[R\$]$ | $S_\ell[R\$]$ | $\epsilon_\ell[R\$]$ | $\epsilon_\ell\%$ |
|-------------------|---------------|----------------------|-------------------|
| 3500672.5 | 1310520.0 | 595878.0 | 12.5% |

Table A.2: Output of LBE for Instance 01

| $gap[R\$]$ | $gap\%$ |
|------------|---------|
| 1254716.3 | 26.4% |

Table A.3: Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 01

A.2 Instance 02 - HTSPM: 24STG-20SCN-A

For the second instance we consider a 24-stage problem with $n(t) = 20$ for all stages (24STG-20SCN-A). As in Section A.1 we use the full sampled tree, now with 20^{23} nodes, to generate cuts. The cuts are then used to compute the upper bound estimator during the PGP. As before, we consider 128 forward paths in this tree to estimate the local upper bound at each iteration and we use 32 of these paths to compute the backward step of the SDDP algorithm.

Figure A.3 presents the convergence process of the SDDP algorithm during 45 iterations. Table B.2 in Appendix B, presents the results of the lower bound, upper bound, confidence interval halfwidth for the upper bound and the computation time for each iteration for this problem instance. The problem was solved again solved using 128 cores.

As in Section A.1, once the instance is solved we use the optimal solution of the first stage and the collection of cuts obtained for each stage to determine the upper bound estimator with a new collection of forward

| | |
|----------------------------|-------|
| CI for $\mathbb{E}U - z^*$ | CI% |
| [0, 1912851.8] | 40.2% |

Table A.4: CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 01

paths sampled from the distribution of the inflows. Again for this instance we sampled $n_u = 12800$ forward paths and computed the estimator U_{n_u} and the variance S_u^2 described in the PGP procedure. The estimated cost, sample standard deviation and confidence interval width obtained for PGP are presented in Table A.5.

| | | | |
|----------------|------------|-------------------|----------------|
| $U_{n_u}[R\$]$ | $S_u[R\$]$ | $\epsilon_u[R\$]$ | $\epsilon_u\%$ |
| 4182477.9 | 4394349.1 | 63699.1 | 1.5% |

Table A.5: Output of PGP for Instance 02

Once the PGP estimators for this problem instance have been formed, we estimate the lower bound L_{n_ℓ} using the procedure LBE. We selected a total of $n_\ell = 15$ different scenario trees with branch size $n'(t) = 15$ for $t = 1, 2, \dots, T - 1$. We now have 15 trees with 15 scenarios per stage and we run the SDDP algorithm on each of those trees to form L_{n_ℓ} and S_ℓ^2 . Figure A.4 shows the different values of the lower bound for each one of the 15 trees used. We run the SDDP algorithm for 20 iterations for each tree. The values obtained for L_{n_ℓ} , S_ℓ and ϵ_ℓ are presented in Table A.6.

The gap between U_{n_u} and L_{n_ℓ} and the error associated to this gap is presented in Table A.7. The $(1 - \alpha)$ -level confidence interval on the optimality

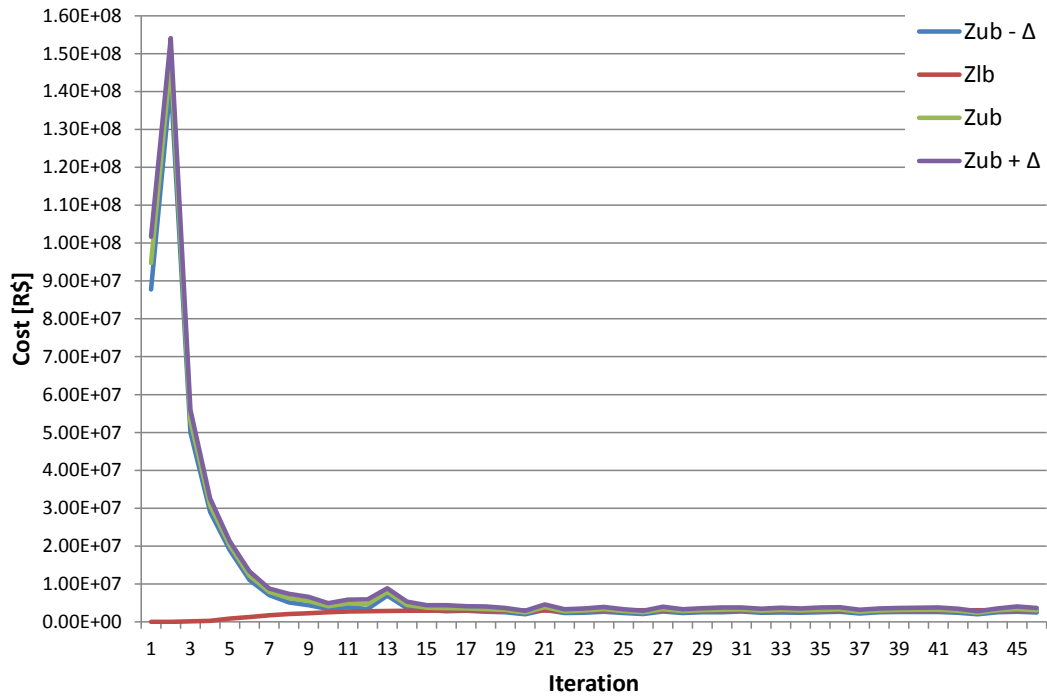


Figure A.3: SDDP Convergence for Instance 02

The terms Zlb, Zub and Δ have the same interpretation as in Figure A.1

gap $\mathbb{E}U - z^*$ and its associated error with respect to U_{n_u} for this instance are presented in Table A.8. The error associated with the CI for this instance is around 16.2%.

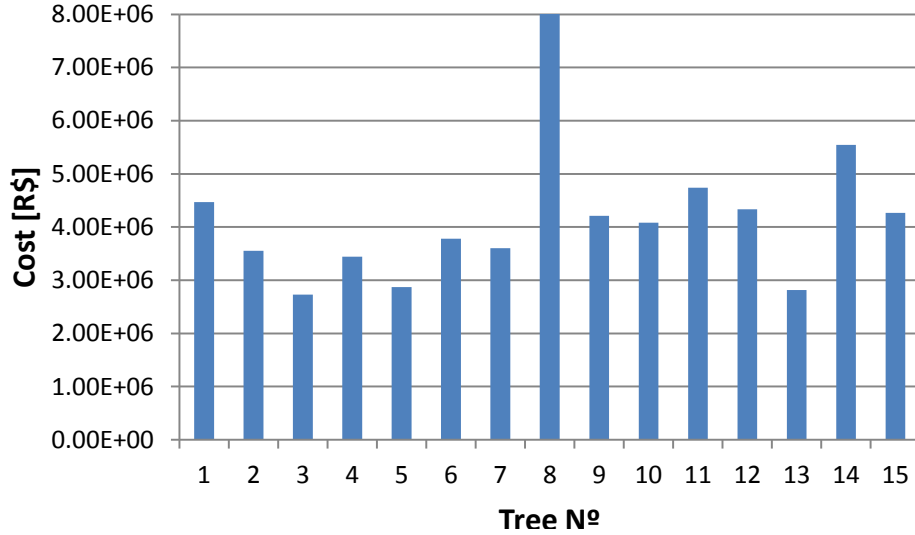


Figure A.4: Lower Bound for Each Tree on Instance 02

| $L_{n_\ell}[R\$]$ | $S_\ell[R\$]$ | $\epsilon_\ell[R\$]$ | $\epsilon_\ell\%$ |
|-------------------|---------------|----------------------|-------------------|
| 4166823.0 | 1318778.4 | 599633.0 | 14.3% |

Table A.6: Output of LBE for Instance 02

A.3 Instance 03 - HTSPM: 24STG-60SCN-F-*SCN-O

For the third instance we consider a 24-stage problem with $n(1) = 60$ and a different number of scenarios for the other stages (24STG-60SCN-F-*SCN-O), where $n(t)$ is defined by (4.3) for $t = 2, \dots, T - 1$. We use (4.3) to reduce the number of scenarios as the stage index t grows in order to reduce computational effort. We choose $\rho = 0.8$ and $n_{min} = 20$ in (4.3). As before we consider 128 forward paths in this tree to estimate the local upper bound at each iteration and we use 32 of these paths to compute the backward step of the SDDP

| $gap[R\$]$ | $gap\%$ |
|------------|---------|
| 15654.9 | 0.4% |

Table A.7: Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 02

| CI for $\mathbb{E}U - z^*$ | CI% |
|----------------------------|-------|
| [0, 678987.1] | 16.2% |

Table A.8: CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 02

algorithm.

Figure A.5 presents the convergence process of the SDDP algorithm during 45 iterations for this instance. Table B.3 in Appendix B, presents the results of the lower bound, upper bound, confidence interval halfwidth for the upper bound and the computation time for each iteration for this problem instance. The problem was solved again using 128 cores.

The same procedure was used to determine the upper bound estimator with PGP with an independent collection of forward paths. Again for this instance we sampled $n_u = 12800$ forward paths and computed the estimator U_{n_u} and the variance S_u^2 . The estimated cost, sample standard deviation and confidence interval width obtained for PGP are presented in Table A.9.

| $U_{n_u}[R\$]$ | $S_u[R\$]$ | $\epsilon_u[R\$]$ | $\epsilon_u\%$ |
|----------------|------------|-------------------|----------------|
| 4256700.1 | 4027352.1 | 58379.2 | 1.4% |

Table A.9: Output of PGP for Instance 03

Once the PGP estimators for this problem instance have been formed,

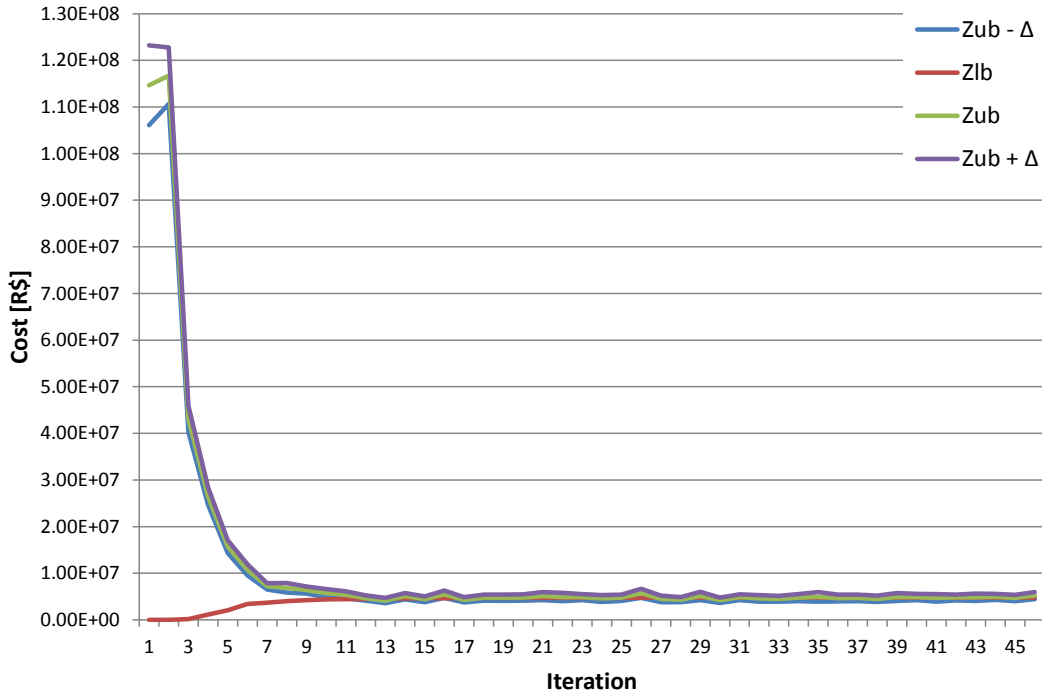


Figure A.5: SDDP Convergence for Instance 03

The terms Zlb, Zub and Δ have the same interpretation as in Figure A.1

we estimate the lower bound L_{n_ℓ} using the LBE procedure. We selected a total of $n_\ell = 15$ different scenario trees with branch size $n'(t) = 45$ for $t = 1$ and $n'(t)$ defined by (4.3) for $t = 2, \dots, T - 1$. We now have 15 trees with 45 scenarios on the first stage and a different number of scenarios defined using (4.3), with $n(1)$ replaced by $n'(1)$, for the other stages. We run the SDDP algorithm on each of those trees for 20 iterations to form L_{n_ℓ} and S_ℓ^2 . Figure A.6 shows the different values of the lower bound for each one of the 15 trees used. The values obtained for L_{n_ℓ} , S_ℓ and ϵ_ℓ are presented in Table A.10.

The gap between U_{n_u} and L_{n_ℓ} and the error associated to this gap is

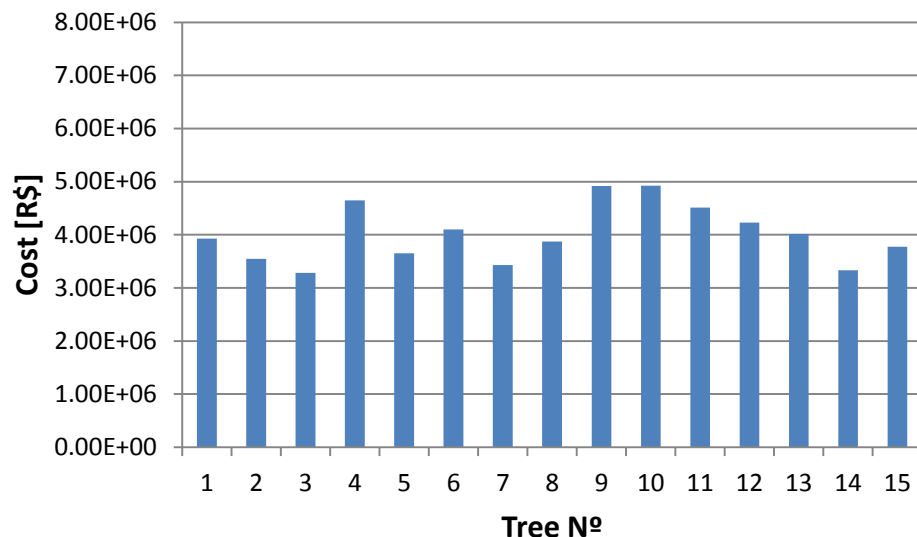


Figure A.6: Lower Bound for Each Tree on Instance 03

| $L_{n_\ell}[R\$]$ | $S_\ell[R\$]$ | $\epsilon_\ell[R\$]$ | $\epsilon_\ell\%$ |
|-------------------|---------------|----------------------|-------------------|
| 4011643.7 | 542428.7 | 246636.0 | 5.8% |

Table A.10: Output of LBE for Instance 03

presented in Table A.11. The $(1-\alpha)$ -level confidence interval on the optimality gap $\mathbb{E}U - z^*$ and its associated error with respect to U_{n_u} for this instance are presented in Table A.12. The error associated with the CI for this instance is around 12.9%.

| $gap[R\$]$ | $gap\%$ |
|------------|---------|
| 245056.4 | 5.8% |

Table A.11: Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 03

Note that the improvement in the percentage error if we compare with

| CI for $\mathbb{E}U - z^*$ | CI % |
|----------------------------|-------|
| [0, 550071.6] | 12.9% |

Table A.12: CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 03

Instance 02 was about 3%. In order to reduce even more the error associated with the CI we further increase even more the sample size for the other instances.

A.4 Instance 04 - HTSPM: 24STG-100SCN-F-*SCN-O

For the fourth instance we consider a 24-stage problem with $n(1) = 100$ and a different number of scenarios for the other stages (24STG-100SCN-F-*SCN-O), where $n(t)$ is defined by (4.3) for $t = 2, \dots, T - 1$. We choose $\rho = 0.8$ and $n_{min} = 20$ in (4.3). As before we consider 128 forward paths in this tree to estimate the local upper bound at each iteration and we use 32 of these paths to compute the backward step of the SDDP algorithm.

Figure A.7 presents the convergence process of the SDDP algorithm during 45 iterations for this instance. Table B.4 in Appendix B, presents the results of the lower bound, upper bound, confidence interval halfwidth for the upper bound and the computation time for each iteration for this problem instance. The problem was solved again using 128 cores.

The same procedure was used to determine the upper bound estimator with PGP with an independent collection of forward paths. Again for this instance we sampled $n_u = 12800$ forward paths and computed the estimator U_{n_u} and the variance S_u^2 . The estimated cost, sample standard deviation and confidence interval width obtained for PGP are presented in Table A.13.

| $U_{n_u}[R\$]$ | $S_u[R\$]$ | $\epsilon_u[R\$]$ | $\epsilon_u\%$ |
|----------------|------------|-------------------|----------------|
| 4171942.8 | 4368341.8 | 63322.1 | 1.5% |

Table A.13: Output of PGP for Instance 04

Once the PGP estimators for this problem instance have been formed, we estimate the lower bound L_{n_ℓ} using the LBE procedure. We selected a

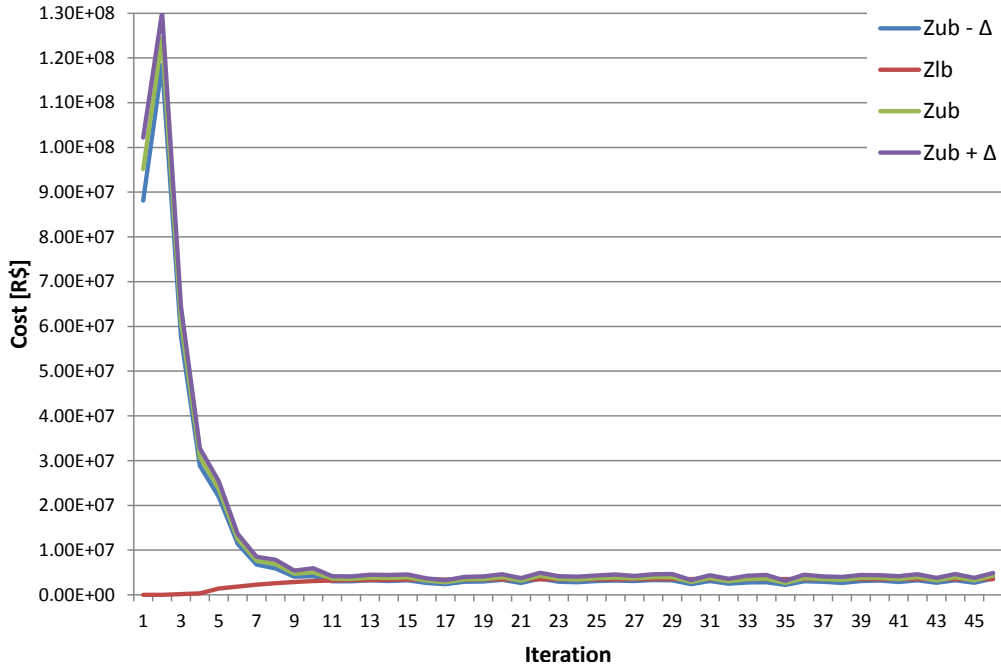


Figure A.7: SDDP Convergence for Instance 04

The terms Zlb, Zub and Δ have the same interpretation as in Figure A.1

total of $n_\ell = 15$ different scenario trees with branch size $n'(t) = 75$ for $t = 1$ and $n'(t)$ defined by (4.3) for $t = 2, \dots, T - 1$. We now have 15 trees with 75 scenarios on the first stage and a different number of scenarios defined using (4.3), with $n(1)$ replaced by $n'(1)$, for the other stages. We run the SDDP algorithm on each of those trees for 20 iterations to form L_{n_ℓ} and S_ℓ^2 . Figure A.8 shows the different values of the lower bound for each one of the 15 trees used. The values obtained for L_{n_ℓ} , S_ℓ and ϵ_ℓ are presented in Table A.14.

The gap between U_{n_u} and L_{n_ℓ} and the error associated to this gap is presented in Table A.15. The $(1 - \alpha)$ -level confidence interval on the optimality

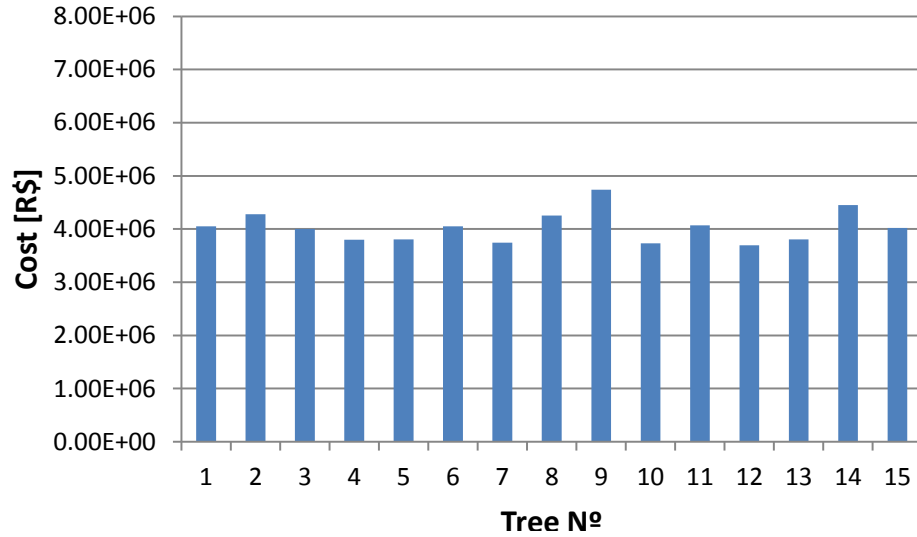


Figure A.8: Lower Bound for Each Tree on Instance 04

| $L_{n_\ell}[R\$]$ | $S_\ell[R\$]$ | $\epsilon_\ell[R\$]$ | $\epsilon_\ell\%$ |
|-------------------|---------------|----------------------|-------------------|
| 4032745.6 | 297365.2 | 135208.5 | 3.2% |

Table A.14: Output of LBE for Instance 04

gap $\mathbb{E}U - z^*$ and its associated error with respect to U_{n_u} for this instance are presented in Table A.16. In this case, the error associated with the CI is around 8%.

| $gap[R\$]$ | $gap\%$ |
|------------|---------|
| 139197.1 | 3.3% |

Table A.15: Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 04

| CI for $\mathbb{E}U - z^*$ | CI% |
|----------------------------|------|
| [0, 337727.74] | 8.1% |

Table A.16: CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 04

A.5 Instance 05 - HTSPM: 24STG-200SCN-F-*SCN-O

For the fifth instance we consider a 24-stage problem with $n(1) = 200$ and a different number of scenarios for the other stages (24STG-200SCN-F-*SCN-O), where $n(t)$ is defined by (4.3) for $t = 2, \dots, T - 1$. We choose $\rho = 0.8$ and $n_{min} = 20$ in (4.3). As before we consider 128 forward paths in this tree to estimate the local upper bound at each iteration and we use 32 of these paths to compute the backward step of the SDDP algorithm.

Figure A.9 presents the convergence process of the SDDP algorithm during 45 iterations for this instance. Table B.5 in Appendix B, presents the results of the lower bound, upper bound, confidence interval halfwidth for the upper bound and the computation time for each iteration for this problem instance. The problem was solved again using 128 cores.

The same procedure was used to determine the upper bound estimator with PGP with an independent collection of forward paths. Again for this instance we sampled $n_u = 12800$ forward paths and computed the estimator U_{n_u} and the variance S_u^2 . The estimated cost, sample standard deviation and confidence interval width obtained for PGP are presented in Table A.17.

Once the PGP estimators for this problem instance have been formed, we estimate the lower bound L_{n_ℓ} using the LBE procedure. We selected a

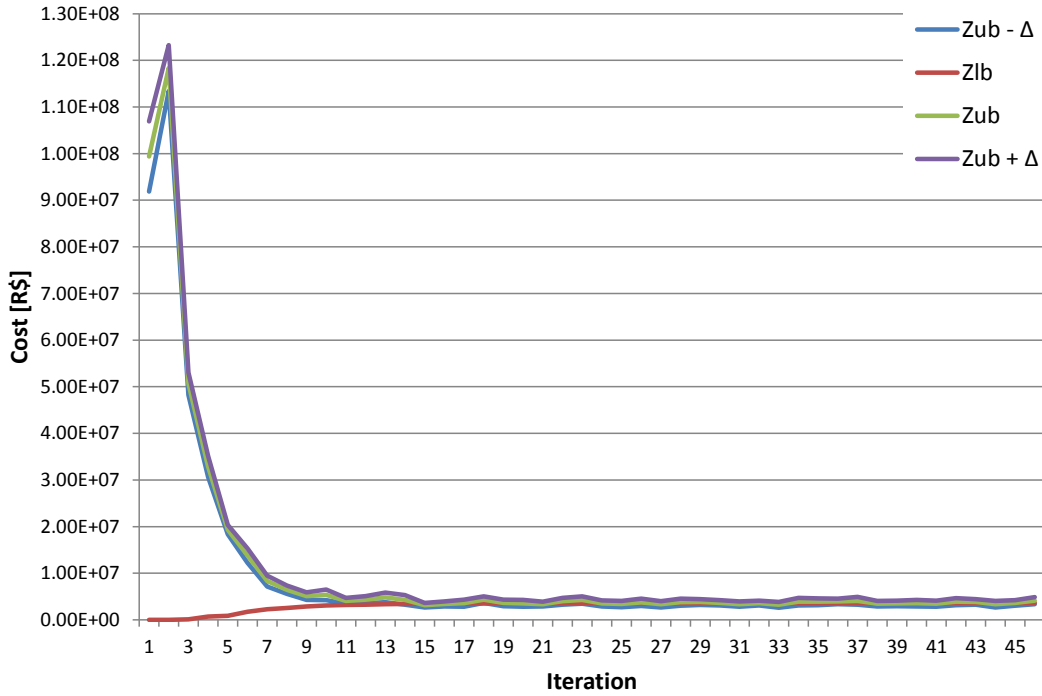


Figure A.9: SDDP Convergence for Instance 05

The terms Zlb, Zub and Δ have the same interpretation as in Figure A.1

total of $n_\ell = 15$ different scenario trees with branch size $n'(t) = 150$ for $t = 1$ and $n'(t)$ defined by (4.3) for $t = 2, \dots, T - 1$. We now have 15 trees with 150 scenarios on the first stage and a different number of scenarios defined using (4.3), with $n(1)$ replaced by $n'(1)$, for the other stages. We run the SDDP algorithm on each of those trees for 20 iterations to form L_{n_ℓ} and S_ℓ^2 . Figure A.10 shows the different values of the lower bound for each one of the 15 trees used. The values obtained for L_{n_ℓ} , S_ℓ and ϵ_ℓ are presented in Table A.18.

The gap between U_{n_u} and L_{n_ℓ} and the error associated to this gap is presented in Table A.19. The $(1 - \alpha)$ -level confidence interval on the optimality

| $U_{n_u}[R\$]$ | $S_u[R\$]$ | $\epsilon_u[R\$]$ | $\epsilon_u\%$ |
|----------------|------------|-------------------|----------------|
| 4120735.7 | 4219702.7 | 61167.5 | 1.5% |

Table A.17: Output of PGP for Instance 05

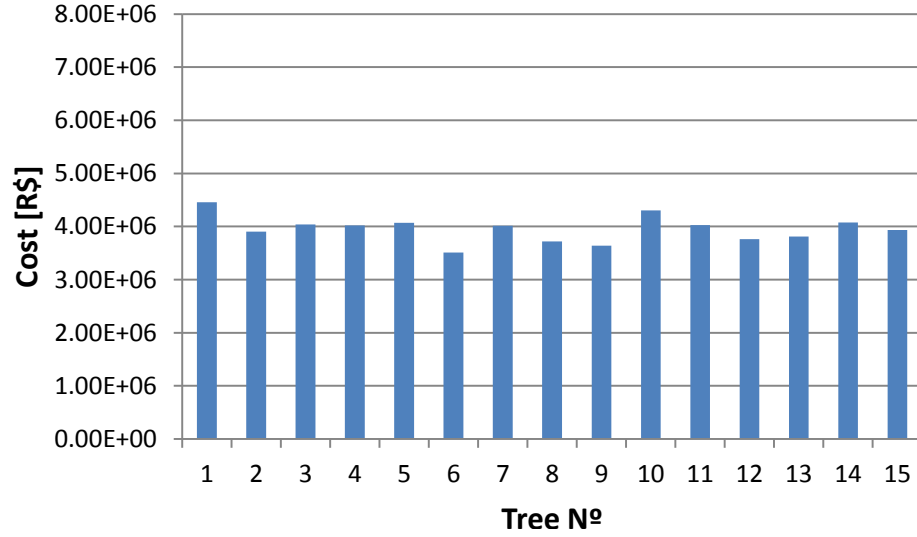


Figure A.10: Lower Bound for Each Tree on Instance 05

gap $\mathbb{E}U - z^*$ and its associated error with respect to U_{n_u} for this instance are presented in Table A.20. The error associated with the CI for this instance is around 8.2%.

| $L_{n_\ell}[R\$]$ | $S_\ell[R\$]$ | $\epsilon_\ell[R\$]$ | $\epsilon_\ell\%$ |
|-------------------|---------------|----------------------|-------------------|
| 3952628.1 | 244042.6 | 110963.3 | 2.7% |

Table A.18: Output of LBE for Instance 05

| $gap[R\$]$ | $gap\%$ |
|------------|---------|
| 168107.6 | 4.1% |

Table A.19: Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 05

| CI for $\mathbb{E}U - z^*$ | CI% |
|----------------------------|------|
| [0, 340238.4] | 8.3% |

Table A.20: CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 05

A.6 Instance 06 - HTSPM: 24STG-1000SCN-F-*SCN-O

For the sixth instance we consider a 24-stage problem with $n(1) = 1000$ and a different number of scenarios for the other stages (24STG-1000SCN-F-*SCN-O), where $n(t)$ is defined by (4.3) for $t = 2, \dots, T - 1$. We choose $\rho = 0.8$ and $n_{min} = 20$ in (4.3). As before we consider 128 forward paths in this tree to estimate the local upper bound at each iteration and we use 32 of these paths to compute the backward step of the SDDP algorithm.

Figure A.11 presents the convergence process of the SDDP algorithm during 45 iterations for this instance. Table B.6 in Appendix B, presents the results of the lower bound, upper bound, confidence interval halfwidth for the upper bound and the computation time for each iteration for this problem instance. The problem was solved again using 128 cores.

The same procedure was used to determine the upper bound estimator with PGP with an independent collection of forward paths. Again for this instance we sampled $n_u = 12800$ forward paths and computed the estimator U_{n_u} and the variance S_u^2 . The estimated cost, sample standard deviation and confidence interval width obtained for PGP are presented in Table A.21.

| $U_{n_u}[R\$]$ | $S_u[R\$]$ | $\epsilon_u[R\$]$ | $\epsilon_u\%$ |
|----------------|------------|-------------------|----------------|
| 4090672.5 | 4037218.8 | 58522.3 | 1.4% |

Table A.21: Output of PGP for Instance 06

Once the PGP estimators for this problem instance have been formed,

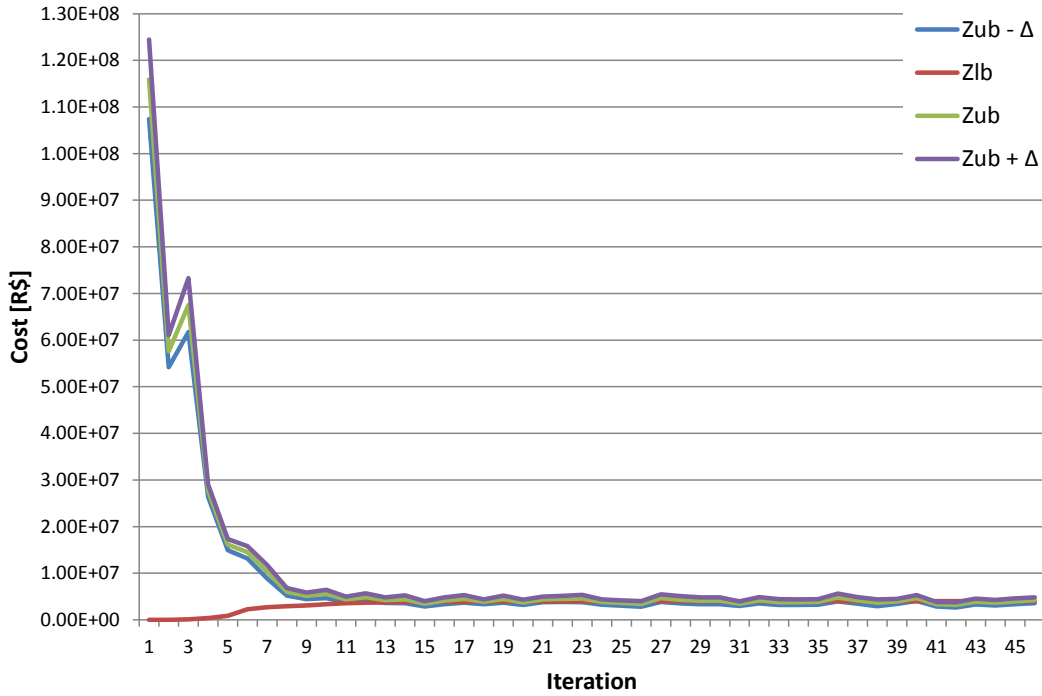


Figure A.11: SDDP Convergence for Instance 06
The terms Zlb, Zub and Δ have the same interpretation as in Figure A.1

we estimate the lower bound L_{n_ℓ} using the LBE procedure. We selected a total of $n_\ell = 15$ different scenario trees with branch size $n'(t) = 750$ for $t = 1$ and $n'(t)$ defined by (4.3) for $t = 2, \dots, T - 1$. We now have 15 trees with 750 scenarios on the first stage and a different number of scenarios defined using (4.3), with $n(1)$ replaced by $n'(1)$, for the other stages. We run the SDDP algorithm on each of those trees for 20 iterations to form L_{n_ℓ} and S_ℓ^2 . Figure A.12 shows the different values of the lower bound for each one of the 15 trees used. The values obtained for L_{n_ℓ} , S_ℓ and ϵ_ℓ are presented in Table A.22.

The gap between U_{n_u} and L_{n_ℓ} and the error associated to this gap is

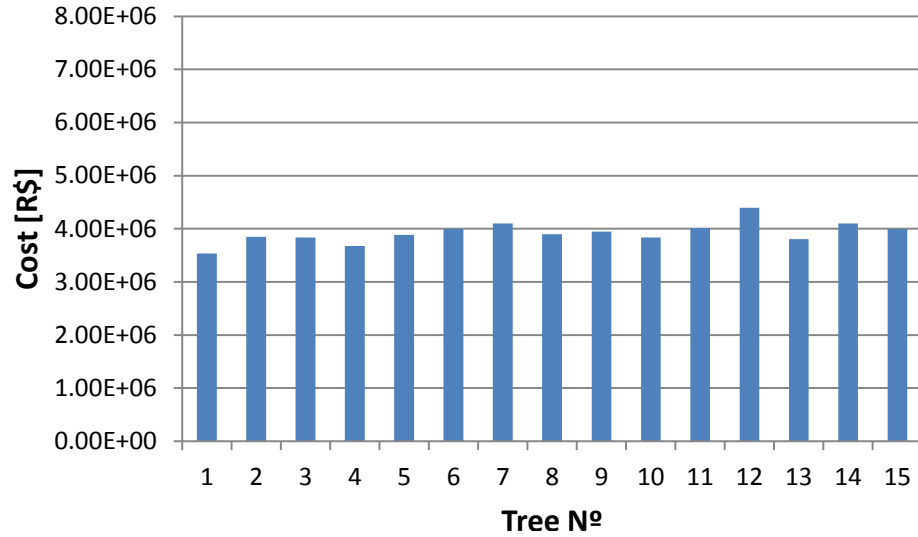


Figure A.12: Lower Bound for Each Tree on Instance 06

| $L_{n_\ell}[R\$]$ | $S_\ell[R\$]$ | $\epsilon_\ell[R\$]$ | $\epsilon_\ell\%$ |
|-------------------|---------------|----------------------|-------------------|
| 3924632.1 | 199001.4 | 90483.6 | 2.2% |

Table A.22: Output of LBE for Instance 06

presented in Table A.23. The $(1-\alpha)$ -level confidence interval on the optimality gap $\mathbb{E}U - z^*$ and its associated error with respect to U_{n_u} for this instance are presented in Table A.24. The error associated with the CI for this instance is around 7.7%.

| $gap[R\$]$ | $gap\%$ |
|------------|---------|
| 166040.4 | 4.1% |

Table A.23: Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 06

| CI for $\mathbb{E}U - z^*$ | CI% |
|----------------------------|------|
| [0, 315046.3] | 7.7% |

Table A.24: CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 06

A.7 Instance 07 - HTSPM: 24STG-2000SCN-F-*SCN-O

For the seventh instance we consider a 24-stage problem with $n(1) = 2000$ and a different number of scenarios for the other stages (24STG-2000SCN-F-*SCN-O), where $n(t)$ is defined by (4.3) for $t = 2, \dots, T - 1$. We choose $\rho = 0.8$ and $n_{min} = 20$ in (4.3). As before we consider 128 forward paths in this tree to estimate the local upper bound at each iteration and we use 32 of these paths to compute the backward step of the SDDP algorithm.

Figure A.13 presents the convergence process of the SDDP algorithm during 45 iterations for this instance. Table B.7 in Appendix B, presents the results of the lower bound, upper bound, confidence interval halfwidth for the upper bound and the computation time for each iteration for this problem instance. The problem was solved again using 128 cores.

The same procedure was used to determine the upper bound estimator with PGP with an independent collection of forward paths. Again for this instance we sampled $n_u = 12800$ forward paths and computed the estimator U_{n_u} and the variance S_u^2 . The estimated cost, sample standard deviation and confidence interval width obtained for PGP are presented in Table A.25.

Once the PGP estimators for this problem instance have been formed,

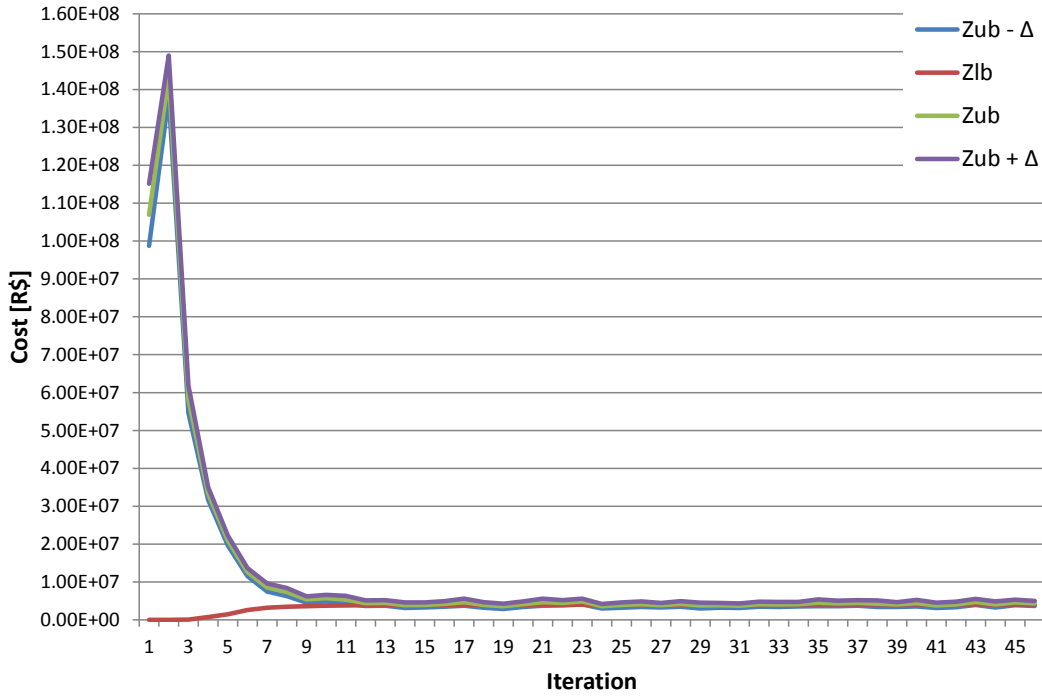


Figure A.13: SDDP Convergence for Instance 07

The terms Zlb, Zub and Δ have the same interpretation as in Figure A.1

we estimate the lower bound L_{n_ℓ} using the LBE procedure. We selected a total of $n_\ell = 15$ different scenario trees with branch size $n'(t) = 1500$ for $t = 1$ and $n'(t)$ defined by (4.3) for $t = 2, \dots, T - 1$. We now have 15 trees with 1500 scenarios on the first stage and a different number of scenarios defined using (4.3), with $n(1)$ replaced by $n'(1)$, for the other stages. We run the SDDP algorithm on each of those trees for 20 iterations to form L_{n_ℓ} and S_ℓ^2 . Figure A.14 shows the different values of the lower bound for each one of the 15 trees used. The values obtained for L_{n_ℓ} , S_ℓ and ϵ_ℓ are presented in Table A.26.

The gap between U_{n_u} and L_{n_ℓ} and the error associated to this gap is

| $U_{n_u}[R\$]$ | $S_u[R\$]$ | $\epsilon_u[R\$]$ | $\epsilon_u\%$ |
|----------------|------------|-------------------|----------------|
| 4076822.7 | 4036476.5 | 58511.5 | 1.4% |

Table A.25: Output of PGP for Instance 07

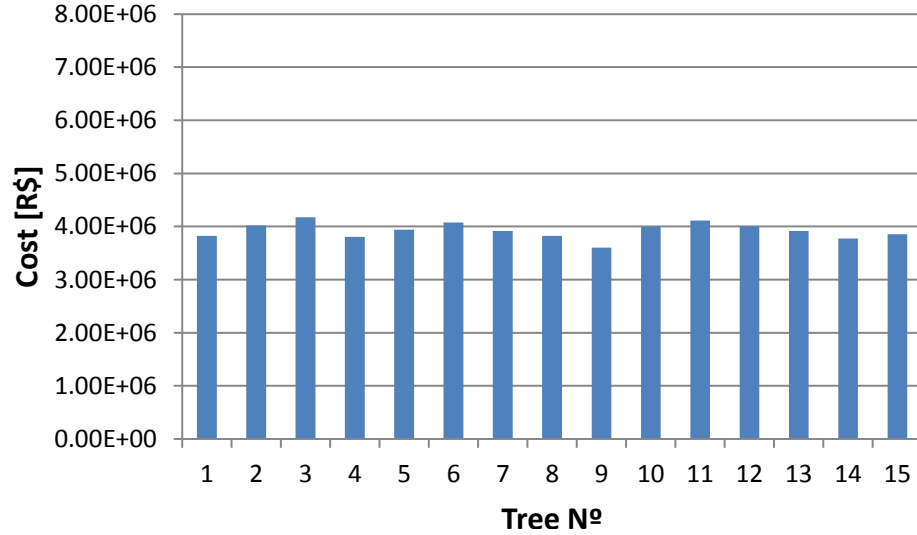


Figure A.14: Lower Bound for Each Tree on Instance 07

presented in Table A.27. The $(1-\alpha)$ -level confidence interval on the optimality gap $\mathbb{E}U - z^*$ and its associated error with respect to U_{n_u} for this instance are presented in Table A.28. The error associated with the CI for this instance is around 6.9%.

| $L_{n_\ell}[R\$]$ | $S_\ell[R\$]$ | $\epsilon_\ell[R\$]$ | $\epsilon_\ell\%$ |
|-------------------|---------------|----------------------|-------------------|
| 3922280.6 | 148825.7 | 67669.3 | 1.7% |

Table A.26: Output of LBE for Instance 07

| $gap[R\$]$ | $gap\%$ |
|------------|---------|
| 154542.1 | 3.79% |

Table A.27: Gap Between U_{n_u} and L_{n_ℓ} and Associated Error for Instance 07

| CI for $\mathbb{E}U - z^*$ | CI% |
|----------------------------|------|
| [0, 280722.9] | 6.9% |

Table A.28: CI for $\mathbb{E}U - z^*$ and Associated Error for Instance 07

Appendix B

SDDP Convergence Tables

| Iteration | Zub - Δ | Zlb | Zub | Zub + Δ | Time [s] |
|-----------|----------------|----------|----------|----------------|----------|
| 0 | 1.09E+08 | 0.00E+00 | 1.17E+08 | 1.25E+08 | 5.53 |
| 1 | 1.49E+08 | 4.40E+04 | 1.55E+08 | 1.60E+08 | 6.06 |
| 2 | 6.03E+07 | 4.37E+05 | 6.42E+07 | 6.80E+07 | 7.07 |
| 3 | 4.32E+07 | 7.10E+05 | 4.55E+07 | 4.78E+07 | 8.06 |
| 4 | 2.67E+07 | 1.37E+06 | 2.83E+07 | 2.98E+07 | 9.18 |
| 5 | 1.78E+07 | 1.90E+06 | 1.89E+07 | 2.00E+07 | 10.14 |
| 6 | 1.41E+07 | 2.97E+06 | 1.55E+07 | 1.68E+07 | 11.00 |
| 7 | 8.45E+06 | 3.63E+06 | 9.30E+06 | 1.01E+07 | 12.03 |
| 8 | 8.16E+06 | 4.15E+06 | 9.09E+06 | 1.00E+07 | 12.94 |
| 9 | 5.57E+06 | 4.54E+06 | 6.32E+06 | 7.08E+06 | 13.95 |
| 10 | 5.54E+06 | 4.69E+06 | 6.46E+06 | 7.38E+06 | 15.00 |
| 11 | 4.36E+06 | 4.77E+06 | 4.97E+06 | 5.57E+06 | 16.06 |
| 12 | 4.65E+06 | 4.84E+06 | 5.34E+06 | 6.04E+06 | 17.37 |
| 13 | 4.71E+06 | 4.89E+06 | 5.46E+06 | 6.22E+06 | 18.41 |
| 14 | 5.36E+06 | 4.92E+06 | 6.23E+06 | 7.10E+06 | 19.62 |
| 15 | 4.43E+06 | 4.95E+06 | 5.23E+06 | 6.03E+06 | 20.88 |
| 16 | 4.54E+06 | 5.00E+06 | 5.23E+06 | 5.93E+06 | 22.11 |
| 17 | 4.88E+06 | 5.01E+06 | 5.74E+06 | 6.60E+06 | 23.23 |
| 18 | 4.08E+06 | 5.02E+06 | 4.90E+06 | 5.73E+06 | 24.42 |
| 19 | 4.10E+06 | 5.03E+06 | 4.66E+06 | 5.22E+06 | 25.66 |
| 20 | 4.43E+06 | 5.04E+06 | 5.21E+06 | 5.99E+06 | 26.88 |
| 21 | 4.29E+06 | 5.04E+06 | 4.93E+06 | 5.58E+06 | 28.01 |
| 22 | 4.68E+06 | 5.05E+06 | 5.45E+06 | 6.22E+06 | 29.56 |
| 23 | 4.27E+06 | 5.05E+06 | 4.90E+06 | 5.52E+06 | 30.68 |
| 24 | 4.36E+06 | 5.06E+06 | 4.99E+06 | 5.62E+06 | 31.90 |
| 25 | 4.27E+06 | 5.06E+06 | 4.92E+06 | 5.58E+06 | 33.11 |
| 26 | 4.37E+06 | 5.06E+06 | 5.36E+06 | 6.36E+06 | 34.63 |
| 27 | 4.48E+06 | 5.07E+06 | 5.18E+06 | 5.88E+06 | 35.62 |
| 28 | 4.43E+06 | 5.07E+06 | 5.18E+06 | 5.93E+06 | 36.73 |
| 29 | 4.29E+06 | 5.07E+06 | 4.91E+06 | 5.54E+06 | 37.88 |
| 30 | 4.42E+06 | 5.07E+06 | 5.16E+06 | 5.89E+06 | 39.99 |
| 31 | 4.12E+06 | 5.08E+06 | 4.65E+06 | 5.17E+06 | 40.99 |
| 32 | 4.58E+06 | 5.08E+06 | 5.36E+06 | 6.14E+06 | 42.40 |
| 33 | 4.70E+06 | 5.08E+06 | 5.42E+06 | 6.15E+06 | 43.41 |
| 34 | 4.48E+06 | 5.08E+06 | 5.15E+06 | 5.82E+06 | 44.93 |
| 35 | 4.25E+06 | 5.08E+06 | 4.95E+06 | 5.64E+06 | 46.49 |
| 36 | 4.00E+06 | 5.08E+06 | 4.61E+06 | 5.22E+06 | 47.87 |
| 37 | 5.12E+06 | 5.08E+06 | 6.01E+06 | 6.90E+06 | 48.61 |
| 38 | 4.80E+06 | 5.08E+06 | 5.63E+06 | 6.45E+06 | 49.97 |
| 39 | 4.67E+06 | 5.09E+06 | 5.32E+06 | 5.97E+06 | 50.70 |
| 40 | 4.35E+06 | 5.09E+06 | 4.95E+06 | 5.55E+06 | 52.62 |
| 41 | 4.14E+06 | 5.09E+06 | 4.80E+06 | 5.47E+06 | 54.18 |
| 42 | 4.44E+06 | 5.09E+06 | 5.08E+06 | 5.71E+06 | 55.01 |
| 43 | 3.91E+06 | 5.09E+06 | 4.54E+06 | 5.17E+06 | 56.68 |
| 44 | 3.98E+06 | 5.09E+06 | 4.53E+06 | 5.09E+06 | 57.71 |
| 45 | 4.39E+06 | 5.09E+06 | 5.05E+06 | 5.71E+06 | 60.56 |

Table B.1: SDDP Convergence Instance 01. Zlb is the lower bound, Zub is the sample mean upper bound, and Δ is a 95% confidence interval half-width associated with upper bound estimator. The column labeled time is the time in seconds, required for that iteration.

| Iteration | Zub - Δ | Zlb | Zub | Zub + Δ | Time [s] |
|-----------|----------------|----------|----------|----------------|----------|
| 0 | 8.77E+07 | 0.00E+00 | 9.47E+07 | 1.02E+08 | 9.99 |
| 1 | 1.42E+08 | 5.59E+02 | 1.48E+08 | 1.54E+08 | 11.28 |
| 2 | 5.00E+07 | 1.57E+05 | 5.30E+07 | 5.60E+07 | 13.07 |
| 3 | 2.90E+07 | 2.69E+05 | 3.08E+07 | 3.26E+07 | 14.59 |
| 4 | 1.90E+07 | 8.45E+05 | 2.01E+07 | 2.12E+07 | 16.31 |
| 5 | 1.11E+07 | 1.30E+06 | 1.22E+07 | 1.33E+07 | 17.78 |
| 6 | 7.06E+06 | 1.76E+06 | 7.92E+06 | 8.78E+06 | 19.26 |
| 7 | 5.12E+06 | 2.06E+06 | 6.26E+06 | 7.39E+06 | 20.83 |
| 8 | 4.44E+06 | 2.30E+06 | 5.51E+06 | 6.59E+06 | 22.19 |
| 9 | 3.40E+06 | 2.52E+06 | 4.17E+06 | 4.94E+06 | 24.13 |
| 10 | 3.87E+06 | 2.72E+06 | 4.86E+06 | 5.84E+06 | 25.73 |
| 11 | 3.41E+06 | 2.83E+06 | 4.69E+06 | 5.96E+06 | 27.53 |
| 12 | 6.91E+06 | 2.89E+06 | 7.89E+06 | 8.87E+06 | 29.59 |
| 13 | 3.62E+06 | 2.92E+06 | 4.48E+06 | 5.34E+06 | 31.58 |
| 14 | 3.14E+06 | 2.95E+06 | 3.76E+06 | 4.38E+06 | 33.58 |
| 15 | 2.79E+06 | 2.98E+06 | 3.59E+06 | 4.38E+06 | 35.50 |
| 16 | 2.92E+06 | 3.00E+06 | 3.52E+06 | 4.12E+06 | 37.91 |
| 17 | 2.68E+06 | 3.01E+06 | 3.37E+06 | 4.06E+06 | 39.51 |
| 18 | 2.56E+06 | 3.02E+06 | 3.12E+06 | 3.68E+06 | 41.52 |
| 19 | 1.99E+06 | 3.03E+06 | 2.45E+06 | 2.91E+06 | 43.65 |
| 20 | 3.17E+06 | 3.04E+06 | 3.88E+06 | 4.58E+06 | 45.34 |
| 21 | 2.32E+06 | 3.04E+06 | 2.83E+06 | 3.33E+06 | 47.22 |
| 22 | 2.44E+06 | 3.04E+06 | 2.99E+06 | 3.54E+06 | 49.52 |
| 23 | 2.70E+06 | 3.05E+06 | 3.34E+06 | 3.97E+06 | 52.35 |
| 24 | 2.35E+06 | 3.05E+06 | 2.85E+06 | 3.34E+06 | 53.98 |
| 25 | 2.10E+06 | 3.05E+06 | 2.51E+06 | 2.93E+06 | 55.61 |
| 26 | 2.73E+06 | 3.06E+06 | 3.37E+06 | 4.00E+06 | 58.96 |
| 27 | 2.35E+06 | 3.06E+06 | 2.85E+06 | 3.35E+06 | 61.31 |
| 28 | 2.52E+06 | 3.06E+06 | 3.06E+06 | 3.61E+06 | 63.79 |
| 29 | 2.56E+06 | 3.06E+06 | 3.18E+06 | 3.79E+06 | 64.76 |
| 30 | 2.71E+06 | 3.06E+06 | 3.26E+06 | 3.81E+06 | 68.04 |
| 31 | 2.38E+06 | 3.06E+06 | 2.93E+06 | 3.47E+06 | 70.48 |
| 32 | 2.50E+06 | 3.07E+06 | 3.12E+06 | 3.73E+06 | 74.18 |
| 33 | 2.44E+06 | 3.07E+06 | 2.99E+06 | 3.55E+06 | 75.52 |
| 34 | 2.51E+06 | 3.07E+06 | 3.17E+06 | 3.84E+06 | 78.94 |
| 35 | 2.67E+06 | 3.07E+06 | 3.27E+06 | 3.87E+06 | 82.54 |
| 36 | 2.19E+06 | 3.07E+06 | 2.70E+06 | 3.22E+06 | 85.51 |
| 37 | 2.53E+06 | 3.07E+06 | 3.03E+06 | 3.54E+06 | 87.93 |
| 38 | 2.60E+06 | 3.07E+06 | 3.13E+06 | 3.66E+06 | 89.39 |
| 39 | 2.62E+06 | 3.07E+06 | 3.20E+06 | 3.77E+06 | 90.96 |
| 40 | 2.61E+06 | 3.07E+06 | 3.22E+06 | 3.83E+06 | 93.65 |
| 41 | 2.40E+06 | 3.07E+06 | 2.94E+06 | 3.48E+06 | 96.71 |
| 42 | 1.98E+06 | 3.07E+06 | 2.40E+06 | 2.81E+06 | 99.30 |
| 43 | 2.56E+06 | 3.07E+06 | 3.05E+06 | 3.54E+06 | 101.57 |
| 44 | 2.69E+06 | 3.07E+06 | 3.37E+06 | 4.05E+06 | 103.93 |
| 45 | 2.47E+06 | 3.07E+06 | 3.07E+06 | 3.66E+06 | 108.05 |

Table B.2: SDDP Convergence Instance 02. The columns in this table are analogous to those in Table B.1.

| Iteration | Zub - Δ | Zlb | Zub | Zub + Δ | Time [s] |
|-----------|----------------|----------|----------|----------------|----------|
| 0 | 1.06E+08 | 0.00E+00 | 1.15E+08 | 1.23E+08 | 10.62 |
| 1 | 1.11E+08 | 1.54E+04 | 1.17E+08 | 1.23E+08 | 12.09 |
| 2 | 4.03E+07 | 1.67E+05 | 4.31E+07 | 4.59E+07 | 14.11 |
| 3 | 2.46E+07 | 1.13E+06 | 2.65E+07 | 2.84E+07 | 15.89 |
| 4 | 1.43E+07 | 2.09E+06 | 1.57E+07 | 1.71E+07 | 17.96 |
| 5 | 9.54E+06 | 3.44E+06 | 1.07E+07 | 1.19E+07 | 19.75 |
| 6 | 6.51E+06 | 3.67E+06 | 7.19E+06 | 7.87E+06 | 21.50 |
| 7 | 5.88E+06 | 4.02E+06 | 6.90E+06 | 7.92E+06 | 23.96 |
| 8 | 5.63E+06 | 4.23E+06 | 6.38E+06 | 7.14E+06 | 25.07 |
| 9 | 4.89E+06 | 4.37E+06 | 5.75E+06 | 6.60E+06 | 27.36 |
| 10 | 4.61E+06 | 4.46E+06 | 5.36E+06 | 6.11E+06 | 29.33 |
| 11 | 4.11E+06 | 4.52E+06 | 4.69E+06 | 5.27E+06 | 31.56 |
| 12 | 3.56E+06 | 4.57E+06 | 4.13E+06 | 4.71E+06 | 34.10 |
| 13 | 4.40E+06 | 4.61E+06 | 5.08E+06 | 5.76E+06 | 36.43 |
| 14 | 3.82E+06 | 4.65E+06 | 4.44E+06 | 5.05E+06 | 39.02 |
| 15 | 4.71E+06 | 4.66E+06 | 5.49E+06 | 6.27E+06 | 41.84 |
| 16 | 3.74E+06 | 4.68E+06 | 4.31E+06 | 4.88E+06 | 45.06 |
| 17 | 4.10E+06 | 4.69E+06 | 4.75E+06 | 5.41E+06 | 47.26 |
| 18 | 4.07E+06 | 4.70E+06 | 4.75E+06 | 5.42E+06 | 50.04 |
| 19 | 4.11E+06 | 4.71E+06 | 4.78E+06 | 5.45E+06 | 52.70 |
| 20 | 4.24E+06 | 4.73E+06 | 5.10E+06 | 5.96E+06 | 56.35 |
| 21 | 4.02E+06 | 4.73E+06 | 4.91E+06 | 5.79E+06 | 59.13 |
| 22 | 4.24E+06 | 4.74E+06 | 4.88E+06 | 5.53E+06 | 61.82 |
| 23 | 3.85E+06 | 4.74E+06 | 4.58E+06 | 5.32E+06 | 64.02 |
| 24 | 4.09E+06 | 4.74E+06 | 4.75E+06 | 5.42E+06 | 67.46 |
| 25 | 4.74E+06 | 4.75E+06 | 5.70E+06 | 6.66E+06 | 70.66 |
| 26 | 3.80E+06 | 4.75E+06 | 4.51E+06 | 5.22E+06 | 73.67 |
| 27 | 3.78E+06 | 4.75E+06 | 4.34E+06 | 4.89E+06 | 76.30 |
| 28 | 4.24E+06 | 4.76E+06 | 5.11E+06 | 5.99E+06 | 79.89 |
| 29 | 3.64E+06 | 4.76E+06 | 4.22E+06 | 4.79E+06 | 81.50 |
| 30 | 4.23E+06 | 4.76E+06 | 4.86E+06 | 5.49E+06 | 85.58 |
| 31 | 3.89E+06 | 4.76E+06 | 4.62E+06 | 5.34E+06 | 87.85 |
| 32 | 3.88E+06 | 4.76E+06 | 4.51E+06 | 5.14E+06 | 91.62 |
| 33 | 4.00E+06 | 4.76E+06 | 4.76E+06 | 5.52E+06 | 94.07 |
| 34 | 3.89E+06 | 4.77E+06 | 4.92E+06 | 5.94E+06 | 95.41 |
| 35 | 3.95E+06 | 4.77E+06 | 4.68E+06 | 5.40E+06 | 100.83 |
| 36 | 4.00E+06 | 4.77E+06 | 4.70E+06 | 5.41E+06 | 103.78 |
| 37 | 3.85E+06 | 4.77E+06 | 4.52E+06 | 5.19E+06 | 104.93 |
| 38 | 4.06E+06 | 4.77E+06 | 4.90E+06 | 5.73E+06 | 107.39 |
| 39 | 4.24E+06 | 4.77E+06 | 4.90E+06 | 5.56E+06 | 109.35 |
| 40 | 3.92E+06 | 4.77E+06 | 4.74E+06 | 5.55E+06 | 112.70 |
| 41 | 4.19E+06 | 4.77E+06 | 4.81E+06 | 5.43E+06 | 117.04 |
| 42 | 4.05E+06 | 4.77E+06 | 4.85E+06 | 5.64E+06 | 119.44 |
| 43 | 4.26E+06 | 4.77E+06 | 4.93E+06 | 5.59E+06 | 124.19 |
| 44 | 4.04E+06 | 4.77E+06 | 4.69E+06 | 5.35E+06 | 126.46 |
| 45 | 4.46E+06 | 4.78E+06 | 5.22E+06 | 5.97E+06 | 128.78 |

Table B.3: SDDP Convergence Instance 03. The columns in this table are analogous to those in Table B.1.

| Iteration | Zub - Δ | Zlb | Zub | Zub + Δ | Time [s] |
|-----------|----------------|----------|----------|----------------|----------|
| 0 | 8.81E+07 | 0.00E+00 | 9.52E+07 | 1.02E+08 | 11.65 |
| 1 | 1.19E+08 | 1.16E+04 | 1.24E+08 | 1.30E+08 | 13.59 |
| 2 | 5.77E+07 | 1.67E+05 | 6.10E+07 | 6.44E+07 | 16.07 |
| 3 | 2.88E+07 | 3.58E+05 | 3.08E+07 | 3.27E+07 | 18.33 |
| 4 | 2.20E+07 | 1.40E+06 | 2.37E+07 | 2.54E+07 | 20.68 |
| 5 | 1.14E+07 | 1.83E+06 | 1.25E+07 | 1.36E+07 | 22.95 |
| 6 | 6.79E+06 | 2.26E+06 | 7.65E+06 | 8.50E+06 | 25.00 |
| 7 | 5.89E+06 | 2.62E+06 | 6.88E+06 | 7.87E+06 | 27.34 |
| 8 | 4.09E+06 | 2.88E+06 | 4.74E+06 | 5.40E+06 | 29.55 |
| 9 | 4.16E+06 | 3.09E+06 | 5.05E+06 | 5.94E+06 | 32.25 |
| 10 | 3.06E+06 | 3.21E+06 | 3.61E+06 | 4.16E+06 | 34.85 |
| 11 | 3.01E+06 | 3.30E+06 | 3.56E+06 | 4.11E+06 | 37.44 |
| 12 | 3.27E+06 | 3.34E+06 | 3.88E+06 | 4.48E+06 | 40.14 |
| 13 | 3.11E+06 | 3.37E+06 | 3.78E+06 | 4.45E+06 | 43.05 |
| 14 | 3.24E+06 | 3.39E+06 | 3.90E+06 | 4.56E+06 | 46.37 |
| 15 | 2.66E+06 | 3.41E+06 | 3.19E+06 | 3.71E+06 | 49.46 |
| 16 | 2.37E+06 | 3.43E+06 | 2.82E+06 | 3.27E+06 | 52.47 |
| 17 | 2.92E+06 | 3.45E+06 | 3.47E+06 | 4.03E+06 | 55.14 |
| 18 | 3.00E+06 | 3.46E+06 | 3.55E+06 | 4.10E+06 | 58.36 |
| 19 | 3.34E+06 | 3.47E+06 | 3.98E+06 | 4.63E+06 | 61.40 |
| 20 | 2.64E+06 | 3.47E+06 | 3.20E+06 | 3.76E+06 | 64.09 |
| 21 | 3.66E+06 | 3.48E+06 | 4.30E+06 | 4.94E+06 | 67.16 |
| 22 | 2.96E+06 | 3.48E+06 | 3.57E+06 | 4.19E+06 | 70.55 |
| 23 | 2.83E+06 | 3.49E+06 | 3.45E+06 | 4.06E+06 | 72.71 |
| 24 | 3.11E+06 | 3.50E+06 | 3.73E+06 | 4.34E+06 | 76.12 |
| 25 | 3.19E+06 | 3.50E+06 | 3.87E+06 | 4.56E+06 | 78.43 |
| 26 | 3.08E+06 | 3.50E+06 | 3.66E+06 | 4.24E+06 | 82.65 |
| 27 | 3.29E+06 | 3.50E+06 | 3.96E+06 | 4.63E+06 | 85.54 |
| 28 | 3.27E+06 | 3.51E+06 | 3.95E+06 | 4.64E+06 | 88.69 |
| 29 | 2.39E+06 | 3.51E+06 | 2.86E+06 | 3.32E+06 | 89.71 |
| 30 | 3.10E+06 | 3.52E+06 | 3.76E+06 | 4.41E+06 | 94.70 |
| 31 | 2.51E+06 | 3.52E+06 | 3.06E+06 | 3.60E+06 | 97.97 |
| 32 | 2.75E+06 | 3.52E+06 | 3.51E+06 | 4.26E+06 | 102.26 |
| 33 | 2.80E+06 | 3.52E+06 | 3.62E+06 | 4.44E+06 | 103.55 |
| 34 | 2.22E+06 | 3.53E+06 | 2.67E+06 | 3.12E+06 | 108.11 |
| 35 | 3.07E+06 | 3.53E+06 | 3.77E+06 | 4.47E+06 | 110.84 |
| 36 | 2.96E+06 | 3.53E+06 | 3.54E+06 | 4.12E+06 | 114.02 |
| 37 | 2.67E+06 | 3.53E+06 | 3.34E+06 | 4.01E+06 | 116.31 |
| 38 | 3.11E+06 | 3.53E+06 | 3.79E+06 | 4.47E+06 | 118.75 |
| 39 | 3.20E+06 | 3.53E+06 | 3.80E+06 | 4.41E+06 | 121.41 |
| 40 | 2.87E+06 | 3.53E+06 | 3.52E+06 | 4.17E+06 | 124.65 |
| 41 | 3.27E+06 | 3.53E+06 | 3.95E+06 | 4.63E+06 | 128.00 |
| 42 | 2.71E+06 | 3.53E+06 | 3.24E+06 | 3.78E+06 | 131.36 |
| 43 | 3.24E+06 | 3.54E+06 | 3.96E+06 | 4.68E+06 | 136.15 |
| 44 | 2.72E+06 | 3.54E+06 | 3.26E+06 | 3.81E+06 | 136.64 |
| 45 | 3.65E+06 | 3.54E+06 | 4.26E+06 | 4.88E+06 | 141.41 |

Table B.4: SDDP Convergence Instance 04. The columns in this table are analogous to those in Table B.1.

| Iteration | Zub - Δ | Zlb | Zub | Zub + Δ | Time [s] |
|-----------|----------------|----------|----------|----------------|----------|
| 0 | 9.19E+07 | 0.00E+00 | 9.94E+07 | 1.07E+08 | 14.93 |
| 1 | 1.13E+08 | 3.18E+04 | 1.18E+08 | 1.23E+08 | 18.07 |
| 2 | 4.82E+07 | 1.35E+05 | 5.07E+07 | 5.31E+07 | 21.83 |
| 3 | 3.06E+07 | 7.06E+05 | 3.29E+07 | 3.51E+07 | 25.60 |
| 4 | 1.83E+07 | 8.98E+05 | 1.93E+07 | 2.03E+07 | 29.68 |
| 5 | 1.22E+07 | 1.71E+06 | 1.38E+07 | 1.53E+07 | 33.57 |
| 6 | 7.17E+06 | 2.27E+06 | 8.33E+06 | 9.48E+06 | 37.06 |
| 7 | 5.56E+06 | 2.58E+06 | 6.47E+06 | 7.39E+06 | 41.04 |
| 8 | 4.29E+06 | 2.85E+06 | 5.10E+06 | 5.90E+06 | 44.69 |
| 9 | 4.25E+06 | 3.08E+06 | 5.38E+06 | 6.52E+06 | 49.42 |
| 10 | 3.48E+06 | 3.18E+06 | 4.09E+06 | 4.71E+06 | 53.77 |
| 11 | 3.56E+06 | 3.27E+06 | 4.33E+06 | 5.09E+06 | 57.88 |
| 12 | 3.77E+06 | 3.36E+06 | 4.81E+06 | 5.84E+06 | 62.28 |
| 13 | 3.26E+06 | 3.42E+06 | 4.28E+06 | 5.29E+06 | 67.37 |
| 14 | 2.66E+06 | 3.46E+06 | 3.15E+06 | 3.64E+06 | 72.87 |
| 15 | 2.86E+06 | 3.49E+06 | 3.42E+06 | 3.98E+06 | 77.35 |
| 16 | 2.82E+06 | 3.50E+06 | 3.59E+06 | 4.36E+06 | 82.55 |
| 17 | 3.61E+06 | 3.52E+06 | 4.32E+06 | 5.03E+06 | 85.66 |
| 18 | 2.93E+06 | 3.54E+06 | 3.64E+06 | 4.35E+06 | 90.93 |
| 19 | 2.84E+06 | 3.54E+06 | 3.55E+06 | 4.26E+06 | 97.04 |
| 20 | 2.86E+06 | 3.55E+06 | 3.37E+06 | 3.89E+06 | 100.53 |
| 21 | 3.28E+06 | 3.56E+06 | 4.00E+06 | 4.72E+06 | 104.40 |
| 22 | 3.48E+06 | 3.57E+06 | 4.26E+06 | 5.05E+06 | 111.05 |
| 23 | 2.86E+06 | 3.57E+06 | 3.53E+06 | 4.20E+06 | 116.09 |
| 24 | 2.74E+06 | 3.58E+06 | 3.41E+06 | 4.09E+06 | 121.36 |
| 25 | 2.96E+06 | 3.58E+06 | 3.76E+06 | 4.56E+06 | 123.74 |
| 26 | 2.66E+06 | 3.59E+06 | 3.34E+06 | 4.03E+06 | 131.08 |
| 27 | 3.06E+06 | 3.59E+06 | 3.80E+06 | 4.54E+06 | 137.15 |
| 28 | 3.20E+06 | 3.59E+06 | 3.83E+06 | 4.47E+06 | 142.30 |
| 29 | 3.10E+06 | 3.60E+06 | 3.65E+06 | 4.21E+06 | 143.52 |
| 30 | 2.80E+06 | 3.60E+06 | 3.38E+06 | 3.95E+06 | 152.47 |
| 31 | 3.07E+06 | 3.60E+06 | 3.58E+06 | 4.10E+06 | 156.78 |
| 32 | 2.59E+06 | 3.60E+06 | 3.24E+06 | 3.88E+06 | 163.43 |
| 33 | 3.10E+06 | 3.60E+06 | 3.91E+06 | 4.72E+06 | 166.83 |
| 34 | 3.17E+06 | 3.61E+06 | 3.88E+06 | 4.60E+06 | 164.84 |
| 35 | 3.34E+06 | 3.61E+06 | 3.95E+06 | 4.57E+06 | 169.13 |
| 36 | 3.25E+06 | 3.61E+06 | 4.08E+06 | 4.91E+06 | 172.74 |
| 37 | 2.88E+06 | 3.61E+06 | 3.47E+06 | 4.05E+06 | 176.90 |
| 38 | 2.94E+06 | 3.61E+06 | 3.53E+06 | 4.12E+06 | 181.21 |
| 39 | 2.87E+06 | 3.61E+06 | 3.57E+06 | 4.27E+06 | 184.84 |
| 40 | 2.80E+06 | 3.62E+06 | 3.45E+06 | 4.10E+06 | 189.50 |
| 41 | 3.16E+06 | 3.62E+06 | 3.91E+06 | 4.66E+06 | 194.43 |
| 42 | 3.25E+06 | 3.62E+06 | 3.84E+06 | 4.42E+06 | 197.98 |
| 43 | 2.67E+06 | 3.62E+06 | 3.36E+06 | 4.05E+06 | 205.63 |
| 44 | 3.01E+06 | 3.62E+06 | 3.63E+06 | 4.24E+06 | 208.58 |
| 45 | 3.37E+06 | 3.62E+06 | 4.13E+06 | 4.88E+06 | 213.70 |

Table B.5: SDDP Convergence Instance 05. The columns in this table are analogous to those in Table B.1.

| Iteration | Zub - Δ | Zlb | Zub | Zub + Δ | Time [s] |
|-----------|----------------|----------|----------|----------------|----------|
| 0 | 1.07E+08 | 0.00E+00 | 1.16E+08 | 1.24E+08 | 48.10 |
| 1 | 5.42E+07 | 3.53E+04 | 5.76E+07 | 6.11E+07 | 61.68 |
| 2 | 6.17E+07 | 1.16E+05 | 6.75E+07 | 7.33E+07 | 77.64 |
| 3 | 2.64E+07 | 3.92E+05 | 2.77E+07 | 2.90E+07 | 92.00 |
| 4 | 1.50E+07 | 8.92E+05 | 1.61E+07 | 1.73E+07 | 105.98 |
| 5 | 1.32E+07 | 2.28E+06 | 1.45E+07 | 1.58E+07 | 119.67 |
| 6 | 8.89E+06 | 2.70E+06 | 1.03E+07 | 1.18E+07 | 133.01 |
| 7 | 5.20E+06 | 2.94E+06 | 6.01E+06 | 6.82E+06 | 147.83 |
| 8 | 4.46E+06 | 3.11E+06 | 5.15E+06 | 5.84E+06 | 161.12 |
| 9 | 4.69E+06 | 3.34E+06 | 5.56E+06 | 6.43E+06 | 178.91 |
| 10 | 3.82E+06 | 3.58E+06 | 4.40E+06 | 4.98E+06 | 195.39 |
| 11 | 4.10E+06 | 3.68E+06 | 4.90E+06 | 5.69E+06 | 214.52 |
| 12 | 3.65E+06 | 3.77E+06 | 4.23E+06 | 4.81E+06 | 232.12 |
| 13 | 3.59E+06 | 3.83E+06 | 4.41E+06 | 5.24E+06 | 248.51 |
| 14 | 2.86E+06 | 3.88E+06 | 3.45E+06 | 4.04E+06 | 268.79 |
| 15 | 3.39E+06 | 3.90E+06 | 4.10E+06 | 4.81E+06 | 288.89 |
| 16 | 3.71E+06 | 3.92E+06 | 4.52E+06 | 5.33E+06 | 312.10 |
| 17 | 3.38E+06 | 3.93E+06 | 3.89E+06 | 4.40E+06 | 327.58 |
| 18 | 3.71E+06 | 3.94E+06 | 4.46E+06 | 5.21E+06 | 349.79 |
| 19 | 3.22E+06 | 3.95E+06 | 3.76E+06 | 4.31E+06 | 367.42 |
| 20 | 3.78E+06 | 3.97E+06 | 4.40E+06 | 5.01E+06 | 385.90 |
| 21 | 3.86E+06 | 3.98E+06 | 4.52E+06 | 5.17E+06 | 403.99 |
| 22 | 3.77E+06 | 3.98E+06 | 4.56E+06 | 5.35E+06 | 426.87 |
| 23 | 3.23E+06 | 3.98E+06 | 3.83E+06 | 4.42E+06 | 443.54 |
| 24 | 3.07E+06 | 3.99E+06 | 3.62E+06 | 4.17E+06 | 461.84 |
| 25 | 2.85E+06 | 4.00E+06 | 3.43E+06 | 4.00E+06 | 477.15 |
| 26 | 3.85E+06 | 4.00E+06 | 4.67E+06 | 5.49E+06 | 502.64 |
| 27 | 3.52E+06 | 4.00E+06 | 4.30E+06 | 5.08E+06 | 524.18 |
| 28 | 3.35E+06 | 4.01E+06 | 4.10E+06 | 4.85E+06 | 544.33 |
| 29 | 3.38E+06 | 4.01E+06 | 4.10E+06 | 4.81E+06 | 548.81 |
| 30 | 2.99E+06 | 4.01E+06 | 3.47E+06 | 3.96E+06 | 586.93 |
| 31 | 3.53E+06 | 4.01E+06 | 4.20E+06 | 4.86E+06 | 602.12 |
| 32 | 3.18E+06 | 4.02E+06 | 3.81E+06 | 4.44E+06 | 630.99 |
| 33 | 3.19E+06 | 4.02E+06 | 3.79E+06 | 4.40E+06 | 638.96 |
| 34 | 3.26E+06 | 4.02E+06 | 3.85E+06 | 4.45E+06 | 670.54 |
| 35 | 3.94E+06 | 4.02E+06 | 4.78E+06 | 5.63E+06 | 690.56 |
| 36 | 3.46E+06 | 4.02E+06 | 4.17E+06 | 4.88E+06 | 705.73 |
| 37 | 2.93E+06 | 4.03E+06 | 3.66E+06 | 4.39E+06 | 720.06 |
| 38 | 3.44E+06 | 4.03E+06 | 3.97E+06 | 4.49E+06 | 740.68 |
| 39 | 4.06E+06 | 4.03E+06 | 4.68E+06 | 5.29E+06 | 750.80 |
| 40 | 2.86E+06 | 4.03E+06 | 3.36E+06 | 3.86E+06 | 773.49 |
| 41 | 2.64E+06 | 4.03E+06 | 3.21E+06 | 3.78E+06 | 796.42 |
| 42 | 3.30E+06 | 4.03E+06 | 3.93E+06 | 4.56E+06 | 810.05 |
| 43 | 3.09E+06 | 4.03E+06 | 3.69E+06 | 4.28E+06 | 836.66 |
| 44 | 3.34E+06 | 4.03E+06 | 3.99E+06 | 4.63E+06 | 851.37 |
| 45 | 3.60E+06 | 4.03E+06 | 4.22E+06 | 4.85E+06 | 869.42 |

Table B.6: SDDP Convergence Instance 06. The columns in this table are analogous to those in Table B.1.

| Iteration | Zub - Δ | Zlb | Zub | Zub + Δ | Time [s] |
|-----------|----------------|----------|----------|----------------|----------|
| 0 | 9.87E+07 | 0.00E+00 | 1.07E+08 | 1.15E+08 | 94.18 |
| 1 | 1.38E+08 | 3.56E+03 | 1.43E+08 | 1.49E+08 | 120.81 |
| 2 | 5.46E+07 | 9.39E+04 | 5.82E+07 | 6.18E+07 | 152.23 |
| 3 | 3.17E+07 | 7.26E+05 | 3.34E+07 | 3.51E+07 | 180.34 |
| 4 | 1.97E+07 | 1.45E+06 | 2.10E+07 | 2.23E+07 | 207.62 |
| 5 | 1.15E+07 | 2.62E+06 | 1.26E+07 | 1.37E+07 | 235.04 |
| 6 | 7.44E+06 | 3.24E+06 | 8.53E+06 | 9.62E+06 | 260.78 |
| 7 | 6.28E+06 | 3.50E+06 | 7.35E+06 | 8.43E+06 | 292.29 |
| 8 | 4.55E+06 | 3.64E+06 | 5.38E+06 | 6.20E+06 | 316.71 |
| 9 | 4.87E+06 | 3.73E+06 | 5.75E+06 | 6.63E+06 | 352.50 |
| 10 | 4.30E+06 | 3.82E+06 | 5.31E+06 | 6.32E+06 | 381.89 |
| 11 | 3.70E+06 | 3.88E+06 | 4.41E+06 | 5.12E+06 | 417.74 |
| 12 | 3.76E+06 | 3.91E+06 | 4.49E+06 | 5.23E+06 | 456.20 |
| 13 | 3.13E+06 | 3.93E+06 | 3.88E+06 | 4.64E+06 | 492.25 |
| 14 | 3.31E+06 | 3.94E+06 | 3.94E+06 | 4.58E+06 | 537.44 |
| 15 | 3.48E+06 | 3.95E+06 | 4.21E+06 | 4.94E+06 | 575.35 |
| 16 | 3.75E+06 | 3.96E+06 | 4.67E+06 | 5.60E+06 | 611.37 |
| 17 | 3.24E+06 | 3.98E+06 | 3.96E+06 | 4.67E+06 | 647.16 |
| 18 | 2.87E+06 | 3.98E+06 | 3.58E+06 | 4.29E+06 | 679.98 |
| 19 | 3.40E+06 | 3.99E+06 | 4.14E+06 | 4.88E+06 | 723.73 |
| 20 | 3.76E+06 | 4.01E+06 | 4.69E+06 | 5.62E+06 | 759.28 |
| 21 | 3.79E+06 | 4.01E+06 | 4.48E+06 | 5.18E+06 | 794.67 |
| 22 | 4.10E+06 | 4.01E+06 | 4.86E+06 | 5.62E+06 | 840.84 |
| 23 | 2.98E+06 | 4.02E+06 | 3.61E+06 | 4.23E+06 | 881.82 |
| 24 | 3.21E+06 | 4.02E+06 | 3.91E+06 | 4.61E+06 | 912.92 |
| 25 | 3.44E+06 | 4.03E+06 | 4.15E+06 | 4.85E+06 | 947.24 |
| 26 | 3.30E+06 | 4.03E+06 | 3.90E+06 | 4.50E+06 | 997.94 |
| 27 | 3.49E+06 | 4.03E+06 | 4.22E+06 | 4.95E+06 | 1039.28 |
| 28 | 2.99E+06 | 4.04E+06 | 3.78E+06 | 4.57E+06 | 1071.48 |
| 29 | 3.21E+06 | 4.04E+06 | 3.85E+06 | 4.49E+06 | 1085.47 |
| 30 | 3.16E+06 | 4.04E+06 | 3.74E+06 | 4.33E+06 | 1166.31 |
| 31 | 3.46E+06 | 4.04E+06 | 4.12E+06 | 4.79E+06 | 1205.69 |
| 32 | 3.42E+06 | 4.04E+06 | 4.10E+06 | 4.77E+06 | 1252.01 |
| 33 | 3.55E+06 | 4.05E+06 | 4.14E+06 | 4.73E+06 | 1269.85 |
| 34 | 3.62E+06 | 4.05E+06 | 4.52E+06 | 5.41E+06 | 1323.57 |
| 35 | 3.58E+06 | 4.05E+06 | 4.31E+06 | 5.05E+06 | 1358.18 |
| 36 | 3.73E+06 | 4.05E+06 | 4.46E+06 | 5.19E+06 | 1398.16 |
| 37 | 3.39E+06 | 4.05E+06 | 4.28E+06 | 5.17E+06 | 1431.09 |
| 38 | 3.42E+06 | 4.05E+06 | 4.06E+06 | 4.70E+06 | 1471.88 |
| 39 | 3.52E+06 | 4.05E+06 | 4.39E+06 | 5.26E+06 | 1490.13 |
| 40 | 3.12E+06 | 4.06E+06 | 3.83E+06 | 4.55E+06 | 1536.66 |
| 41 | 3.35E+06 | 4.06E+06 | 4.09E+06 | 4.84E+06 | 1576.45 |
| 42 | 3.95E+06 | 4.06E+06 | 4.76E+06 | 5.57E+06 | 1601.00 |
| 43 | 3.29E+06 | 4.06E+06 | 4.08E+06 | 4.88E+06 | 1664.50 |
| 44 | 3.86E+06 | 4.06E+06 | 4.60E+06 | 5.34E+06 | 1682.40 |
| 45 | 3.64E+06 | 4.06E+06 | 4.32E+06 | 5.00E+06 | 1726.67 |

Table B.7: SDDP Convergence Instance 07. The columns in this table are analogous to those in Table B.1.

Appendix C

Study Case Parameters

| Month | Region 01 | Region 02 | Region 03 | Region 04 |
|-------|-----------|-----------|-----------|-----------|
| 1 | 35632 | 9647 | 8181 | 3772 |
| 2 | 36269 | 9747 | 8212 | 3792 |
| 3 | 34882 | 9329 | 8242 | 3904 |
| 4 | 34248 | 9095 | 8109 | 3971 |
| 5 | 34091 | 9080 | 7960 | 3975 |
| 6 | 34141 | 9017 | 7980 | 3944 |
| 7 | 34596 | 8968 | 8073 | 3983 |
| 8 | 34739 | 8876 | 8254 | 3995 |
| 9 | 34903 | 8926 | 8440 | 3980 |
| 10 | 34521 | 9017 | 8464 | 3968 |
| 11 | 33973 | 9156 | 8422 | 3933 |
| 12 | 36223 | 9841 | 8793 | 4201 |
| 13 | 37169 | 10061 | 8740 | 4191 |
| 14 | 37658 | 10165 | 8760 | 4180 |
| 15 | 36940 | 9729 | 8683 | 4179 |
| 16 | 36200 | 9485 | 8542 | 4250 |
| 17 | 36038 | 9469 | 8386 | 4255 |
| 18 | 36146 | 9403 | 8407 | 4222 |
| 19 | 36638 | 9353 | 8505 | 4264 |
| 20 | 36706 | 9257 | 8695 | 4276 |
| 21 | 36999 | 9308 | 8892 | 4260 |
| 22 | 36614 | 9404 | 8917 | 5385 |
| 23 | 36139 | 9548 | 8873 | 5259 |
| 24 | 38070 | 10250 | 9226 | 5420 |

Table C.1: Electricity Demand in MW-month for Each Region

| Month | Region 01 | Region 02 | Region 03 | Region 04 |
|-------|-----------|-----------|-----------|-----------|
| 1 | 1660 | 407 | 253 | 39 |
| 2 | 1727 | 378 | 255 | 37 |
| 3 | 2782 | 650 | 225 | 45 |
| 4 | 3603 | 715 | 250 | 38 |
| 5 | 3766 | 748 | 254 | 36 |
| 6 | 3667 | 779 | 301 | 32 |
| 7 | 3894 | 780 | 422 | 29 |
| 8 | 3907 | 838 | 515 | 28 |
| 9 | 3950 | 874 | 605 | 30 |
| 10 | 3859 | 852 | 589 | 37 |
| 11 | 3111 | 783 | 535 | 45 |
| 12 | 3178 | 782 | 410 | 41 |
| 13 | 3316 | 788 | 398 | 49 |
| 14 | 3170 | 761 | 313 | 49 |
| 15 | 3547 | 826 | 284 | 45 |
| 16 | 4606 | 898 | 332 | 38 |
| 17 | 4775 | 941 | 341 | 36 |
| 18 | 4671 | 972 | 425 | 32 |
| 19 | 4592 | 967 | 523 | 29 |
| 20 | 4573 | 1020 | 630 | 28 |
| 21 | 4647 | 1013 | 710 | 30 |
| 22 | 4418 | 959 | 683 | 37 |
| 23 | 3443 | 864 | 623 | 45 |
| 24 | 3282 | 820 | 464 | 41 |

Table C.2: Generation From Small Hydro Plants in MW-month for Regions

| Month | Region 01 | Region 02 | Region 03 | Region 04 |
|-------|-----------|-----------|-----------|-----------|
| 1 | 105.98 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 121.23 | 0 | 0 | 0 |
| 4 | 142.38 | 0 | 0 | 0 |
| 5 | 133.65 | 22.95 | 0 | 0 |
| 6 | 52.2 | 0 | 0 | 0 |
| 7 | 52.2 | 0 | 0 | 0 |
| 8 | 0 | 192.42 | 0 | 0 |
| 9 | 0 | 192.42 | 0 | 0 |
| 10 | 95.67 | 384.84 | 0 | 0 |
| 11 | 95.67 | 384.84 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 |
| 13 | 91.71 | 34.65 | 0 | 122.31 |
| 14 | 91.71 | 0 | 0 | 244.62 |
| 15 | 0 | 0 | 0 | 244.62 |
| 16 | 0 | 105.03 | 0 | 366.93 |
| 17 | 24.12 | 105.03 | 0 | 366.93 |
| 18 | 0 | 0 | 0 | 489.24 |
| 19 | 0 | 0 | 0 | 489.24 |
| 20 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 |
| 23 | 14.4 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 |

Table C.3: Hydro Generation From Plants Operating in State I [MW-month]

| Month | Region 01 | Region 02 | Region 03 | Region 04 |
|-------|-----------|-----------|-----------|-----------|
| 1 | 2311.92 | 346 | 349.5 | 0 |
| 2 | 2316.84 | 333.69 | 349.5 | 0 |
| 3 | 2316.84 | 722.59 | 349.5 | 0 |
| 4 | 2316.84 | 733.21 | 349.5 | 0 |
| 5 | 2316.84 | 733.21 | 349.5 | 0 |
| 6 | 2104.28 | 733.21 | 349.5 | 0 |
| 7 | 2184.35 | 852.74 | 349.5 | 0 |
| 8 | 2343.94 | 871.19 | 349.5 | 0 |
| 9 | 1862.74 | 871.19 | 349.5 | 0 |
| 10 | 2015.35 | 725.19 | 349.5 | 0 |
| 11 | 2763.71 | 875.19 | 349.5 | 0 |
| 12 | 2755.4 | 864.09 | 572.5 | 0 |
| 13 | 2755.4 | 776.09 | 572.5 | 0 |
| 14 | 2755.4 | 860.92 | 572.5 | 0 |
| 15 | 2755.4 | 917.96 | 572.5 | 0 |
| 16 | 2755.4 | 921.13 | 572.5 | 0 |
| 17 | 2755.4 | 921.13 | 572.5 | 0 |
| 18 | 2755.4 | 939.96 | 572.5 | 0 |
| 19 | 2755.4 | 1038.96 | 572.5 | 0 |
| 20 | 2755.4 | 1060.63 | 572.5 | 0 |
| 21 | 2755.4 | 936.63 | 572.5 | 0 |
| 22 | 2755.4 | 835.59 | 572.5 | 507.62 |
| 23 | 2755.4 | 838.92 | 572.5 | 507.62 |
| 24 | 2755.4 | 864.09 | 572.5 | 507.62 |

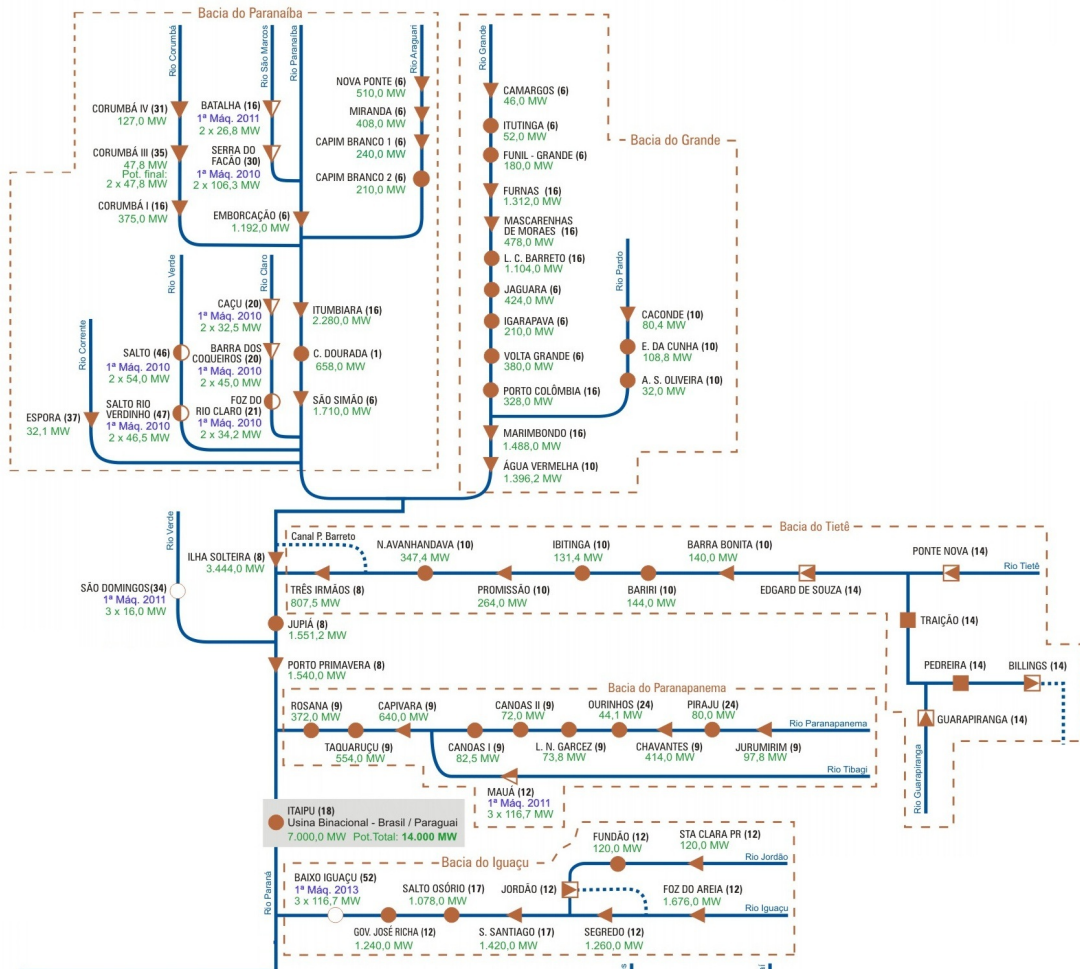
Table C.4: Minimal Thermal Generation for Each Region [MW-month]

| Month | High | Med | Low |
|-------|--------|--------|--------|
| 1 | 0.1025 | 0.5171 | 0.3804 |
| 2 | 0.1089 | 0.5349 | 0.3562 |
| 3 | 0.1 | 0.5083 | 0.3917 |
| 4 | 0.1008 | 0.5108 | 0.3884 |
| 5 | 0.1042 | 0.5208 | 0.375 |
| 6 | 0.1089 | 0.5349 | 0.3562 |
| 7 | 0.1048 | 0.5228 | 0.3724 |
| 8 | 0.1042 | 0.5208 | 0.375 |
| 9 | 0.1009 | 0.5114 | 0.3877 |
| 10 | 0.1 | 0.5083 | 0.3917 |
| 11 | 0.1048 | 0.5228 | 0.3724 |
| 12 | 0.1008 | 0.5108 | 0.3884 |
| 13 | 0.107 | 0.5305 | 0.3625 |
| 14 | 0.1048 | 0.5228 | 0.3724 |
| 15 | 0.1 | 0.5083 | 0.3917 |
| 16 | 0.1048 | 0.5228 | 0.3724 |
| 17 | 0.1042 | 0.5208 | 0.375 |
| 18 | 0.1048 | 0.5228 | 0.3724 |
| 19 | 0.1089 | 0.5349 | 0.3562 |
| 20 | 0.1042 | 0.5208 | 0.375 |
| 21 | 0.1009 | 0.5114 | 0.3877 |
| 22 | 0.1 | 0.5083 | 0.3917 |
| 23 | 0.1089 | 0.5349 | 0.3562 |
| 24 | 0.1048 | 0.5228 | 0.3724 |

Table C.5: Load Levels Duration

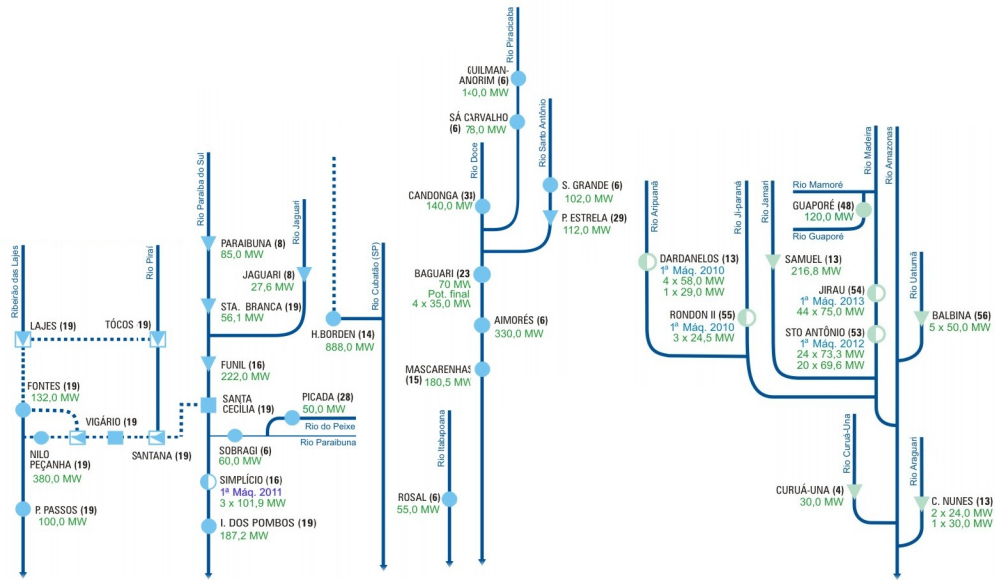
| Month | Region 01 | | | Region 02 | | | Region 03 | | | Region 04 | | |
|-------|-----------|------|------|-----------|------|------|-----------|------|------|-----------|------|------|
| | High | Med | Low | High | Med | Low | High | Med | Low | High | Med | Low |
| 1 | 1.15 | 1.07 | 0.86 | 1.1 | 1.1 | 0.84 | 1.15 | 1.03 | 0.92 | 1.07 | 1.01 | 0.96 |
| 2 | 1.18 | 1.07 | 0.84 | 1.15 | 1.09 | 0.81 | 1.15 | 1.04 | 0.89 | 1.08 | 1.02 | 0.95 |
| 3 | 1.21 | 1.08 | 0.84 | 1.23 | 1.11 | 0.8 | 1.17 | 1.05 | 0.89 | 1.09 | 1.02 | 0.95 |
| 4 | 1.24 | 1.08 | 0.83 | 1.26 | 1.11 | 0.79 | 1.17 | 1.05 | 0.89 | 1.09 | 1.02 | 0.95 |
| 5 | 1.24 | 1.07 | 0.83 | 1.26 | 1.1 | 0.79 | 1.18 | 1.04 | 0.89 | 1.08 | 1.02 | 0.95 |
| 6 | 1.24 | 1.07 | 0.82 | 1.26 | 1.1 | 0.77 | 1.18 | 1.04 | 0.88 | 1.08 | 1.02 | 0.95 |
| 7 | 1.23 | 1.07 | 0.83 | 1.25 | 1.1 | 0.79 | 1.18 | 1.05 | 0.88 | 1.08 | 1.02 | 0.95 |
| 8 | 1.22 | 1.07 | 0.84 | 1.25 | 1.1 | 0.79 | 1.17 | 1.05 | 0.88 | 1.09 | 1.02 | 0.95 |
| 9 | 1.2 | 1.08 | 0.84 | 1.21 | 1.11 | 0.8 | 1.16 | 1.05 | 0.89 | 1.09 | 1.01 | 0.96 |
| 10 | 1.17 | 1.08 | 0.85 | 1.16 | 1.12 | 0.81 | 1.16 | 1.05 | 0.9 | 1.09 | 1.01 | 0.96 |
| 11 | 1.17 | 1.07 | 0.85 | 1.12 | 1.1 | 0.83 | 1.16 | 1.03 | 0.91 | 1.08 | 1.01 | 0.96 |
| 12 | 1.13 | 1.08 | 0.86 | 1.07 | 1.1 | 0.85 | 1.13 | 1.04 | 0.91 | 1.07 | 1.01 | 0.96 |
| 13 | 1.15 | 1.07 | 0.86 | 1.09 | 1.09 | 0.84 | 1.15 | 1.03 | 0.91 | 1.07 | 1.01 | 0.96 |
| 14 | 1.18 | 1.08 | 0.84 | 1.16 | 1.1 | 0.82 | 1.15 | 1.04 | 0.9 | 1.08 | 1.02 | 0.95 |
| 15 | 1.21 | 1.08 | 0.84 | 1.23 | 1.11 | 0.8 | 1.17 | 1.05 | 0.89 | 1.09 | 1.02 | 0.95 |
| 16 | 1.23 | 1.08 | 0.83 | 1.25 | 1.1 | 0.78 | 1.17 | 1.05 | 0.88 | 1.09 | 1.02 | 0.95 |
| 17 | 1.24 | 1.07 | 0.83 | 1.26 | 1.1 | 0.79 | 1.18 | 1.04 | 0.89 | 1.08 | 1.02 | 0.95 |
| 18 | 1.25 | 1.07 | 0.83 | 1.27 | 1.11 | 0.78 | 1.18 | 1.05 | 0.88 | 1.08 | 1.02 | 0.95 |
| 19 | 1.23 | 1.07 | 0.83 | 1.24 | 1.09 | 0.78 | 1.17 | 1.05 | 0.88 | 1.08 | 1.02 | 0.95 |
| 20 | 1.22 | 1.07 | 0.84 | 1.25 | 1.1 | 0.79 | 1.17 | 1.05 | 0.88 | 1.09 | 1.02 | 0.95 |
| 21 | 1.2 | 1.08 | 0.84 | 1.21 | 1.11 | 0.8 | 1.16 | 1.05 | 0.89 | 1.09 | 1.01 | 0.96 |
| 22 | 1.17 | 1.08 | 0.85 | 1.16 | 1.12 | 0.81 | 1.16 | 1.05 | 0.9 | 1.09 | 1.01 | 0.96 |
| 23 | 1.17 | 1.07 | 0.85 | 1.12 | 1.09 | 0.83 | 1.16 | 1.03 | 0.91 | 1.08 | 1.01 | 0.96 |
| 24 | 1.13 | 1.07 | 0.86 | 1.06 | 1.1 | 0.84 | 1.12 | 1.04 | 0.91 | 1.07 | 1.01 | 0.96 |

Table C.6: Load Levels Base Multipliers



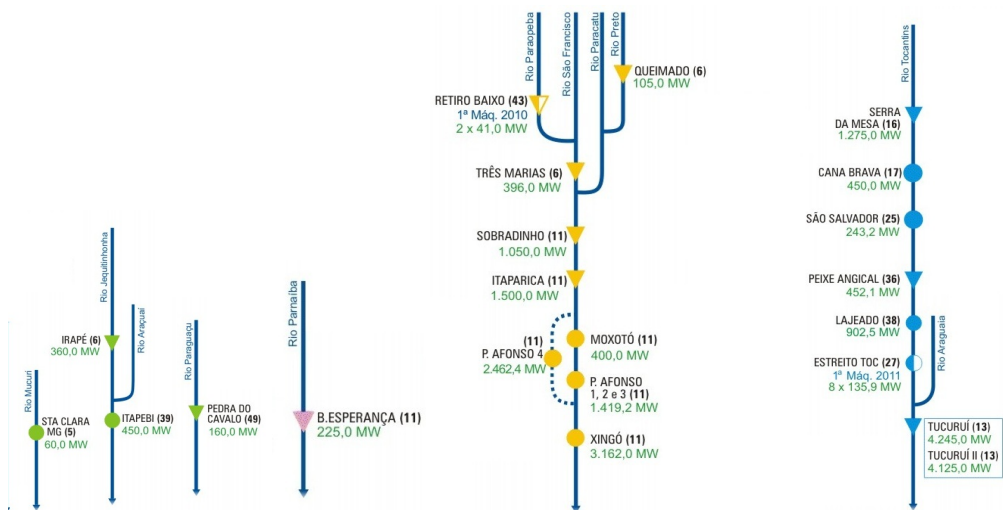
(d) Parana

Figure C.1: Power Plants Cascade Scheme by Basin, source: ONS (2009)



(e) Southeast Atlantic

(f) Amazon



(g) East Atlantic

(h) Parnaíba

(i) São Francisco

(j) Tocantins

Figure C.1: Power Plants Cascade Scheme by Basin, source: ONS (2009)

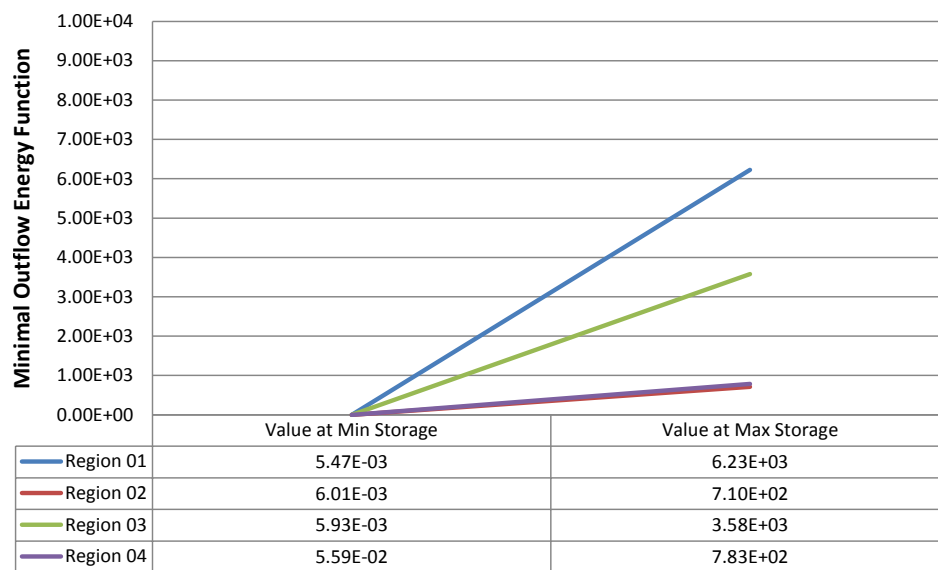


Figure C.2: Minimal Outflow Energy Functions of Storage for Each Region - 1st Month

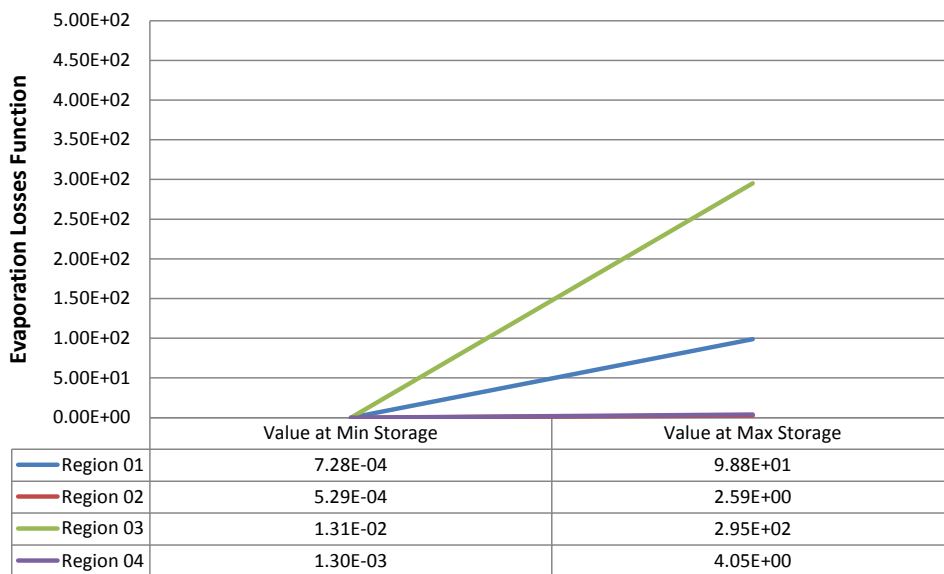


Figure C.3: Evaporation Losses Functions of Storage for Each Region - 1st Month

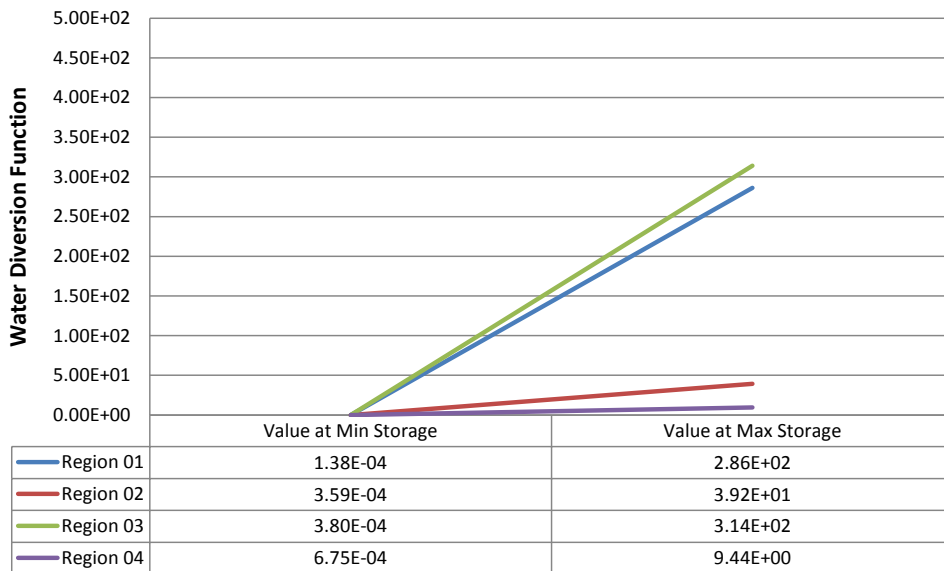


Figure C.4: Water Diversion Functions of Storage for Each Region - 1st Month

Bibliography

- ANEEL, . Manual da metodologia nodal para calculo de tarifas de uso dos sistemas eletricos, NT 003/1999, 1999. website <http://www.aneel.gov.br>.
- Arvanitidis, N.V. and Rosing, J. Composite representation of a multireservoir hydroelectric power system. *IEEE Transactions on Power Apparatus and Systems*, PAS-89:319–326, 1970a.
- Arvanitidis, N.V. and Rosing, J. Optimal operation of multireservoir systems using a composite representation. *IEEE Transactions on Power Apparatus and Systems*, PAS-89:327–335, 1970b.
- Bailey, T. G., Jensen, P. A., and Morton, D. P. Response surface analysis of two-stage stochastic linear programming with recourse. *Naval Research Logistics*, 46:753–778, 1999.
- Bayraksan, G. and Morton, D. P. Assessing solution quality in stochastic programs. *Mathematical Programming*, pages 495–514, 2006.
- Bayraksan, G. and Morton, D. P. A sequential sampling procedure for stochastic programming, 2009a. To appear in *Operations Research*.
- Bayraksan, G. and Morton, D. P. Assessing solution quality in stochastic programs via sampling. In Oskoorouchi, M., editor, *INFORMS Tutorials*. To appear. INFORMS, 2009b.

- Bellman, R. E. *Dynamic Programming*. Princeton University Press, 1957.
- Benders, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.
- Bertsekas, D.P. *Dynamic Programming and Optimal Control, Volume 1 and 2*. Athena Scientific, Belmont, Massachusetts, 3rd edition, 2005.
- Birge, J.R. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33:989–1007, 1985.
- CCEE, . Download de Deck de Precos NEWAVE, Deck de Fevereiro de 2010, 2010. Downloaded from the website <http://www.ccee.org.br>.
- Cepel, Eletrobras. Manual de referencia - modelo NEWAVE. Technical report, Centro de Pesquisas de Energia Eletrica, 2001.
- Chen, Z.L. and Powell, W.B. Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. *Journal of Optimization Theory and Applications*, 102(3):497–524, 1999.
- Chiralaksanakul, A. *Monte Carlo Methods for Multistage Stochastic Programming*. PhD thesis, The University of Texas at Austin, 2003.
- Chiralaksanakul, A. and Morton, D. P. Assessing policy quality in multi-stage stochastic programming. *The Stochastic Programming E-Print Series*, 2004. www.speps.info.

- da Cruz Jr., G. and Soares, S. General composite representation of hydroelectric systems. In *PICA IEEE International Conference*, pages 177–182, 1999.
- de Carvalho, J.F. Prioridades para investimentos em usinas eletricas. *Scielo Brasil*, 2008.
- de Matos, V. L., Finardi, E. C., and da Silva, E. L. Comparison between the energy equivalent reservoir per subsystem and per cascade in the long-term operation planning in Brazil. In *EngOpt 2008—International Conference on Engineering Optimization*, Rio de Janeiro, Brazil, 2008.
- de Matos, V.L. *Analise Comparativa entre as Modelagens de Reservatorio Equivalente de Energia Agregado por Subistema e por Cascata no Problema do Planejamento Anual da Operacao Energetica*. UFSC Florianopolis, Santa Catarina, 2008. M.S. Thesis.
- Diwekar, U. M. and Kalagnanam, J. R. An efficient sampling technique for optimization under uncertainty. *American Institute of Chemical Engineers Journal*, 43:440, 1997.
- Donohue, C. *Stochastic Network Programming and the Dynamic Vehicle Allocation Problem*. PhD thesis, The University of Michigan, Ann Arbor, 1996.
- Donohue, C. J. and Birge, J. R. The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithmic Operations Research*, 1:20–30, 2006.

- Douglas, T. “Green” hydro power understanding impacts, approvals, and sustainability of run-of-river independent power projects in British Columbia. Technical report, Watershed Watch Salmon Society, 2007.
- EPE, . Cesb vence leilao de concessao da hidreletrica de Jirau, no Rio Madeira, 2008. website <http://www.epe.gov.br>.
- Freimer, M., Thomas, D., and Linderoth, J. T. The impact of sampling methods on bias and variance in stochastic linear programs. Technical report, Technical Report 05T-002, Industrial and Systems Engineering, Lehigh University, 2005.
- Guan, Z. and Philpott, A.B. A multistage stochastic programming model for the New Zealand dairy industry. *International Journal of Production Economics*, 2009.
- Homem-de-Mello, T. On rates of convergence for stochastic optimization problems under non-independent and identically distributed sampling. *SIAM Journal on Optimization*, 19:524–551, 2008.
- Homem-de-Mello, T., de Matos, V.L., and Finardi, E.C. Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Systems*, 2:1–31, 2011.
- Infanger, G. and Morton, D. P. Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming*, 75:241–256, 1996.

- Jacobs, J., Freeman, G., Grygier, J., Morton, D., Schultz, G., Staschus, K., and Stedinger, J. SOCRATES: a system for scheduling hydroelectric generation under uncertainty. *Annals of Operations Research*, 59:99–133, 1995.
- Klingerman, A.S. *Operao tima de Subsistemas Hidrotrmicos Interligados Utilizando Programao Dinamica Estocstica Dual*. UNICAMP Campinas, So Paulo, 1992. M.S. Thesis.
- Linderoth, J. T., Shapiro, A., and Wright, S. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142:215–241, 2006.
- Linowsky, K. and Philpott, A. B. On the convergence of sampling-based decomposition algorithms for multi-stage stochastic programs. *Journal of Optimization Theory and Applications*, 125:349–366, 2005.
- Maceira, M.E.P., Mercio, C.B., Gorestin, B.G., Cunha, S.H.F., Suanno, C., Sacramento, M.C., and Klingerman, A.S. Energy evaluation of the north/northeastern and south/southeastern interconnection with NEWAVE model. In *SEPOPE Salvador Brazil*, 1998.
- Maceira, M.E.P., Terry, L.A., Costa, F.S., Damzio, J.M., and Melo, A.C.G. Chain of optimization models for setting the energy dispatch and spot price in the Brazilian system. In *PSCC Sevilla*, pages 1–7, 2002.
- Maceira, M.E.P., Duarte, V.S., Penna, D.D.J., Moraes, L.A.M., and Melo, A.C.G. Ten years of application of stochastic dual dynamic programming

- in official and agent studies in Brazil description of the newave program. In *PSCC Glasgow*, 2008.
- Mak, W. K., Morton, D. P., and Wood, R. K. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24:47–56, 1999.
- Marangon Lima, L.M. *Modeling and forecast of Brazilian reservoir inflows via dynamic linear models under climate change scenarios*. PhD thesis, The University of Texas at Austin, 2011.
- Marcato, A.L.M. *Representacao hibrida de sistemas equivalentes e individualizados para o planejamento da operacao de medio prazo de sistemas de potencia de grande porte*. PhD thesis, Pontificia Universidade Catolica do Rio de Janeiro, 2001.
- Morton, D. P. An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling. *Annals of Operations Research*, 64:211–235, 1996.
- ONS, . Diagrama Esquemático das Usinas Hidrelétricas do SIN, 2009. Downloaded from the website http://www.ons.org.br/download/biblioteca_virtual/publicacoes/dados_relevantes_2009/02-Diagrama-Esquemático-das-Usinas-Hidrelétricas-do-SIN.html.
- ONS, . Produção por Subsistema - GWh e Taxa de Crescimento, 2011. Downloaded from the website <http://www.ons.org.br>.

- Pacheco, P.S. *Parallel programming with MPI*. Morgan Kaufmann, San Francisco, California, 1st edition, 1997.
- Partani, A. *Adaptive Jackknife Estimators for Stochastic Programming*. PhD thesis, The University of Texas at Austin, 2007.
- Partani, A., Morton, D. P., and Popova, I. Jackknife estimators for reducing bias in asset allocation. In *Proceedings of the Winter Simulation Conference*, 2006.
- Pennanen, T. and Koivu, M. Epi-convergent discretizations of stochastic programs via integration quadratures. *Numerische Mathematik*, 100:141–163, 2005.
- Pereira, M.V.F. Optimal stochastic operations scheduling of large hydroelectric systems. *International Journal of Electrical Power and Energy Systems*, 11:161–169, 1989.
- Pereira, M.V.F. and Pinto, L.M.V.G. Stochastic optimization of a multireservoir hydroelectric system—a decomposition approach. *Water Resources Research*, 21:779–792, 1985.
- Pereira, M.V.F. and Pinto, L.M.V.G. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52:359–375, 1991.
- Philpott, A.B. and de Matos, V.L. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. Technical report, Electric Power Optimization Centre, University of Auckland, 2010.

- Philpott, A.B. and Guan, Z. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36:450–455, 2008.
- Quinn, M.J. *Parallel programming in C with MPI and OpenMP*. TATA McGRAW-HILL, New York, New York, 1st edition, 2004.
- Rebennack, S. *A unified state-space and scenario tree framework for multi-stage stochastic optimization: an application to emission-constrained hydro-thermal scheduling*. PhD thesis, University of Florida, 2010.
- Rockafellar, R.T. and Uryasev, S. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.
- Shapiro, A. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209:63–72, 2011.
- Supatgiat, C. Applications of Stochastic Programming in the Energy Industry, 2001. <http://www.uoptimize.com/chonawee/presentations/houston01.pdf>.
- Terry, L.A. Modelo a sistema equivalente - descricao geral. Technical report, Centro de Pesquisas de Energia Eletrica, 1980.
- Van Slyke, R. M. and Wets, R. J.-B. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

Velasquez, J., Restrepo, P.J., and Campo, R. Dual dynamic programming: A note on implementation. *Water Resources Research*, 35:2269–2271, 1999.

Vita

Anderson Rodrigo de Queiroz was born in Mogi Mirim, São Paulo on 11 August 1982, the son of Norival Antonio de Queiroz and Aparecida Solange Balasini. He received his B.Sc. and M.Sc. degrees in Electrical Engineering both from Federal University at Itajuba, Itajuba, Brazil, in 2005 and 2007, respectively. In July of 2007 he was awarded with a Capes/Fulbright Scholarship and he was accepted as a Ph.D. student in Operations Research and Industrial Engineering (ORIE) at the Mechanical Engineering Department at the University of Texas at Austin. He received his Ph.D. degree in ORIE in December of 2011. His research interest include operations research and electricity markets. His focus is on optimization techniques as linear, integer, stochastic and large scale programming, electricity prices, transmission and distribution regulation, and hydro-thermal scheduling and coordination.

Permanent address: ar_queiroz@yahoo.com.br

This dissertation was typeset with L^AT_EX[†] by the author.

[†]L^AT_EX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T_EX Program.