UNIVERSIDADE FEDERAL DE SANTA CATARINA

CAMPUS REITOR JOAO DAVID FERREIRA LIMA

POST-GRADUATION PROGRAM IN ELECTRICAL ENGINEERING

FELIPE BELTRÁN RODRÍGUEZ

A MULTI-STAGE STOCHASTIC OPTIMIZATION MODEL FOR MEDIUM-TERM GENERATION SCHEDULING PROBLEM

Florianópolis

2021

Felipe Beltrán Rodíguez

A MULTI-STAGE STOCHASTIC OPTIMIZATION MODEL FOR MEDIUM-TERM GENERATION SCHEDULING PROBLEM

Doctoral Thesis submitted to the Post-graduation Program in Electrical Engineering of the Universidade Federal de Santa Catarina for obtaining the Grade of Doctor of Electrical Engineering.
Advisor
Prof. Dr. Erlon Cristian Finardi, Universidade Federal de Santa Catarina
Co-advisor
Prof. D. Habil. Welington Luis de Oliveira, MINES ParisTech, PSL-Research University, CMA-Centre de Mathématiques Appliquées, France

Florianópolis

2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Beltrán Rodríguez, Felipe
A MULTI-STAGE STOCHASTIC OPTIMIZATION MODEL FOR MEDIUM TERM GENERATION SCHEDULING PROBLEM / Felipe Beltrán Rodríguez ; orientador, Erlon Finardi, coorientador, Welington de Oliveira, 2021.
136 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2021.

Inclui referências.

1. Engenharia Elétrica. 2. Programação estocástica. 3. Problema de planejamento de curto prazo. 4. Método de nível estendido. I. Finardi, Erlon. II. de Oliveira, Welington. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

Felipe Beltrán Rodríguez

A MULTI-STAGE STOCHASTIC OPTIMIZATION MODEL FOR MEDIUM-TERM GENERATION SCHEDULING PROBLEM

This Doctoral Thesis has been evaluated and approved by the examining committee composed of the following members:

Prof.(a) D. Habil. Claudia Alejandra Sagastizábal
Universidade Estadual de Campinas, Campinas, Brasil

Prof. Dr. André Luiz Diniz
CEPEL, Rio de Janeiro, Brasil

Dr. Paulo Vitor Larroyd
NORUS, Florianópolis, Brasil

We certify that this is **the original and final version** of the Doctoral Thesis, which was deemed suitable to obtain the Grade of Doctor of Electrical Engineering.

ICP-Edu
Documento assinado digitalmente
Telles Brunelli Lazzarin
Data: 08/04/2021 18:56:34-0300
CPF: 023.400.719-25
Verifique as assinaturas em https://v.ufsc.br

Prof. Dr. Telles Brunelli Lazzarin
Coordinator
Universidade Federal de Santa Catarina

Erlon Cristian Finardi
Prof. Erlon Cristian Finardi, Dr Eng
LabPlan/EEL/CTC/UFSC
Assinado de forma digital por Erlon Cristian Finardi:02036474918
Dados: 2021.04.07 15:28:20 -03'00'

Prof. Dr. Erlon Cristian Finardi
Advisor
Universidade Federal de Santa Catarina

Florianópolis, 2021.

This thesis is dedicated to my mom Patricia, my sister Laura, my dad Iván and my love Paula.

ACKNOWLEDGEMENTS

This is the most crucial section of the thesis for the author since it is possible to gratefully acknowledge all the people that are always supporting this project, giving me all their help, advice, and good energy throughout all these years of work. The only thing I can say to you is: Thank you so much. I want to mention all my family for, despite the physical distance, they have always made me feel close. To my dad Iván Beltrán, my mom Patricia Rodríguez and my sister Laura Beltrán for waiting patiently at the end of every year, to be able to see each other again.

I have special acknowledgements to my advisor Erlon Finardi and my co-advisor Welington de Oliveira for trusting me and giving me the opportunity of becoming a doctor (it was not in my plans). I feel so proud of the team that we formed. I highlight their kindness, availability, good sense of humor and professionalism. It is very easy to work this way, thank you. This work was improved by all the qualifying board recommendations, by Claudia Sagastizábal, André Diniz and Paulo Larroyd. Thank you for your dedication in the reviewing process. The present work reached an international publication level because of your suggestions.

I would like to thank Paula Muntal for all her support and patience during all these years when I only talked about the thesis (no one said it was easy being in a relationship with a doctoral student). Also, she is practically the editor of this thesis and other published papers, since she reviewed the English language proficiency level and learned together about stochastic programming.

This research has the financial support of CNPq and P&D project SPARHTACUS II PD-07427-0318/2018. Also, I want to thank the company Norus Energy for opening their doors to me. Norus is the place where I could maintain an equilibrium that allowed me to continue my research.

Finally, I thank all my friends and Labplan teammates: Guilherme Fredo, Guilherme Matiussi, Gilseu Von Muhlen, Renata Pedrini, Marcelo Cordova, Bruno Colonetti, Brunno Brito, Kenny Vinente, Vitor de Matos, Paulo Larroyd, Rodolfo Machado, Murilo Scuzziato, Pedro Vieira and Rodolfo Bielecki. In addition, my beloved friends back home, Maria Claudia Machado, Lorena Mora, Daniel Giraldo, Carlos Moreno and Juan Giraldo.

RESUMO

O problema do Planejamento da Operação Energética visa obter uma política operativa para um horizonte de planejamento multianual. Devido a uma quantidade considerável de complexidades, tal problema é decomposto em uma série de problemas acoplados entre si. O problema de curto prazo pertencente a esta série, acoplando o problema de médio prazo com a programação diária da operação. Para aprimorar esse acoplamento, o presente trabalho apresenta uma nova abordagem do problema de curto prazo caraterizada por: (*i*) inclusão da incerteza nas afluências a partir da segunda semana do horizonte de planejamento, por meio de uma árvore de cenários com amostras não comuns, (*ii*) discretização horária na primeira semana para serem inseridas restrições térmicas com variáveis binárias na primeira semana. Para encontrar um equilíbrio entre a precisão de uma solução calculada e o desempenho computacional, é proposta uma decomposição do problema em dois estágios. A ideia fundamental é obter cortes mais eficientes, quando comparado a decomposição multi-estágio, para representar a função de custo futuro, reduzindo assim, o esforço computacional. Este efeito é reforçado pelo uso da versão desagregada dos cortes e pela regularização do método estendido de nível. Os testes numéricos realizados com os dados do Sistema Elétrico Brasileiro indicam que: (*i*) a decomposição em dois estágios reduz em um 85% o tempo computacional requerido pela decomposição multi-estágio, (*ii*) a versão desagregada dos cortes melhora o desempenho computacional em torno de 30% quando comparado com a versão agregada, e (*iii*) o uso do método estendido de nível ao final do processo de resolução fornece uma redução de 20% do tempo computacional em relação a dois métodos clássicos: o L-Shaped e o algoritmo da Programação Dinâmica Dual Estocástica.

Palavras-chave: Problema do planejamento da operação energética, programação estocástica multiestágio, método de nível.

RESUMO EXPANDIDO

Introdução

A sustentabilidade técnica e econômica de um sistema elétrico de potência é suportada na implementação do problema do Planejamento da Operação Energética (POE). Este problema visa a obtenção de uma política operativa para um horizonte de planejamento multianual. Contudo, o POE é um problema de difícil solução devido, entre outros fatores, ao acoplamento espacial e temporal das usinas hidrelétricas, às não linearidades presentes nas modelagens dos elementos do sistema e à incerteza nas vazões afluentes. Na pratica, o POE é decomposto em uma série de problemas acoplados entre si, os quais consideram diferentes horizontes de planejamento e distintos graus de detalhamento da modelagem do sistema. O problema de curto prazo pertence a esta série, acoplando o problema de médio prazo com a programação diária da operação. Para aprimorar esse acoplamento, o presente trabalho apresenta uma nova abordagem do problema de curto prazo caraterizada por: (*i*) inclusão da incerteza nas afluências a partir da segunda semana do horizonte de planejamento, por meio de uma árvore de cenários com amostras não comuns, e (*ii*) uma discretização horária na primeira semana para serem inseridas restrições térmicas com variáveis binárias na primeira semana. Estas considerações resultam em um problema de grande porte, o qual precisa de técnicas sofisticadas de solução para manter um equilíbrio entre a precisão de uma solução calculada e o desempenho computacional. A principal proposta de solução é decompor o problema resultante de curto prazo em dois estágios, com o fim de obter iterativamente aproximações lineares da função de custo futuro mais eficientes. Adicionalmente, reforçar este efeito com o uso da versão desagregada dos cortes e pela regularização do método estendido de nível.

Objetivos

Propor técnicas de solução eficientes para um problema de curto prazo de grande porte, viabilizando o aprimoramento da representação das incertezas do problema e da representação do sistema nesta etapa de planejamento.

Metodologia

As metodologias de solução avaliadas neste trabalho podem ser divididas em técnicas para a decomposição multi-estágio e em dois estágios. Para o caso multi-estagio são avaliados os métodos da decomposição aninhada, a versão com centros de Chebyshev e a Programação Dinâmica Dual Estocástica (PDDE). Os métodos L-Shaped e o método estendido de nível são considerados para o caso da decomposição em dois estágios. Diferentes exemplos ilustrativos das técnicas de solução são apresentados. A avaliação computacional das metodologias de solução considera diferentes estruturas de árvores de cenários. Comparações entre a decomposição multi-estagio e dois estágios são realizadas, assim como a validação do impacto na convergência dos métodos de solução utilizando a versão agregada e desagregada dos cortes.

Resultados e Discussão

A qualidade das diretrizes operacionais para a programação diária da operação dependem significativamente da modelagem e a representação do processo estocástico no modelo de curto prazo. A inclusão das restrições térmicas de *Unit Commitmment* na primeira semana do problema de curto prazo apresenta uma redução da flexibilidade na geração térmica neste período (em ordem de 3%), impactando a região onde a função de custo futuro é aproximada. Respeito aos métodos de solução, a decomposição aninhada com centros de Chebyshev não

apresenta resultados superiores à decomposição aninhada clássica devido ao acoplamento inerente entre os modelos de curto e longo prazo. Contudo, a PDDE é melhorada utilizando os conceitos do método Chebyshev para a solução de problemas de longo prazo.
Respeito ao problema de curto prazo proposto, os testes numéricos utilizando um sistema de grande porte com dados do Sistema Elétrico Brasileiro indicam que: (*i*) a decomposição em dois estágios reduz em um 85% o tempo computacional requerido pela decomposição multi-estágio, (*ii*) a versão desagregada dos cortes melhora o desempenho computacional em torno de 30% quando comparado com a versão agregada, e (*iii*) o uso do método estendido de nível ao final do processo de resolução fornece uma redução de 20% do tempo computacional em relação a dois métodos clássicos: o L-Shaped e o algoritmo da Programação Dinâmica Dual Estocástica.

Considerações finais
Os benefícios dos aprimoramentos na modelagem do problema de curto prazo devem ser avaliados em um estudo de horizonte rolante acoplando todos os modelos da cadeia de planejamento. Metodologicamente, os benefícios da construção de cortes mais eficientes iterativamente podem ser estendidos ao método da PDDE. Finalmente, heurísticas para atualizar eficientemente o parâmetro de nível podem reduzir significativamente o esforço computacional do método estendido de nível

Palavras-chave: Problema do planejamento da operação energética, programação estocástica multiestágio, método de nível.

ABSTRACT

The Generation Scheduling (GS) problem aims at obtaining an optimal operation policy over a multi-year planning horizon. Due to a considerable amount of complexities, such a problem is decomposed into a series of optimization models coupled with each other. The medium-term GS (MTGS) problem belongs to this series of models, coupling long-term GS (LTGS) and short-term (STGS) models. To improve the aforementioned coupling, this work presents a new MTGS model characterized by (*i*) inflow uncertainty is considered from the second week of the planning horizon and is represented by a scenario tree with non-common sample realizations; (*ii*) an hourly discretization in the first week to insert the nonconvex thermal unit-commitment constraints, which exceed the computing capacity of the current solvers. To ensure an equilibrium between solution accuracy and computational performance, a two-stage decomposition of the MTGS problem is proposed. The main objective is to obtain better Benders cuts than those issued by a multi-stage decomposition, reducing the computational burden. This effect is strengthened by using a multi-cut approach and a regularization technique. Numerical assessments on the large-scale Brazilian MTGS problem indicate that: (*i*) the two-stage decomposition reduces by up to 85% the CPU time required by the multi-stage decomposition, (*ii*) the multi-cut version improves the computational performance by 30% when compared to the single-cut version, and (*iii*) the use of level set regularization at the end of the solution process provides higher computational performance by 20% compared with two classical methods: the L-Shaped and the Stochastic Dual Dynamic Programming algorithm.

Keywords: Generation scheduling problem, multi-stage stochastic programming, level bundle methods.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CCO	Chance constraint optimization;
CI	Confidence interval;
ECC	Energy commercialization chamber;
ELD	Extended level decomposition;
GS	Generation Scheduling;
HPF	Hydro production function;
ISO	Independent system operator;
LB	Level bundle;
LD	Level decomposition;
LP	Linear programming;
LTGS	Long-term generation scheduling;
MILP	Integer-mixed linear programming;
MSSLP	Multi-stage stochastic linear programming;
MTGS	Medium-term generation scheduling;
NBD	Nested Benders decomposition;
NBD-CC	Nested Benders decomposition with Chebyshev centers;
NC	Number of constraints;
NV	Number of variables;
PDF	Probability density function;
PH	Progressive hedging;
RO	Robust optimization;
SDDP	Stochastic dual dynamic programming;
SDDiP	Stochastic dual dynamic integer programming;
SLP	Stochastic linear programming;
SP	Stochastic programming;
STGS	Short-term generation scheduling;
UC	Unit commitment;

NOMENCLATURE

T	Index of stages, where t=1, …, T;
P	Index of periods, where p=1, …, P;
τ_t	Length of stage t;
Δp	Time-step (hourly, daily, weekly, monthly);
ξ, (Ξ)	Stochastic process (sampled space);
ξ^s	Scenario s of the stochastic process;
ξ_t^s	Realization of the random variable associated with scenario s and stage t;
$\mathbb{P}$	Filtered probability space;
$\Im$, (P)	Sigma algebra (probability measure);
c_t, (b_t)	Vector with the objective function (demand) constants in stage t;
A_t, (B_t)	Recourse (technological) matrix in stage t;
$\mathcal{Q}$	Variable related to the cost-to-go function in stage t;
$\mathcal{J}$	Set of cutting planes in the single-cut modeling of the cost-to-go function in stage t;
nj_t	Number of cutting planes in stage t of the single-cut modeling;
θ_{t+1}	Variable associated with the expected future cost of the single-cut modeling;
$\beta_{t+1}{}^j$, $\alpha_{t+1}{}^j$	Coefficients associated with the cut j of single-cut modeling;
$\mathcal{J}$:	Set of cutting planes associated with realization r of multi-cut modeling;
$nj_{t,r}$:	Number of cutting planes in stage t associated with realization r of multi-cut modeling;
$\theta_{t+1,r}$	Variable related to the cost-to-go function and random variable realization r in stage t of multi-cut modeling;
$\beta_{t+1,r}{}^j$, ($\alpha_{t+1,r}{}^j$)	Coefficients of the cut associated with $\theta_{t+1,r}$;
dv_t	The devaluation cost factor of stage t;
$\mathcal{L}$	Set of load levels of stage t;
$\mathcal{S}$	Set of subsystems;
$\mathcal{F}$, ($\mathcal{F}$	Set of thermal plants in stage t (and subsystem s);
$\mathcal{D}$, ($\mathcal{D}$	Set of deficit levels in stage t (and subsystem s);
$\mathcal{H}$, ($\mathcal{H}$	Set of hydro plants of stage t (and subsystem s);
$\mathcal{V}$	Set of hydro plants with storage capacity in stage t;
$\mathcal{R}$	Set of run-of-river hydro plants in stage t;
$\mathcal{M}$	Set of hydro plants upstream to the hydro plant h;
$\mathcal{I}$, ($\mathcal{I}$	Set of power interconnection lines arriving (coming) to subsystem s;
$\mathcal{G}$	Set of linear constraints that represent the production function of hydro h;
$\mathcal{N}$, ($\mathcal{N}$	Set of nodes of the scenario tree (set of nodes in stage t);
n	Index of the nodes in the scenario tree;
NS	Number of scenarios;

$l,(d)$	Load level (deficit level);
$s,(i)$	Index of subsystems (power interchanges);
$f,\ (h)$	Index of thermal (hydro) plants;
M	Index of upstream hydro plants;
v_{tphn}	Volume at the beginning of stage t and period p of hydro h related to node n;
q_{tphn}	Turbined outflow during stage t and period p of hydro h related to node n;
sh_{tphn}	Spillage during stage t and period p of hydro h related to node n;
y_{tphn}	Incremental inflow during stage t and period p of hydro h related to node n;
ph_{tphn}	Power generation during stage t and period p of hydro h related to node n;
pf_{tpfn}	Power generation during stage t and period p of thermal f related to node n;
pd_{tpdn}	Power deficit during stage t and period p on level d related to node n;
pi_{tpin}	Power during stage t and period p of interchange i related to node n;
pl_{tps}	Power load during stage t and period p of the subsystem s;
x_t	Decision vector of stage t, given by $x_t = [v_{tphn}, q_{tphn}, sh_{tphn}, ph_{tphn}, pf_{tpfn}, pd_{tpdn}, pi_{tpin}]^\top$
$\mathcal{X}_t$	Feasible set associated with decision x_t;
$\underline{\mathrm{F}}_f,(\overline{\mathrm{F}}_f)$	Minimum (maximum) power generation of thermal plant f;
$\underline{\mathrm{D}}_d,(\overline{\mathrm{D}}_d)$	Minimum (maximum) power deficit of the deficit level d;
$\underline{\mathrm{I}}_i,(\overline{\mathrm{I}}_i)$	Minimum (maximum) power capacity of the interconnection i;
$\underline{\mathrm{H}}_h,(\overline{\mathrm{H}}_h)$	Minimum (maximum) power generation of hydro plant h;
$\underline{\mathrm{Q}}_h,(\overline{\mathrm{Q}}_h)$	Minimum (maximum) turbined outflow of the hydro h;
$\underline{\mathrm{V}}_h,(\overline{\mathrm{V}}_h)$	Minimum (maximum) stored volume of the hydro h;
CF_f	The incremental cost of thermal power plant f;
CD_d	The incremental cost of the power deficit in deficit level d;
$\mathrm{CP}_{0,1,2,3}$	Hydro production function coefficients;
K_{tp}	Constant to convert flow to volume during stage t and period p;
TL_l	The time duration of power load level l;
u_{tpfn}, (up_{tpfn})	Binary variable equal to 1 whether the thermal plant f is online (offline) during stage t and period p; otherwise, it is equal to 0;
ud_{tpfn}	Binary variable taking the value of 1 if thermal plant f is turned off stage t and period p, or 0 otherwise;
UP_f, (DW_f)	Minimum uptime (downtime) of thermal plant f;
$\Delta\mathrm{UP}_f$, $(\Delta\mathrm{DW}_f)$	ramp-up (down) rate of thermal f.

CONTENTS

1 INTRODUCTION

The technical and economic sustainability of an electric power system is supported by the implementation and solution of the so-called Generation Scheduling (GS) problem. In a hydro-based power system, the GS problem aims at obtaining a generation policy that minimizes the expected cost of the thermal generation and some risk-measure, fulfilling several system constraints over a multi-year planning horizon. The GS is a challenging problem due to spatial and temporal coupling of hydro plants operation, hydro production function nonlinearities, and water inflow uncertainties. A well-known solution strategy is based on the decomposition of the GS problem (PEREIRA; PINTO, 1982) into a series of coupled problems:

i. a long-term GS (LTGS) problem with one to several years planning horizon, usually discretized in monthly steps;
ii. a medium-term GS (MTGS) problem for studies a few months ahead, with weekly time steps;
iii. a short-term (STGS) problem with a horizon composed of a few days, where the first day usually employs hourly time steps.

Usually, problem (ii) includes the opportunity cost of the water obtained in (i) and also determines this same cost for (iii). It should be noted that problems (i) and (ii) are eminently stochastic due to, in most cases, uncertainties on water inflows and market prices. Usually, uncertainties are represented via a scenario tree to yield a numerically tractable optimization problem. Further details and examples of the GS partitions (i)-(iii) can be found in (SHERKAT et al., 1985) (FOSSO et al., 1999) and (MACEIRA et al., 2002) for the Colombian, Norwegian and Brazilian cases, respectively. Essential details of the model hierarchy are also found in the STGS problem in (GIL; BUSTOS; RUDNICK, 2003) and the LTGS work presented in (ROTTING; GJELSVIK, 1992).

This thesis addresses the MTGS problem. Briefly, one of the most important purposes of the MTGS problem is to link the LTGS problem (pluriannual analysis) with the STGS problem (day-ahead decisions). To link the long and short-term problems efficiently, the MTGS problem requires considering, at the same time, an appropriate representation of the uncertainties and a reasonable detail level of the system. The MTGS modeling must carry out an individualized representation of the hydro plants, considering the water delay time between plants and the effect of head variation in the hydro production function (HPF).

Furthermore, thermal plants and the power demand must have detailed modeling compared to the one employed in the LTGS. Also, the time-step Δp (period discretization) can vary over the planning horizon. For the first period, Δp can be a fraction of a day or week; subsequently, Δp can be weekly or monthly. The scenario tree must attempt to capture the best possible representation of the uncertainties since the cost-to-go functions, which provide boundary conditions for the STGS problem, usually improve when the number of scenarios increases. Therefore, a tradeoff usually needs to be made between desired solution accuracy and computational performance.

The main features of the MTGS modeling, i.e., size of the planning horizon, time resolution, and scenario tree structure, depend considerably on the electricity market framework and power system physical characteristics (JOHANNESEN; FLATABØ, 1989). In particular, the Brazilian MTGS problem has the following characteristics (DINIZ et al., 2018):

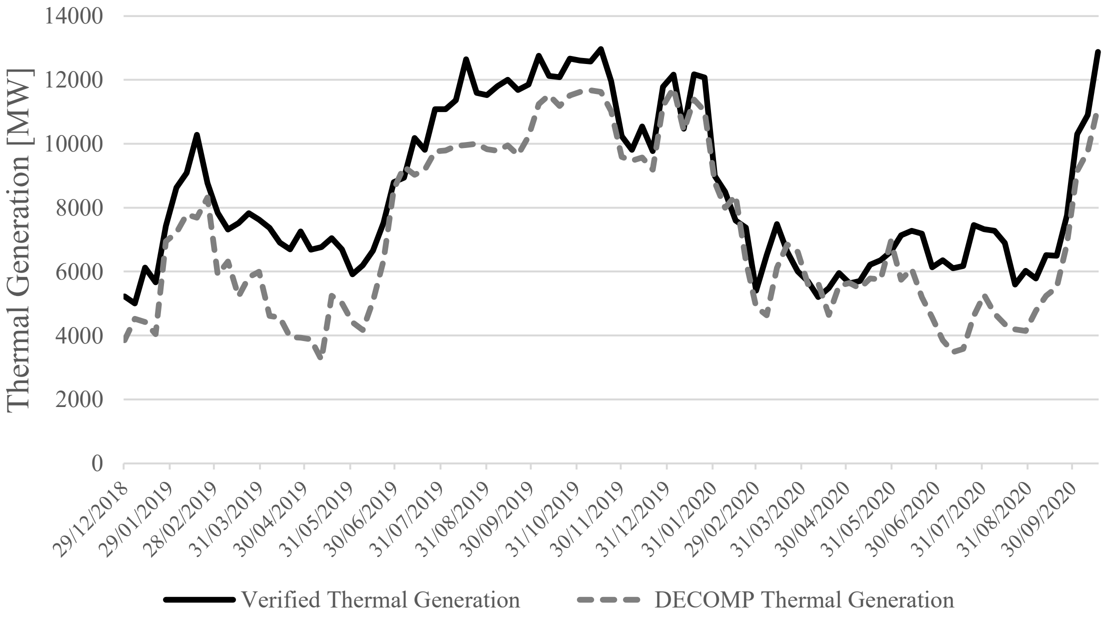
- the planning horizon can be up to 12 months. In practice, a planning horizon length of 2 months is considered;
- the maximum number of scenarios that can be considered is 12,000 scenarios. Depending on the season, between 100 and 500 scenarios are considered;
- deterministic weekly stages in the first month and a single month stage with finitely many inflow realizations;
- demand levels in all stages.

The use of a model that considers only the uncertainty in the second month can be justified by the following particularities: (*i*) it was necessary, two decades ago, to handle a detailed optimization model computationally with the inflows uncertainty; and (*ii*) a centralized dispatch of a large-scale power system, which is composed by approximately 130 thermal and 200 hydro plants. Nevertheless, improvements in modeling and solution techniques are possible due to more efficient optimization algorithms for stochastic programming and the current availability of powerful (mixed-integer) linear programming solvers.

Supported by these aspects, this work focuses on the MTGS optimization problem used in the Brazilian power system. In such a context, the model DECOMP (DINIZ et al., 2018) is the main computational tool to elaborate the operational guidelines for the next week of the system, as well as to define the energy price in the short-term market. DECOMP is a very sophisticated model. It has been developed by the Brazilian Electric Energy Research Center (CEPEL) for several decades, with constant improvements required by the modern Brazilian power industry. Nevertheless, the proposed contributions can be considered more generic, useful for markets with different dispatch models, and other problems using multi-stage stochastic linear programming (MSLP) models.

To further contextualize the Brazilian case, from 2001 to 2020, the DECOMP model was used by the Independent System Operator (ISO) to determine the week-ahead generation dispatch and the Energy Commercialization Chamber (ECC) to set the spot price for the same period. In practice, significant differences are found between the dispatch obtained by solving the MTGS problem and the real operation. Some of these differences are expected, given the dramatic distance between weekly and daily modeling details or non-planning dispatches that the ISO can carry out to attain security constraints. On the other hand, the poor representation of the inflow's randomness during the first month can trigger such disparities since the inflows present a high uncertainty (SOARES; STREET; VALLADÃO, 2017). For instance, Figure 1 shows the thermal production supplied by DECOMP and that one verified in real-time operation between the years 2019 and 2020.

Figure 1 – Real-time verified generation and Decomp thermal generation.



Source: Adapted from ("ONS - Operador Nacional do Sistema Elétrico", 2020a).

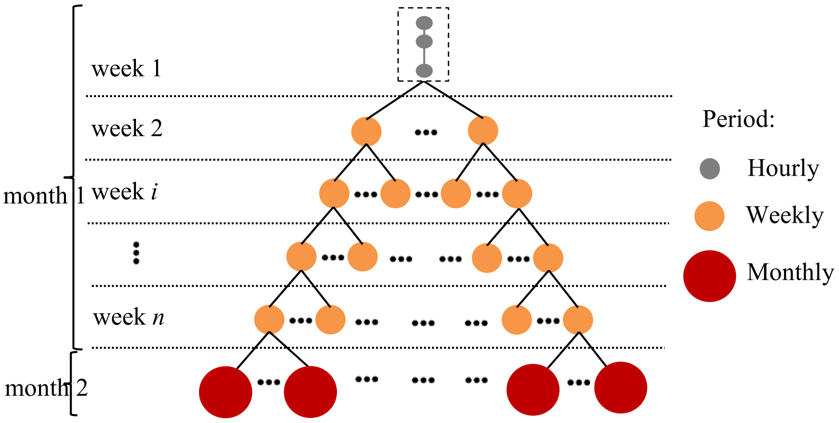
It is important to mention that the discrepancies shown in Figure 1 should decrease with the use of the DESSEM model (SANTOS et al., 2020) from 2021, which will be the principal computational tool for the Brazilian STGS problem. However, new challenges arise concerning the MTGS model for improving the coupling with the STGS problem, such as the insertion of wind power (considered known in the current model) and the system's regularization capacity reduction due to the priority construction of run-of-river hydro plants. In particular, such a reduction of regularization capacity highlights the importance of considering a better representation of uncertainties in the MTGS problem.

Based on this initial discussion, in terms of modeling issues, this work proposes an MTGS model with the following characteristics:

i. an hourly discretization in the first week to allow the inclusion of a chronological power load and the thermal unit commitment (UC);

ii. uncertainty is considered for all periods from the second week represented by a scenario tree with non-common sample realizations, as shown in Figure 2.

Figure 2 – Proposed model.



Source: Author.

To further exploit hydropower's flexibility, providing a more precise cost-to-go-function to the STGS problem, it becomes increasingly essential to capture parts of the operational characteristics described in (i) and (ii). In (i), however, nonconvexities arise from binary variables used to model minimum generation limits and unit commitments of thermal plants and are vital to capture generation units' capability. These nonconvexities should be represented in the STGS problem and in the MTGS model that provides the expected opportunity cost of water for the STGS.

Considering the uncertainty for all periods from the second week (ii) leads to the necessity of solving a large-scale problem. This fact demands a significant computational burden since the number of scenarios exponentially grows with the number of realizations per stage (SHAPIRO; NEMIROVSKI, 2005). Moreover, a more accurate first stage model requires extra variables (even with binary nature) and constraints, increasing the computational burden. Therefore, efficient solution techniques must be exploited to ensure an adequate tradeoff between desired accuracy and computational performance. In this work, algorithms with convergence guarantees and well-defined stopping criteria are considered.

1.1 LITERATURE REVIEW ON METHODS FOR TWO AND MULTI-STAGE STOCHASTIC PROGRAMMING

A well-known strategy to deal with large-scale problems is decomposition (SAGASTIZÁBAL, 2012). In this spirit, the Nested Benders Decomposition (NBD) (BIRGE, 1985) is a widely used solution methodology for MSLPs. The general procedure is given by a forward step for generating trial decisions (solving all nodes of the underlying scenario tree) and a backward one where these decisions are used to construct Benders-like cuts. The NBD is a cutting plane based-method (KELLEY, 1960) and (VAN SLYKE; WETS, 1969), which is well-known for non-monotonic characteristics and slow convergence. As a result, the NBD can also exhibit slow convergence, especially in high-dimensional problems. Several strategies have been proposed to overcome this issue. For instance, work (KOLOMVOS; SAHARIDIS, 2017) proposes a hybrid version of aggregated and disaggregated cuts, reporting significant savings in terms of CPU time. The work (VAN ACKOOIJ; DE OLIVEIRA; SONG, 2019) proposes regularizing the NBD by defining trial states as normal solutions, combining ideas from finite perturbation of convex programs and level bundle methods. In (BELTRÁN et al., 2020), cuts in the forward step are transported using ideas related to the definition of Chebyshev centers of certain polyhedrons. The Chebyshev approach obtained better optimality gaps in less CPU time applying in a large-scale multi-stage problem. These approaches are also applicable to the stochastic dual dynamic programming (SDDP) of (PEREIRA; PINTO, 1991), one of the main algorithms in multi-stage stochastic programming. The SDDP furnishes the NBD with sampling and cut-sharing approaches that make the method capable of handling huge-scale multi-stage stochastic problems. For instance, the SDDP is the solution method applied in the Brazilian long-term problem (MACEIRA et al., 2018). SDDP improvements can be found in (DINIZ et al., 2020) (DE MATOS; PHILPOTT; FINARDI, 2015). For further analysis, see (SHAPIRO, 2011).

Still, in the multi-stage setting, another important optimization technique is the progressive hedging (PH) algorithm of (ROCKAFELLAR; WETS, 1991). It is essential to mention that the PH does not construct an approximation of the cost-to-go functions. At least for the problems of interest, this fact appears as a drawback. This is one of the reasons why this thesis does not contemplate the PH.

In a two-stage setting, efficient methodologies combine the two-stage decomposition with bundle methods, e.g., (DE OLIVEIRA; SAGASTIZÁBAL; SCHEIMBERG, 2011) and (WOLF et al., 2014). In particular, (DE OLIVEIRA; SAGASTIZÁBAL; SCHEIMBERG, 2011) equips the two-stage decomposition with an inexact proximal bundle algorithm to reduce computational burden in computing an approximate solution. The works (FABIÁN, 2000) and (WOLF et al., 2014) investigate level bundle methods for solving convex two-stage stochastic programs up to optimality, even though inaccurate information is employed along the iterative process to speed up calculations. Level bundle methods were introduced in (LEMARÉCHAL; NEMIROVSKII; NESTEROV, 1995) for deterministic convex optimization, and some variants possess (nearly) dimension independent iteration complexity (BEN-TAL; NEMIROVSKI, 2005). An application of the level bundle method in (FINARDI et al., 2020) presented an excellent performance for solving a stochastic hydro-thermal unit problem. In the two-stage stochastic setting, the level bundle method is known as Level Decomposition (LD) (WOLF et al., 2014). The method's extension to (deterministic) convex mixed-integer problems was proposed in (DE OLIVEIRA, 2016) and denoted by extended level bundle method.

1.2 MTGS LITERATURE REVIEW

This section lists the main references on MTGS models and stochastic programs related to this work. It is important to highlight that there are MTGS models with different planning horizons and modeling details. Table 1 summarizes some characteristics of different MTGS problems found in the literature.

Table 1 – Main characteristics of different MTGS problems.

Problem	Characteristics
Norwegian (FOSSO et al., 1999) and (JOHANNESEN; FLATABØ, 1989)	- The planning horizon can be up to 18 months with weekly stages; - Uncertainty in the spot market price and inflows; - The inflow and market price scenarios are the same as those used by the LTGS problem.
Norwegian (HELSETH; FODSTAD; MO, 2016)	- A risk-neutral formulation to maximize a producer profit in the energy and capacity markets - A planning horizon of 24 months with weekly periods, in which inflows and market prices are stochastic.
Norwegian (HJELMELAND et al., 2018)	- Planning horizon composes of 52 weekly stages; - Uncertainty of inflow and energy price are considered; - Binary variables used to model minimum generation limits and unit commitments constraints are included; - Stochastic dual dynamic integer programming is proposed as a solution technique.
Canadian (CARPENTIER; GENDREAU; BASTIN, 2015)	- Planning horizon composed of 24 months with weekly stages; - Uncertain spot market price can be considered; - The seasonal cycle of the load is out of phase by several months with the natural inflows seasonal cycle; - The scenario tree considers a single inflow realization in intermediate stages (winter conditions).
Colombian (SHERKAT et al., 1988)	- The length of the planning horizon can be up to 12 months; - Weekly stages with a daily discretization for the first week and weekly for the remaining weeks; - Detailed modeling of the thermal and hydro plants in the first week; - A chronological load model in the first week.
New Zealander (PHILPOTT; PRITCHARD, 2013)	- The planning horizon is 12 months with weekly periods; - Weekly demand is represented by three demand levels; - Inflows are sampled from historical inflow realizations.
Spanish	- Planning horizon of 12 months and daily decisions of a single

Problem	Characteristics
(PÉREZ-DÍAZ et al., 2020)	hydro plant in an automatic frequency restoration reserve market.
Greek (OURANI; BASLIS; BAKIRTZIS, 2012)	- The planning horizon is 12 months with monthly periods; - Uncertainty of inflow is considered.
Chinese (LU et al., 2020)	- A risk analysis method for cascade hydro plants participating in a medium-term market considering uncertain market prices and settlement market rules; - The planning horizon is one month with daily periods.
Brazilian (DINIZ et al., 2018)	- The planning horizon can be up to 12 months. In practice, a planning horizon length of 2 months is considered; - The maximum number of scenarios that can be considered is 12,000 scenarios. Depending on the season, between 100 and 500 scenarios are considered; - Deterministic weekly stages in the first month and a single month stage with finitely many inflow realizations; - Demand levels in all stages.

Source: Author.

As shown in Table 1, most of the presented MTGS problems are formulated as an MSLP problem, with many decision stages and uncertainty of inflow and market prices. These (stochastic) models can also be developed for regulated or deregulated electricity markets, prioritizing distinct modeling aspects. For instance, the Norwegian MTGS problem considers an 18-month planning horizon without aggregation of the hydro plants (FOSSO et al., 1999), aiming to maximize a single producer's profit and consider that the spot market price is uncertain. The work (CARPENTIER; GENDREAU; BASTIN, 2012) proposes a weekly decision model with a planning horizon of 98 weeks for the Canadian MTGS problem, considering deterministic inflow value for the first 12 weeks. A nonconvex MTGS solved via Stochastic Dual Dynamic Integer Programming (SDDiP) is proposed in (HJELMELAND et al., 2018) for a reduced configuration of the Norwegian power system, which includes nonconvexities associated with the binary variables used to model minimum generation limits. Another instance is the work (OURANI; BASLIS; BAKIRTZIS, 2012), which presents the Greek MTGS problem with monthly decisions and a 12-month horizon where an inflow scenario tree is solved via SDDP.

Regarding the works related to the Brazilian MTGS problem, (PEREIRA; PINTO, 1982) proposes the representation of the decision process as a chain of hierarchical models with different degrees of detail, generalizing the main characteristics of the MTGS problem, i.e., input data, horizon, and relevant modeling aspects. Then (PEREIRA; PINTO, 1983) addresses the Brazilian MTGS problem and suggests the NBD algorithm as a solution technique. This methodology is generalized for problems with weekly and monthly decisions (PEREIRA; PINTO, 1985). In terms of modeling issues, initial contributions of the HPF are proposed in (CUNHA; PRADO; COSTA, 1997); however, with a strong assumption that the spillage is only present if the plant has reached its maximum turbined outflow. To overcome this issue, (DINIZ; MACEIRA; TERRY, 2004) constructs an HPF considering the spillage as an independent variable. Following this line, (XAVIER et al., 2005) and (DINIZ et al., 2008) include operating constraints in the HPF construction and a discretization of the volume and turbined outflow variables to improve the model accuracy. Regarding the stochastic process representation, significant contributions for generation and reduction of inflow scenario trees are found in (JARDIM; MACEIRA; FALCAO, 2001), (DA COSTA; DE OLIVEIRA; LEGEY, 2006) and (DE OLIVEIRA et al., 2010). Other proposal approaches with different solution methodologies for the Brazilian MTGS problem are explored in (DOS SANTOS et al., 2009). In such work, the PH algorithm is suggested as the solution method. The authors have concluded that, for simplified modeling of the MTGS problem, the PH algorithm with techniques of warm start and heuristics to adjust penalty parameters is competitive regarding the NBD algorithm. Inspired by these results, (GONÇALVES et al., 2011) carried out, for a more detailed MTGS problem and an inflow tree with 2401 scenarios, a comparison between the PH, NBD algorithms, and the solution of a deterministic equivalent problem. The results indicate that the computing time required by the PH algorithm is not higher than the NBD one. Finally, in this line, work (GONÇALVES; FINARDI; SILVA, 2012) proposes, using a detailed modeling, different decomposition schemes for the PH algorithm, obtaining significant time reductions compared with the classical PH one. However, in the works previously mentioned, both the PH and NBD exhibit an exponential computational burden growth as the scenarios number increases.

1.3 CONTRIBUTIONS

This work proposes an MTGS model for improving the coupling between the scheduling chain models. For that, the proposal considers two main modeling characteristics: (i) an hourly discretization in the first week, including the thermal UC constraints, and (ii) the inflow stochasticity from the second week of the planning horizon, through a scenario tree with non-common sample realizations. The main idea is to get the coupling stage modeling in the MTGS problem closer to the STGS modeling. In this context, the operational policies obtained iteratively in the MTGS solution are expected to be constructed in more interesting regions from the STGS view. Also, the policy robustness can be increased by considering a better stochastic process representation.

Motivated by the excellent performance of the LD for two-stage stochastic programs, this work handles the multi-stage stochastic MTGS problem by employing a two-stage-like decomposition, where every second-stage subproblem is indeed a multi-period program itself. The LD is thus applicable, and the numerical performance of the single and multi-cut LD variants are assessed. Our two-stage model has binary variables in the first week of the planning horizon. Every second-stage subproblem is a (set of) multi-period stochastic linear programs, having only continuous variables. The proposed two-stage decomposition is not an approximation of the underlying multi-period MTGS problem (in which a stage normally equals one period). The problem itself is written in an equivalent form, where both the first and the second stages comprise several periods. Such a two-stage decomposition is only possible with a moderate-size scenario tree, which is the case in this work. For that purpose, scenario reduction techniques can be employed, as already investigated in the MTGS setting in (BELTRÁN; DE OLIVEIRA; FINARDI, 2017). In our setting, although the computational burden per subproblem increases (w.r.t. the NBD), the number of iterations decreases considerably because tight cuts are obtained at each iteration. This kind of decomposition has presented promising improvements in different deterministic and stochastic problems found in the literature. For instance, the work (SANTOS; DINIZ, 2009) decomposes a deterministic STGS problem in several ways, and (DEMPSTER; THOMPSON, 1997) investigates, from a parallel computing point of view, the benefits of two-stage decompositions of MSLPs. Such benefits are also evidenced in a small size hydro-thermal scheduling problem in (ZAKERI; PHILPOTT; RYAN, 2000). Differently from the mentioned publications, our model possesses binary variables in the first stage, and so, this work uses the L-Shaped method and an extended variant of the level decomposition.

In addition to the use of bundle methods for handling MTGS problems, as a further contribution, we investigate a new NBD variant inspired by the central cutting-plane algorithm of (ELZINGA; MOORE, 1975). Specifically, trial state variables are obtained in the forward step as Chebyshev centers of certain polyhedral sets. To this end, the forward step requires only an extra parameter in the cut formulation of its LPs, preserving the dimension and nature of the NBDs subproblems. As a result, the computational burden to obtain trial policies in the forward step is not higher than the classical NBD. Our algorithm's backward step is precisely the one of the classical NBD, ensuring that valid lower bounds for the problem are computed. It turns out that such an approach does not always speed up calculations. Determining the conditions under which this methodology is promising is one of the work's contribution.

In summary, the work's first contribution is to reduce significantly the computational burden by applying a two-stage-like decomposition. Another contribution is to explore the single and multi-cut versions of the cost-to-go functions, aiming for computational gains. Finally, the extended level bundle method is applied to strengthen the convergence rate. These contributions outperform the NBD and SDDP algorithms, providing time reductions in the order of 19%. Further enhancements using Chebyshev centers in NBD and SDDP configure a concomitant contribution of this work.

The Thesis's main contributions have been split into two papers (BELTRÁN et al., 2020) and (BELTRÁN; FINARDI; DE OLIVEIRA, 2021), recently accepted for publication in *Optimization and Engineering* and *International Journal of Electrical Power and Energy Systems*, respectively.

1.4 STRUCTURE OF THE WORK

This work is organized as follows: Chapter 2 reviews some essential ingredients of stochastic programming, presenting the well-known two-stage and multi-stage formulations, as well as scenario tree structures and filtration concepts. The problem of interest is detailed in Chapter 3, which presents all the hydro-thermal system characteristics. Chapter 4 presents the proposed algorithms to solve the MTGS problem. In turn, Chapter 5 reports on the numerical performance of the considered algorithms applied to a real-life MTGS problem.

Finally, Chapter 6 closes the work with some concluding comments, remarks, and future directions of research.

2 STOCHASTIC PROGRAMMING: A BRIEF REVIEW

Several real-life decision problems are affected by randomness, for instance, energy planning (DE MATOS; MORTON; FINARDI, 2017); (GOEL; GROSSMANN, 2004) or financial problems (HOCHREITER; PFLUG, 2007); (DUPAČOVÁ, 2009). The planner has the difficult task of providing a solution, based on some criterion of optimality, making decisions before the uncertain data becomes known. Stochastic programming helps in obtaining a decision in problems involving uncertain data (BIRGE; LOUVEAUX, 2011).

At this point, it is natural to ask how good is a decision if randomness is disregarded. Deterministic models select only one value of the random variable and assume that it is known with certainty. The issue is that the resulting optimal decision is valid exclusively for this deterministic value selected. Note that the deterministic planning decision fails to hedge against other possibilities of the uncertain event that may occur in the future (INFANGER, 1994). Consider a hydro-thermal scheduling problem of a reduced power system composed of one hydro plant and four thermal plants with monthly decisions to illustrate this discussion. The aim is to find an optimal operation policy over a given planning horizon, which minimizes the expected cost associated with the thermal generation subject to load requirements and water balance constraints. The hydro plant data is shown in Table 2, given by the maximum values of turbined outflow q, volume v, and the productivity PR. The HPF is given by ph = PR q.

Table 2 – Hydro data.

$\overline{\mathbf{Q}}$ **(m^3/s)**	$\overline{\mathbf{V}}$ **(hm^3)**	**PR (MWa/(m^3/s))**
1,500	4,000	1.0

Source: Author.

The initial volume v_o is equal to 2,400 hm^3, and the maximum value of the spillage s is infinite. The minimum values of all hydro variables equal to zero. Likewise, Table 3 details the main thermal plant data, corresponding with the maximum value of thermal generation pf_f and the unit variable cost CF$_f$ of each thermal plant f. The minimum values of thermal generation are zero. Furthermore, the energy deficit pd, which represents the load shedding, has a unit variable cost of R$ 500/MWa.

Table 3 – Thermal data.

Plant *f*	$\bar{\mathrm{F}}$ **(MWa)**	**CF (R$/MWa)**
1	100	10
2	150	20
3	200	40
4	500	100

Source: Author.

Assume that this system is operated in a two-month planning horizon, and monthly decisions must be taken. The inflow to the first month is y_1 = 200 m^3/s, and the one for the second month $y_2(\xi)$ is unknown. Consider a constant K to convert a given flow rate in m^3/s to an equivalent volume (hm^3) in a month. This hydro-thermal system must meet a 1,000 MWa load each month.

The resulting LP problem with the deterministic approach is shown in (2.1) - (2.7), in which the vector $x = [pf_{1f}, pd_1, q_1, s_1, v_1, pf_{2f}, pd_2, q_2, s_2, v_2]^{\top}$ gathers all decision variables. For simplicity, the variables are presented only with the sub-indices for period p and thermal plant f: pf_{pf}, q_p, s_p, v_p.

$$\min_x \sum_{f=1}^{4} (\mathrm{CF}_f \cdot pf_{1f}) + 500 \cdot pd_1 + \sum_{f=1}^{4} (\mathrm{CF}_f \cdot pf_{2f}) + 500 \cdot pd_2 \tag{2.1}$$

s.t:

$$v_1 + 2.592 \cdot (q_1 + s_1) = 2{,}918.4 \tag{2.2}$$

$$\sum_{f=1}^{4} pf_{1f} + pd_1 + q_1 = 1{,}000 \tag{2.3}$$

$$v_1 \le 4{,}000,\ q_1 \le 1{,}500,\ s_1 \ge 0, pf_{1f} \le \bar{\mathrm{F}}_f, pd_1 \ge 0 \tag{2.4}$$

$$v_2 + 2.592 \cdot (q_2 + s_2) - v_1 = 2.592 \cdot y_2(\xi) \tag{2.5}$$

$$\sum_{f=1}^{4} pf_{2f} + pd_2 + q_2 = 1{,}000 \tag{2.6}$$

$$v_2 \le 4{,}000,\ q_2 \le 1{,}500,\ s_2 \ge 0, pf_{2f} \le \bar{\mathrm{F}}_f, pd_2 \ge 0 \tag{2.7}$$

The objective function (2.1) comprises the sum of the thermal generation and deficit costs associated with the monthly periods 1 and 2. Regarding the constraints, equations (2.2) and (2.5) represent the water balance, and the load requirements are represented by (2.3) and (2.6) for months 1 and 2, respectively. Finally, constraints (2.4) and (2.7) are the bounds of the variables.

For the deterministic case, the second-month inflow is considered known despite being future data. Suppose that the decision-maker estimates y_2 = 500 m^3/s. In this case, the operation cost is R\$ 5,481.48 and the respective solution is pf_{11} = 100 MWa, pf_{12} = 24.07 MWa, q_1 = 875.93 m^3/s, v_1 = 648, pf_{21} = 100 MWa, pf_{22} = 150 MWa and q_2 = 750 m^3/s. All other variables are zero in the solution. Imagine that the dispatch in the first month is applied, but in practice, y_2 = 350 m^3/s (30% less than the value used in the LP); in this condition, the operation cost increases to R\$ 11,481.5 (109% greater than the previous cost). To prevent this cost difference, the planner should decide an operation considering the uncertainties in y_2; for that, think of incorporating in a decision model two possible equiprobable scenarios of inflows, y_2^1 = 800 m^3/s (high inflow) and y_2^2 = 200 m^3/s (low inflow), the resulting stochastic linear programming (SLP) problem is shown in (2.8) - (2.17). The objective function has a term associated with each scenario, and the constraints for month two are duplicated. All variables associated with the second month have a superscript index for the scenarios, being $x = [pf_{1f}, pd_1, q_1, s_1, v_1, pf_{2f}^1, pd_2^1, q_2^1, s_2^1, v_2^1, pf_{2f}^2, pd_2^2, q_2^2, s_2^2, v_2^2]^{\top}$.

$$\min_x \sum_{f=1}^{4} (\mathrm{CF}_f \cdot pf_{1f} + 500 \cdot pd_1) + \frac{1}{2}\left[\left(\sum_{f=1}^{4} \mathrm{CF}_f \cdot pf_{2f}^1 + 500 \cdot pd_2^1\right) + \left(\sum_{f=1}^{4} \mathrm{CF}_f \cdot pf_{2f}^2 + 500 \cdot pd_2^2\right)\right] \tag{2.8}$$

s.t:

$$v_1 + 2.592 \cdot (q_1 + s_1) = 2{,}918.4 \tag{2.9}$$

$$\sum_{f=1}^{4} pf_{1f} + pd_1 + q_1 = 1{,}000 \tag{2.10}$$

$$v_1 \le 4{,}000,\ q_1 \le 1{,}500,\ s_1 \ge 0, pf_{1f} \le \overline{\mathrm{F}}_f, pd_1 \ge 0 \tag{2.11}$$

$$v_2^1 + 2.592 \cdot (q_2^1 + s_2^1) - v_1 = 2{,}073.6 \tag{2.12}$$

$$\sum_{f=1}^{4} pf_{2f}^1 + pd_2^1 + q_2^1 = 1{,}000 \tag{2.13}$$

$$v_2^1 \le 4{,}000,\ q_2^1 \le 1{,}500,\ s_2^1 \ge 0, pf_{2f}^1 \le \overline{\mathrm{F}}_f, pd_2^1 \ge 0 \tag{2.14}$$

$$v_2^2 + 2.592\cdot(q_2^2 + s_2^2) - v_1 = 518.4 \tag{2.15}$$

$$\sum_{f=1}^{4} pf_{2f}^2 + pd_2^2 + q_2^2 = 1{,}000 \tag{2.16}$$

$$v_2^2 \le 4{,}000,\ q_2^2 \le 1{,}500,\ s_2^2 \ge 0, pf_{2f}^2 \le \overline{\mathrm{F}}_f, pd_2^2 \ge 0 \tag{2.17}$$

The expected cost obtained from the stochastic approach is R$ 9,481.48 and the first stage decision is pf_{11} = 100 MWa, pf_{12} = 150 MWa, q_1 = 750 m^3/s, v_1 = 974.40 hm^3. Note that, compared with the deterministic approach, plant 2 operates at its maximum generation, and more water is kept in the reservoir. Using the stochastic decision in the first month and considering a hypothetical real realization of y_2 = 350 m^3/s, the operating cost is R$ 8,962.97, i.e., 22% cheaper than the total cost obtained from the deterministic approach. Therefore, the operation provided by the stochastic programming approach has a better behavior in this hypothetical (although possible) realization of the random variable. This behavior is expected because the first-month decision considers a better representation of uncertainty (in this case, low and high inflow values). The operation costs using the deterministic and stochastic first-month decision for different second-month inflow realizations y_2 are presented in Table 4.

Table 4 – Comparison between deterministic and stochastic first-month decision.

Inflow y_2 (m^3/s)	**Operational cost (R$)**		**Difference (%)**
	Using deterministic first-month decision	**Using stochastic first-month decision**	
150	28,482	18,407	-35.4
250	18,482	12,963	-29.9
350	11,482	8,963	-21.9
450	7,481	6,481	-13.4
550	4,481	4,741	5.8
650	2,481	4,000	61.2

Source: Author.

The stochastic first-month decision is cheaper than the deterministic one for all low inflow scenarios (less than 450 m3/s). More hydropower in the first month is stored, which is logically available in the second month. Therefore, more thermal power must be employed in the first month, which explains the higher cost regarding the one obtained with the deterministic first-month decision in the wet scenarios. This additional cost is insurance to protect the system against low future inflows. This safety is profitable compared to the damage associated with the risk of not meeting the load requirements. Notice that only two realizations of the random variable are used in the stochastic example: the first-month decision can be improved considering a better representation of the future inflows. The following references (KALL; WALLACE, 1994); (BIRGE; LOUVEAUX, 2011); (SHAPIRO; DENTCHEVA; RUSZCZYNSKI, 2009); (FINARDI; DECKER; DE MATOS, 2013) have other instances of the differences between the deterministic and stochastic approaches.

Since the solutions obtained from deterministic models are impractical, different techniques have been helping in decision making under uncertainty. The main ones are briefly introduced in the items below.

The worst-case approach or Robust Optimization (RO): uses the notion of uncertainty set, which reunites the adverse events which are not desired to influence the problem solution. RO is an optimization field in which, instead of scenarios, the uncertainty is specified via intervals. This approach is usually used when a high risk aversion level exists. RO does not require knowing the probability distribution of the underlying random variables, which is a desirable feature in some applications (but not in the one considered in this Thesis). More information about RO can be obtained in (BEN-TAL & NEMIROVSKI, 2000).

Chance-constraint Optimization (CCO): appears as an alternative to balance cost and robustness. In this approach, probabilistic constraints are included in the models. In generation scheduling problems, these constraints are usually associated with load balance requirements and hydro balance equations. CCO can be nonconvex and hard to evaluate, which may increase computing time. There is an interesting link between RO and CCO, which is to select an uncertainty set in such a way as to enforce a probabilistic constraint so that the solutions coming from the RO method are comparable with those coming from the CCO one. Overall, one may aim at replacing the probabilistic constraint with a convex one, albeit more restrictive. More information concerning CCO can be seen in (DENTCHEVA, 2009), (VAN ACKOOIJ et al., 2017), (VAN ACKOOIJ; DE OLIVEIRA, 2014) in energy problem context, and in (DELFINO, 2018) in a mixed-integer setting.

Stochastic Programming (SP) with recourse or Scenario Tree Approach: This approach has been the subject of intense research in the last two decades. The essential advantage of using scenario trees is that uncertainty is assumed to be known in each tree node, allowing SO to be solved as a large-scale equivalent deterministic problem discretized on the tree. Any technique used to solve deterministic problems might be used here. However, depending on the problem and if uncertainty is unknown for each tree node, there may be complicated issues related to determining the PDF of uncertain variables and generating the tree. These two subjects may be too complicated since they involve a wide range of techniques. The SP technique may be classified according to the number of stages that the problem is decomposed: two or multi-stage. Each stage comprises one or several periods, and uncertainties are revealed at the beginning of each stage.

This work is focused on SP with recourse. Section 2.1 reviews the basic theory for two-stage stochastic problems and, Section 2.2 deals with multi-stage programming problems, introducing the main aspects of scenario trees.

2.1 TWO-STAGE STOCHASTIC LINEAR PROGRAMS

SP's most well-known formulation is the two-stage linear programming (SHAPIRO; DENTCHEVA; RUSZCZYNSKI, 2009) and (BIRGE; LOUVEAUX, 2011). Mathematically expressed in (2.18) - (2.19), the first-stage variable $x_1 \in \mathbb{R}^{n_1}$ represents the decision vector that must be taken before realizing the random event ξ of the second stage. The decision variable's cost is contained in the n_1-dimensional vector c, and x_1 must belong to the viable set $\mathcal{X}:=\{x_1 \in \mathbb{R}^{n_1}$: $A\ x_1 = b$, $x_1 \geq 0\}$, where the m_1 x n_1 matrix A, the m_1-dimensional vector b and c are deterministic.

$$\begin{cases} \min\limits_{x_1 \in R^{n_1}} & c^{\top} x_1 \ \mathbb{E}\,\mathcal{Q} \qquad \xi \\ \text{s.t.} & Ax_1 = b, \ \ x_1 \geq 0, \end{cases} \tag{2.18}$$

where $\mathbb{E}$ is the expected value functional and $\mathcal{Q} \qquad \xi$ is the optimal value of the linear problem known as recourse function and associated with the second stage, shown in (2.19):

$$\mathcal{Q} \qquad \xi \quad : \begin{cases} \min\limits_{x_2 \in R^{n_2}} & q^{\top} x_2 \\ s.t. & Tx_1 + Wx_2 = h, \quad x_2 \geq 0. \end{cases} \tag{2.19}$$

Regarding the second stage problem, some or all elements of the vector $\xi=(q, h, T, W) \in \mathbb{R}^{n_\xi}$ can be random, defined on a probability space ***P*** with probability distribution P and supported in a sample space Ξ. Vector $x_2 \in \mathbb{R}^{n_2}$ represents the decision variable with the associated cost included in the n_2-dimensional vector q. The relation between the first and second stage variables is given by $Tx_1+Wx_2 = h$, where $T \in \mathbb{R}^{m_2 \mathrm{x} n_1}$ and $W \in \mathbb{R}^{m_2 \mathrm{x} n_2}$ are known as technological and recourse matrices, respectively. Finally, the m_2-dimensional vector h represents the "demand" or "request" value in the second stage.

In this general formulation, the random variable ξ is continuous with infinite Ξ. This fact hinders the calculation of the mathematical expectation $\mathbb{E}$ of $Q(x_1, \xi)$ and, therefore, a numerical solution in such a setting is only possible in rare cases. To overcome this issue in practice, the sample space Ξ is discretized, and a set S of realizations of the random variable is obtained. These realizations are denominated scenarios ξ^s, with an associated probability of occurrence p^s. The discretized version of the two-stage linear programming problem is shown in (2.20):

$$\begin{cases} \min\limits_{x_1 \in R^n} \quad c^\top x_1 + \sum\limits_{s=1}^{S} p^s Q(x_1, \xi^s) \\ \text{s.t.} \quad Ax_1 = b, \;\; x_1 \geq 0, \end{cases} \tag{2.20}$$

where for each realization $\xi^s = (q^s, h^s, T^s, W^s)$, the following subproblem must be solved:

$$\begin{cases} \min\limits_{x_2 \in R^n} \quad q^{s\top} x_2 \\ \text{s.t.} \quad T^s x_1 + W^s x_2 = h^s, \; x_2 \geq 0. \end{cases} \tag{2.21}$$

The dual form of (2.21) is as follow:

$$\begin{cases} \max\limits_{\lambda^s \in R^{m_2}} \lambda^{s\top}(h^s - T^s x_1) \\ \text{s.t.} \quad W^{s\top} \lambda^s \leq q^s, \end{cases} \tag{2.22}$$

where,

λ^s the dual variable associated with the second stage subproblem for the scenario ξ^s.

The value of S and the selection of the scenarios ξ^s is a relevant subject. The latter determines the quality of the discretized solution regarding the original stochastic process. However, a computational effort increases with the number of subproblems (2.21). The balance between the quality solution and the computational effort is one of the biggest challenges in stochastic programming. This topic is even more relevant in the multi-stage case due to the scenario number's exponential growth. This and more topics are discussed in the following section.

2.2 MULTI-STAGE STOCHASTIC LINEAR PROGRAMS

This work considers stochastic linear programming models to represent the MTGS problem. In this setting, a sequence of decisions $x_t \in \mathbb{R}^{n_t}$, evolves together with the uncertain data $\xi_t \in \mathbb{R}^{n_{\xi_t}}$ over time t, until the end of the planning horizon $T \geq 2$.

The sequence of the random vector ξ_t, with specified probability distribution P, defines the stochastic process $\xi{:=}\{\xi_t\}_{t=1}{}^T$. The data process's history up to time t is denoted $\xi_{[t]} = (\xi_1, \ldots, \xi_t)$. The decision vector must meet the requirement of nonanticipativity, i.e., a given x_t depends on the history $\xi_{[t]}$, but not on the future information. In this context, the sequence of decisions x_t also forms a stochastic process. Specifically, a feasible decision $x_t(\xi_{[t]}){:=}(x_1(\xi_1{}^s), \ldots, x_t(\xi_t{}^s))$, called implementable policy, is a function of the scenario $\xi^s{:=}(\xi_1{}^s, \ldots, \xi_T{}^s)$. Note that the history $\xi_{[t]}$ is a trajectory of the scenario until the stage t. Each scenario ξ^s of the stochastic process belongs to the sample space Ξ, which is part of the filtered probability space $\mathbb{P}{:=}(\Xi, \mathcal{F}, P)$, being $\mathcal{F}$ the filtration generated by the sequence of sigma algebras $\mathcal{F}_t$, for $t = 1, \ldots, T$, generated by $\Xi{:=}\Xi_1 \text{ x } ... \text{ x } \Xi_T$. Furthermore, the policy $x_t(\xi_{[t]})$ is $\mathcal{F}$-measurable if $x_t(\xi_{[t]}) \in \mathcal{F}$ and feasible if $x_t(\xi_{[t]}) \in \mathcal{X}_t(x_{t\text{-}1}, \xi_{[t]})$, the set-valued function $\mathcal{X}_t$: $\mathbb{R}^{n_{t\text{-}1}} \text{ x } \mathbb{R}^{n_{\xi_t}} \rightrightarrows \mathbb{R}^{n_t}$, for $t = 2, \ldots, T$.

An MSLP can be formulated as the nested expression (SHAPIRO; DENTCHEVA; RUSZCZYNSKI, 2009) Chapter 3, presented in (2.23):

$$\min_{\substack{x_1\geq 0\\ A_1x_1=b_1}} c_1^\top x_1 + \mathbb{E}\left[\min_{B_2x_1+A_2x_2=b_2} \quad \mathbb{E}\left[\quad \mathbb{E}\left[\min_{B_Tx_{T-1}+A_Tx_T=b_T} \quad c_T^\top x_T \right]\right]\right]. \tag{2.23}$$

In the above notation, vectors have the following dimensions: $c_t \in \mathbb{R}^{n_t}$, $b_t \in \mathbb{R}^{m_t}$, and the involve matrices A_t, and B_t have appropriate sizes. The feasible set in the first stage is $\mathcal{X}_1=\{x_1 \in \mathbb{R}^{n_1}: A_1\, x_1 = b_1,\, x_1 \geq 0\}$ and, for $t = 2, \ldots, T$, the form of a set-valued function is $\mathcal{X}_t(x_{t-1}, \xi_{[t]})=\{x_t \in \mathbb{R}^{n_t}: B_t\, x_{t-1}+A_t\, x_t = b_t,\, x_t \geq 0\}$. Some or all data of $\xi_t = (c_t, B_t, A_t, b_t)$ can be uncertain for $t = 2, \ldots, T$; the first stage is deterministic, that is, c_1, A_1 and b_1 are known.

To proceed with tractable numerical calculations, the support set Ξ must be finite containing finitely many scenarios ξ^s. With this assumption, the stochastic process has finite cardinality. The filtration $\mathcal{F}$ can be represented by a scenario tree as detailed in Section 2.2.1 and Section 2.2.2.

The multi-stage problem is computationally attractive because it induces a decomposition approach by stages. For that, the equation (2.23) can be represented by the following equivalent dynamic programming equations (BELLMAN, 2003):

(*i*) for the stages $t = 2, \ldots, T$:

$$Q_t(x_{t-1},\xi_{[t]}) := \begin{cases} \min\limits_{x_t\geq 0} \; c_t^\top x_t + \mathcal{Q} \qquad {}_{+1]}) \\ \quad \text{s.t.}\, A_t x_t = b_t - B_t x_{t-1}, \end{cases} \tag{2.24}$$

where the recourse function, also known as cost-to-go function $\mathcal{Q}$, for $t = 1,\ldots, T$-1, is the expected cost of the decisions taken in the future:

$$\mathcal{Q} \qquad {}_{+1]}) := \sum_{s=1}^{S} p^s Q_{t+1}(x_t, \xi^s_{[t+1]}), \tag{2.25}$$

being $\mathcal{Q} \qquad {}_{[T+1]}) \equiv 0$ by definition.

(*ii*) The first-stage problem is presented in (2.26), where $B_1 \equiv 0$ by definition:

$$\begin{cases} \min\limits_{x_t \geq 0} \quad c_1^{\top} x_1 + \mathcal{Q} \qquad\qquad) \\ \quad \text{s.t.}\, A_1 x_1 = b_1. \end{cases} \tag{2.26}$$

An implementable policy $x_t^{*}(\xi_{[t]})$, $t = 1, \ldots, T$, for problem (2.23) is optimal if x_1^{*} solves (2.26) and x_t^{*} solves (2.24). Such an optimal policy can be seen as a chain of decision rules for all horizon planning. See (BIRGE; LOUVEAUX, 2011) for details on optimality conditions.

In this work, a deficit variable in the hydro-thermal scheduling problem is included, detailed in Section 3.1.3.2, which ensures that every solution x_t that satisfies $A_t\, x_t = b_t$ has a feasible completion in the following stages[1]. In this case, the stochastic program has relatively complete recourse (BIRGE; LOUVEAUX, 2011). This condition implies that the cost-to-go functions $\mathcal{Q}$ are finite-valued. Besides, since the support Ξ is considered finite, the cost-to-go functions are convex piecewise linear functions (BIRGE; LOUVEAUX, 2011), Theorem 2. Therefore, these functions can be approximated by a collection of cutting planes, as described below:

$$\begin{cases} \min\limits_{x_t \geq 0} \quad c_t^{\top} x_t + \mathcal{Q} \qquad\qquad {}_{+1]}) \\ \quad \text{s.t.}\, A_t x_t = b_t - B_t x_{t-1}, \end{cases} \equiv \begin{cases} \min\limits_{x_t \geq 0, \theta_{t+1} \geq 0} \; c_t^{\top} x_t + \theta_{t+1} \\ \text{s.t.} \qquad A_t x_t = b_t - B_t x_{t-1}, \\ \qquad\quad \beta_{t+1}^{j\top} x_t + \alpha_{t+1}^{j} \leq \theta_{t+1}, \qquad \forall j \in \mathcal{J} \end{cases} \tag{2.27}$$

where,

$\mathcal{J}$: set of the cutting planes, $\mathcal{J} \qquad \in \mathbb{Z} \qquad nj_t\}$;

nj_t: number of cutting planes in stage t;

j: cut’s index;

θ: recourse function of the expected future cost;

$\beta_{t+1}^{c}, \alpha_{t+1}^{c}$: cut's coefficients.

[1] For cases in which x_t does not guarantee the subproblem’s feasibility of subsequent stages, we refer readers to (BIRGE; LOUVEAUX, 2011) Chapter 5 for further details on feasibility cuts.

The multi-cut version (BIRGE; LOUVEAUX, 1988) constructs one cut per each realization in t+1. In this way, the required number of iterations to achieve the convergence can be lower than the single cut approach (BIRGE; LOUVEAUX, 2011). The computational burden is increased since many more cuts are added to approximate $\mathcal{Q} \qquad {}_{+1]})$, as presented in (2.28):

$$\begin{cases} \min\limits_{x_t \geq 0, \theta_{t+1} \geq 0} \quad c_t^{\top} x_t \;\; \partial_{t+1,1} + \theta_{t+1,2} + \cdots \\ \text{s.t.} \quad A_t x_t = b_t - B_t x_{t-1}, \\ \qquad \beta_{t+1,1}^{j\top} x_t \;\; \alpha_{t+1,1}^{j} \leq \theta_{t+1,1}, \;\; \forall j \in \mathcal{J} \qquad : j \in \mathbb{Z} \qquad nj_{t,1}\}, \\ \qquad\qquad \vdots \\ \qquad \beta_{t+1,r}^{j\top} x_t \;\; \alpha_{t+1,r}^{j} \leq \theta_{t+1,r}, \;\; \forall j \in \mathcal{J} \qquad : j \in \mathbb{Z} \;\; \leq j \leq nj_{t,r}\}. \end{cases} \tag{2.28}$$

Different techniques to compute the cost-to-go functions are addressed in Chapter 4.

A review of the tree structure and filtration concepts to introduce the growth of information over time typical of stochastic programming problems is presented below.

2.2.1 Scenario tree

A scenario tree is a connected and undirected graph without cycles (ROSEN, 1999). The vertices of the graph $\mathcal{N}$ are known as node. The relation between the node $m \in \mathcal{N}$ and its predecessor node $n \in \mathcal{N}$ is indicated as $m \supset n$. The set of the successor nodes of m is denominated as m_{+}. The main properties of a scenario tree are presented below:

All nodes of the stage t belong to the set $\mathcal{N}_t = \{n\colon n \in \mathbb{Z}, 1 \leq n \leq nn_t\}$, such that, $\mathcal{N}$ is the union of the subsets $\mathcal{N}_1, \mathcal{N}_2, ..., \mathcal{N}_T$;

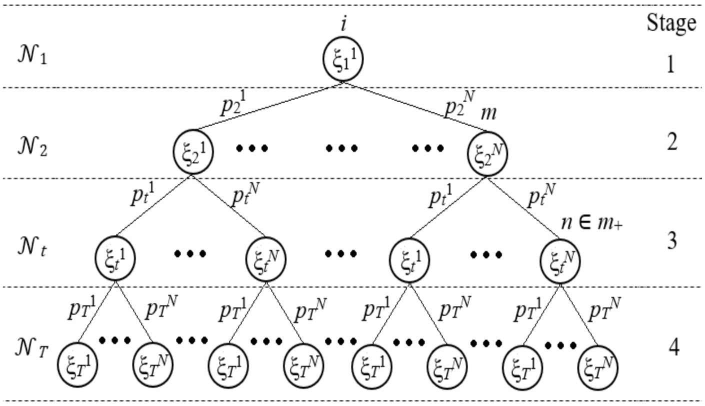
The predecessor node of all the nodes of the tree is denominated root node $r \supset m$ and, the set $\mathcal{N}_1$ contains only the node r;

The nodes $i \in \mathcal{N}_T$ are known as a leaf node, and the set i_{+} is empty;

The direct successor nodes of $n \in \mathcal{N}_t$ belong to the node set of stage t+1, i.e., $n_{+} \in \mathcal{N}_{t+1}$.

Figure 3 illustrates the characteristics of a scenario tree previously indicated. Each node of the tree in the stage t has associated a discretization of the sample space Ξ_t, called realization ξ_t^i with an associated probability of occurrence p_t^i (the subscript index indicates the corresponding stage and the superscript index represents the number of the discretization). The only node with deterministic realization is $r \in \mathcal{N}_1$. Observe that the node n is successor of the node m, i.e., n ∈ m+. Finally, there are no successor nodes in the last stage.

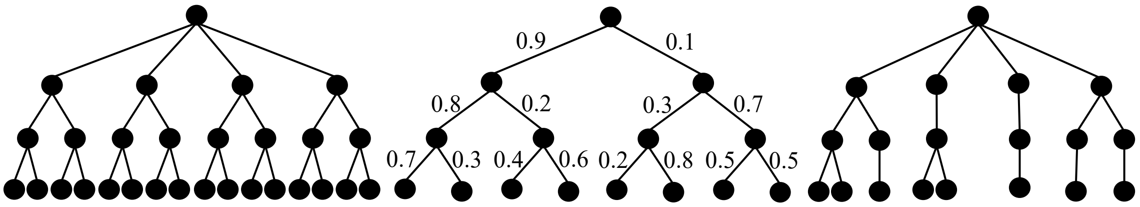
Figure 3 – Scenario tree structure.



Source: Author.

Scenario tree configurations can be summarized into two classes: symmetric and asymmetric scenario trees, see Figure 4. If all nodes of stage t have the same number of successors, for $t = 1, \ldots, T$-1, then the scenario tree has a symmetric configuration. Otherwise, the arrangement is asymmetric.

Figure 4 –Tree scenario configurations.



(*i*) Equiprobable and symmetric tree.

(*ii*) Symmetric tree with different realization probabilities.

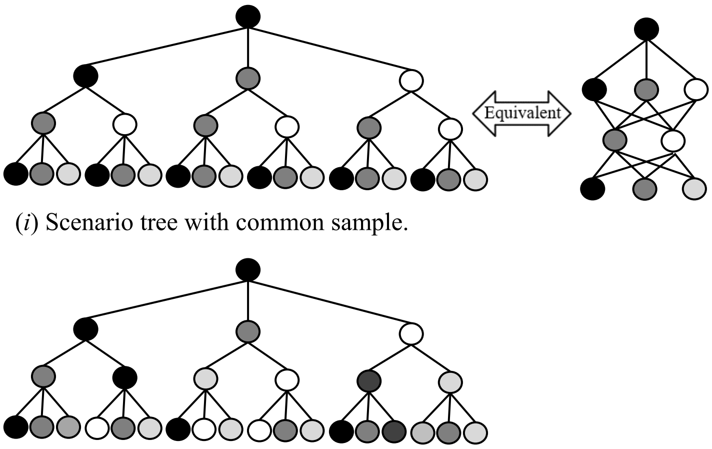
(*iii*) Asymmetric tree.

Source: Author.

A symmetric scenario tree is said to be of the common sample type if the children nodes n_+ are the same for every node $n \in \mathcal{N}_t$, $t = 1, \ldots, T$-1. This characteristic is widely used in sampling solution strategies since it allows sharing cuts between nodes of the same stage.

Figure 5 shows two equivalent ways of representing a multi-stage scenario tree with a common sample (top) and a scenario tree with a different sample (bottom).

Figure 5 – Different recombining scenario trees.



(i) Scenario tree with common sample.

(ii) Scenario tree with different sample.

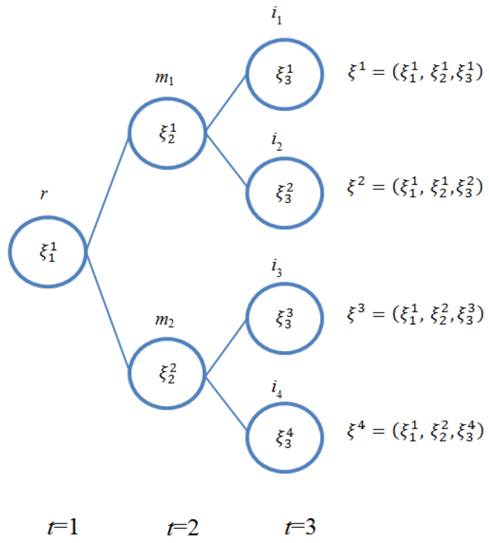
Source: Author.

The scenario tree's size is relevant since every tree node has associated a dynamic LP problem (2.24). The scenario tree's information has incremental behavior due to the random phenomenon's evolution over time. The computational effort required for the stochastic programming solution depends on the tree configuration. For instance, consider a scenario tree with 3 stages and 20 children nodes for each node, i.e., 1 x 20 x 20: 400 scenarios. Note that the increase of another stage (with 20 children nodes) gives a scenario tree with 8000 scenarios. Mathematically, this gradual increase of the information is modeled as a filtration (KOVACEVIC; PICHLER, 2015).

2.2.2 Filtration

A filtration $\mathcal{F}$ on the probability space (Ξ, $\mathcal{F}$, P) is the incremental family of sigma algebras $\mathcal{F}_t$, i.e., $\mathcal{F} := \{\mathcal{F}_t\}_{t=1}{}^{T}$; where each sigma-algebra is contained in the one generated in the next stage, that is $\mathcal{F}_{t\text{-}1} \subset \mathcal{F}_t$, for t = 1, …, T. To exemplify this concept, consider the scenario tree of Figure 6, which is composed by three stages and four scenarios. Each node of the tree has associated the realization $\xi_t^{\,i}$. The scenarios ξ^1, ξ^2, ξ^3, and ξ^4, represent the trajectory of the uncertain event from the root node r until the nodes of the last stages i_1, i_2, i_3, and i_4 respectively. Each scenario is composed of the realizations of the trajectory, i.e., $\xi^s=(\xi_1^s, \xi_2^s,..., \xi_T^s)$. All possible scenarios of the tree form the set $\Xi=\{\xi^1, \xi^2, \xi^3, \xi^4\}$, which is the sample space.

Figure 6 – Example of a scenario tree.


i_1
m_1
ξ_3^1
$\xi^1 = (\xi_1^1, \xi_2^1, \xi_3^1)$
ξ_2^1
i_2
r
ξ_3^2
$\xi^2 = (\xi_1^1, \xi_2^1, \xi_3^2)$
ξ_1^1
i_3
m_2
ξ_3^3
$\xi^3 = (\xi_1^1, \xi_2^2, \xi_3^3)$
ξ_2^2
i_4
ξ_3^4
$\xi^4 = (\xi_1^1, \xi_2^2, \xi_3^4)$
t=1
t=2
t=3

Source: Author.

Considering the tree above, a possible sigma-algebra for each stage t is presented below:

$\mathcal{F}_1=\{\{\xi^1, \xi^2, \xi^3, \xi^4\}, \emptyset\}$;

$\mathcal{F}_2=\{\{\xi^1, \xi^2\},\{\xi^3, \xi^4\},\{\xi^1, \xi^2, \xi^3, \xi^4\}, \emptyset\}$;

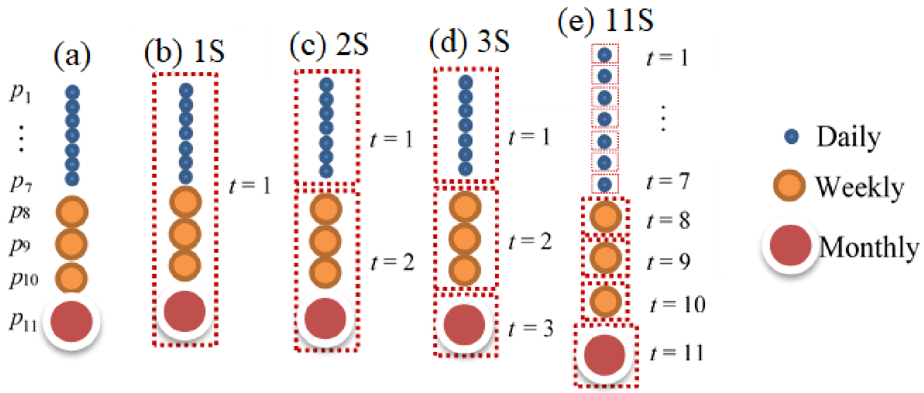
$\mathcal{F}_3=\{\{\xi^1\},\{\xi^2\},\{\xi^3\},\{\xi^4\},\{\xi^1,\xi^2\},\{\xi^1,\xi^3\},\{\xi^1,\xi^4\},\{\xi^2,\xi^3\},\{\xi^2, \quad \xi^4\}\{\xi^3, \quad \xi^4\},\{\xi^1,\xi^2,\xi^3\}, \{\xi^1,\xi^2,\xi^4\},\{\xi^1,\xi^3,\xi^4\},\{\xi^2,\xi^3,\xi^4\}\{\xi^1, \xi^2, \xi^3, \xi^4\}, \emptyset\}$;

Note that $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3$, this sequence is known as filtration $\mathcal{F}$ of the sigma algebras $\mathcal{F}_t$. This modeling of the information growth behavior is analogous to the tree structure. Therefore, it states that a scenario tree is a filtration, and any filtration induces a scenario tree as it is demonstrated in (KOVACEVIC; PICHLER, 2015).

2.3 DECOMPOSITION CONCEPTS

This section defines the concepts of stage and period used in this work based on (BIRGE; LOUVEAUX, 2011). A stage *t* is related to how the problem is decomposed, and a period *p* is the temporal basis (hourly, daily, weekly, monthly) in which the decisions are taken. The period duration is known as time-step Δ*p*. In this context, a stage can be composed of several periods. To illustrate this fact, assume a 2-month deterministic horizon discretized in the following way: the first week of the first month is divided into 7 days, the remaining part of the first month is divided into 3 weeks, and the last part of the horizon is given by a single month. This 2-month problem can be decomposed in different manners, as shown in Figure 7. We can see decompositions with 1, 2, 3, and 11 stages in (b), (c), (d), and (e), respectively.

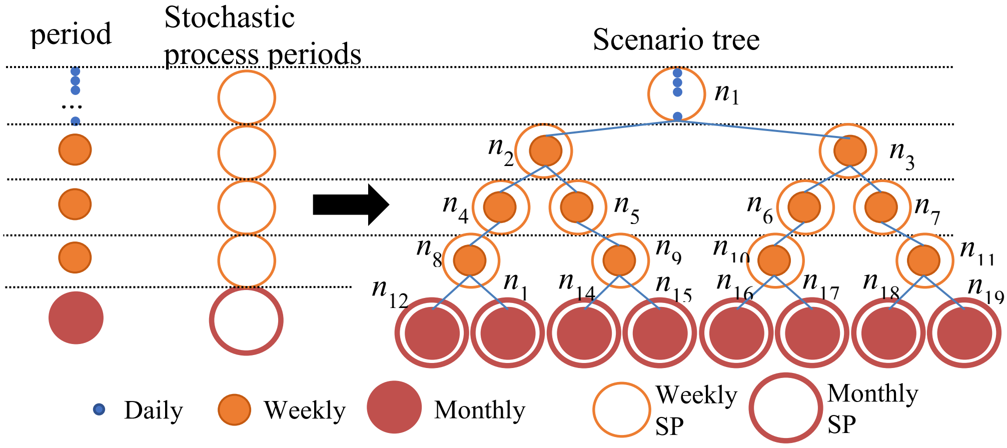
Figure 7 – Different decompositions.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Each decomposition yields different subproblems' size and, therefore, a different computational burden. For instance, the subproblem 1S in (b) is computationally more demanding than whichever subproblem of (e). On the other hand, in (b), it is only necessary (if possible) to solve a single problem to compute a solution. In 11S (e), it is necessary to solve 11 subproblems to obtain the cost-go-functions per iteration. This tradeoff is more relevant when a scenario tree is considered since the size and number of subproblems increase exponentially. Regarding a scenario tree, the temporal basis in which the stochastic process evolves is denominated stochastic process SP period. Figure 8 presents the relation between the periods of the problem and the SP periods. Note that the stochastic process can be composed of realizations with periods with different time-steps regarding the problem discretization. In the example, the stochastic process is composed of a weekly realization in the first week related to the problem's 7 daily initial periods. The scenario tree structure considers 1×2×2×1×2 number of realizations per SP period, i.e., 8 scenarios in a 2-month planning horizon. In the scenario tree, the final state of each node n is connected with the corresponding subsequent nodes. For instance, the final storage volume in n_1 is the initial volume for n_2 and n_3.

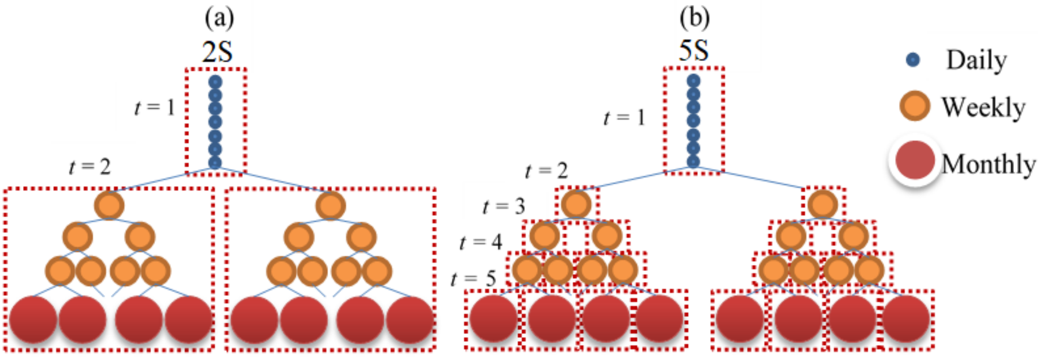
Figure 8 – Scenario tree structure.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Two kinds of decompositions of the resulting multi-stage problem are presented in Figure 9. On the left, we can see a classical two-stage decomposition; on the right, the picture presents a multi-stage decomposition.

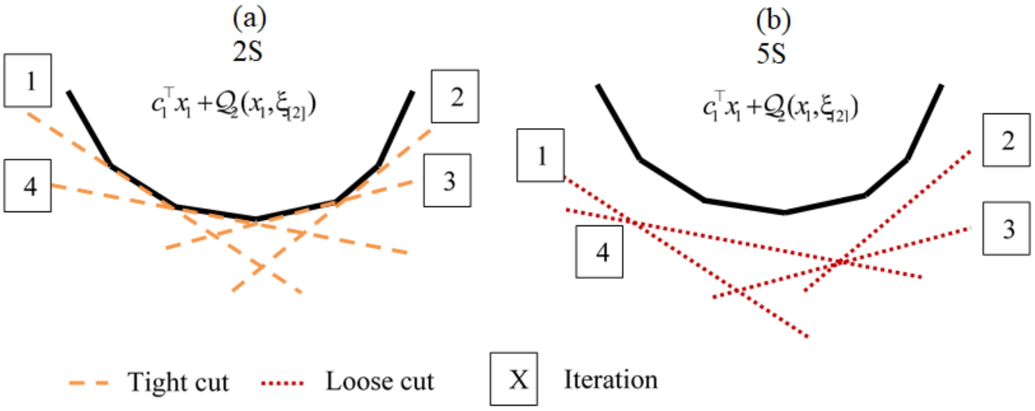
Figure 9 –Scenario tree configuration and decomposition.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

In addition to the number and size of the resulting subproblems presented in each dotted rectangle of Figure 9, the employed decomposition impacts the tightness of the cuts that approximate $\mathcal{Q}$ $_{+1]}$). For example, the 2S decomposition obtains tight cuts to approximate $\mathcal{Q}$); however, large-scale two-stage subproblems must be computed in each iteration. In turn, the 5S decomposition requires several iterations to reach the same cut accuracy to approximate $\mathcal{Q}$). Graphically, Figure 10 exemplifies the cuts obtained via 2S and 5S decompositions in an iterative process.

Figure 10 – Different decomposition methods give cuts of different qualities.


(a)
2S
$c_1^\top x_1 + \mathcal{Q}_2(x_1, \xi_{[21]})$
1
2
3
4
(b)
5S
$c_1^\top x_1 + \mathcal{Q}_2(x_1, \xi_{[21]})$
1
2
3
4
Tight cut
Loose cut
X
Iteration

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

The tight cut concept is one of the motivations to propose a two-stage decomposition for the MTGS problem. If the scenario tree has a moderate-size, the two-stage decompositions can be solved more quickly by the LD than the multi-stage decomposition by the NBD. The second motivation comes from the possible use of binary variables for modeling thermal unit-commitment constraints along the first week of the planning horizon: the cost-to-go functions are no longer convex in the presence of such complicating variables, precluding the application of the NBD if the decomposition is carried out per period. A decomposition strategy for multi-stage stochastic linear programs with binary variables has been recently proposed in (ZOU; AHMED; SUN, 2019), but its computational burden can be prohibitive for the size and properties of the scenario trees considered in this work (we do not assume stage-wise independence, a crucial assumption in the latter reference).

3 MEDIUM-TERM GENERATION SCHEDULING PROBLEM

The MTGS problem aims at obtaining a generation dispatch that minimizes a cost function associated with the thermal generation and deficit over a two to twelve-month planning horizon. This section describes the hydro-thermal system components, indicating, for a medium-term horizon, the primary considerations in modeling the hydro and thermal plants, electric subsystems, deficit, and load requirements. Finally, the proposed MTGS with hourly decisions in the first week and the inflow stochasticity representation from the second week on is presented.

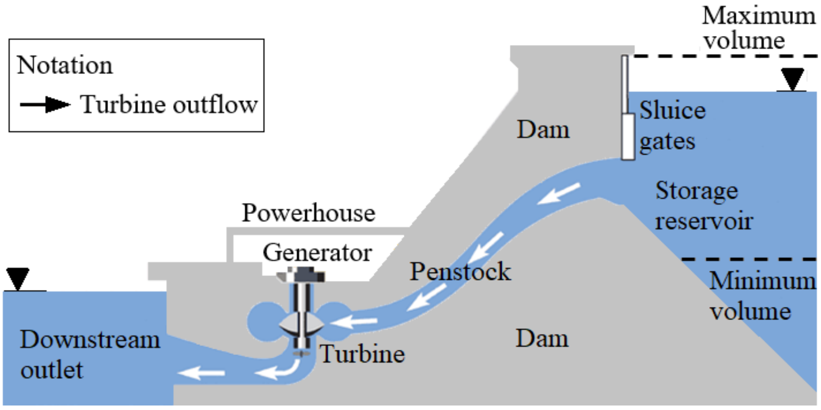
3.1 SYSTEM COMPONENTS

This section introduces the relevant components of a hydro-thermal system and the modeling formulation for the MTGS problem.

3.1.1 Hydro plants

The hydro plant's main characteristics in the optimization model are: (*i*) the reservoirs' behavior and (*ii*) the energy transformation process; through the water conservation constraint and the HPF. We are only interested in representing the reservoir, the spillway, and the powerhouse within all components of a hydro plant, such as control, production, and protection mechanisms. Figure 11 presents a basic diagram of a hydro plant.

Figure 11 – A hydro plant's diagram.



Source: Adapted from ("Hydro Dams for Large-Scale Electricity Supply | ClimateTechWiki", 2017)**.**

The hydropower production process is based on transforming potential energy into kinetic energy, associated with the storage volume and with the turbine movement, respectively. This process starts with a monitored opening of the sluice gates so that a water flow of the reservoir is channeled through the penstock until the powerhouse. In the powerhouse, the water flow velocity turns a turbine that is mechanically coupled with an electrical generator. In turn, the generator is connected to a power transformer in an electrical substation, and the power is injected into the electrical system. Finally, the water used in the hydropower production, known as turbine outflow, is returned to the river through the downstream outlet.

The HPF depends on the forebay and tailrace levels, the turbined outflow, the hydraulic losses in the penstock, and the turbine-generator efficiency, as presented in Section 3.1.1.2.

Another important hydro plant feature is the spillage, which is an unwanted condition but necessary action to guarantee the hydro plant safety. The spillage is a control mechanism to throw away excessive water in the reservoir, maintaining the maximum volume limit permitted. The spillage can increase the downstream level, decreasing power production.

Hydro plants can be classified according to the storage capacity over time. The first category is the hydro plants with a reservoir, having a huge dam to store and control large water volumes. The advantage is the capacity to accumulate water in wet periods to use in the dry season. The hydro plants with insufficient conditions to store water for long periods are known as run-of-river, producing energy if water inflows are available. The mathematical formulation of the HPF and the water balance in the MTGS model is detailed below:

3.1.1.1 Hydro Production Function

A hydro plant possesses a set of generating units that may not have the same operating characteristics (capacity, efficiency). However, the HPF for an MTGS problem is modeled considering an aggregated model, i.e., an approximating HPF per hydro plant is obtained.

The HPF is the mathematical expression that represents the output hydropower generated by a hydro plant as a function of the following variables: (*i*) turbined outflow, (*ii*) net head, defined as the difference between the forebay and tailrace levels, adjusted with the hydraulic losses and, finally, (*iii*) the efficiency of the units. Initially, the HPF considering an individual unit is expressed in (3.1) (DINIZ; MACEIRA, 2008):

$$pu_{pu}(hl_{pu},\eta_{pu},qu_{pu}) = \mathrm{GV}\cdot hl_{pu}\cdot\eta_{pu}\cdot qu_{pu}, \quad u\in\mathcal{U} \qquad \mathcal{H} \tag{3.1}$$

where,

$\mathcal{H}$: set of hydro plants in operation during period *p*, $\mathcal{H} \qquad \imath\in\mathbb{Z} \qquad \imath h_p\}$;

nh_p:

nh_p: number of hydro plants operating in period *p*;

$\mathcal{U}$: set of the units *u* in operation during period *p*, that belong to the hydro plant *h*, $\mathcal{U} \qquad \imath\in\mathbb{Z} \qquad nu_{ph}\}$;

nu_{ph}: number of the units in operation during period *p* that belong to the hydro plant *h*;

pu_{pu}: generating of unit *u* during period *p* (MW);

GV: constant defined by 0.00981(MW/(m^3/s)·m);

hl_{pu}: net head of generating unit *u* during period *p* (m);

η_{pu}: the efficiency of the generating unit *u*, during period *p*;

qu_{pu}: turbined outflow of unit *u*, during period *p* (m^3/s).

The HPF per hydro plant is defined below:

$$ph_{ph}(pu_{pu}) = \sum_{u\in\mathcal{U}} pu_{pu}, \;\; h\in\mathcal{H} \tag{3.2}$$

where,

ph_{ph}: generation of hydro *h* during period *p* (MW).

This work considers that the hydro plant efficiency η_{ph} equals to the efficiency average of the units $u\in\mathcal{U}$. Also, an average penstock head loss of the units *u* is considered for a hydro plant. The turbined outflow in the hydro plant *h* can be expressed by $q_{ph} = nu_{ph}$

qu_{pu}. Taking these simplifications into account and replacing (3.1) in (3.2), the resulting HPF per hydro plant is shown in (3.3):

$$ph_{ph}(hl_{ph}, \eta_{ph}, q_{ph}) = \mathrm{GV} \cdot hl_{ph} \cdot \eta_{ph} \cdot q_{ph}, \tag{3.3}$$

where,

q_{ph}: turbined outflow of hydro h, during period p (m^3/s).

hl_{ph}: net head of hydro h during period p (m);

The ph_{ph} limits are defined in (3.4):

$$\underline{\mathrm{H}}_h \leq ph_{ph} \leq \overline{\mathrm{H}}_h, \qquad \forall h \in \mathcal{H} \tag{3.4}$$

where,

$\underline{\mathrm{H}}_h, \overline{\mathrm{H}}_h$: minimum (max) generation of the hydro h (MW).

The HPF (3.3) is nonlinear, nonconvex, and discontinuous (QUINTERO, 2013) due to the characteristics of the functions hl_{ph} and η_{ph}.

At first, the net head (m) in the reservoir is given by:

$$hl_{ph}(h_{up}, h_{dw}, h_{loss}) = h_{up} - h_{dw} - h_{loss}. \tag{3.5}$$

The functions h_{up}, h_{dw}, and h_{loss} are the forebay level, the tailrace level, and the penstock head losses. These functions can be described; as they are in the Brazilian case, by:

$$h_{up}(v_{ph}) = \mathrm{HF}_0 + \mathrm{HF}_1 v_{ph} + \mathrm{HF}_2 (v_{ph})^2 + \mathrm{HF}_3 (v_{ph})^3 + \mathrm{HF}_4 (vh_{ph})^4, \tag{3.6}$$

$$\begin{aligned} h_{dw}(q_{ph}, sh_{ph}) = \mathrm{HT}_0 + \mathrm{HT}_1 (q_{ph} + sh_{ph}) + \mathrm{HT}_2 (q_{ph} + sh_{ph})^2 + \\ + \mathrm{HT}_3 (q_{ph} + sh_{ph})^3 + \mathrm{HT}_4 (q_{ph} + sh_{ph})^4, \end{aligned} \tag{3.7}$$

$$h_{loss}(q_{ph}) = \mathrm{HP} \cdot (q_{ph})^2. \tag{3.8}$$

where,

v_{ph}:	volume of hydro h, during period p (hm^3);
sh_{ph}:	spillage of hydro h, during period p (m^3/s).

The constant coefficients HF$_0$, …, HF $_4$ and HT$_0$, …, HT$_4$ in (3.6) and (3.7), are particular of each hydro plant. Regarding the equation (3.8), the hydraulic losses are represented by the constant-coefficient HP (s^2/m^2).

On the other hand, the equation (3.9) models the plant efficiency, where CE$_0$, …, CE $_5$ are constants.

$$\begin{aligned}\eta_{ph}(q_{ph}, hl_{ph}) = \mathrm{CE}_0 + \mathrm{CE}_1 q_{ph} + \mathrm{CE}_2 (q_{ph})^2 + \mathrm{CE}_3 hl_{ph} + \\ + \mathrm{CE}_4 (hl_{ph})^2 + \mathrm{CE}_5 (q_{ph} \cdot hl_{ph})^2.\end{aligned} \tag{3.9}$$

Although the HPF (3.3) is an approximation, its application is still not viable in the modeling of large-scale problems due to the nonlinearities shown in (3.5) and (3.9). Therefore, a linearization process of the HPF is carried out[2], obtaining a piecewise linear model.

This work considers a problem with 152 hydro plants. In the 25 hydro plants which represent the 70% of the hydro capacity, a concave piecewise HPF (3.10) with 3 hyperplanes per plant is considered. This work applies the piecewise linear model for the HPF in (MUHLEN, 2019), based on minimizing the error sum of squares w.r.t the nonlinear HPF. For other plants, constant productivity modeling (3.11) is assumed.

$$ph_{ph} \geq \mathrm{CP}_{0g} + \mathrm{CP}_{1g} v_{ph} + \mathrm{CP}_{2g} q_{ph} + \mathrm{CP}_{3g} sh_{ph}, \quad g \in \mathcal{G} \tag{3.10}$$

$$ph_{ph} = \mathrm{CP}_{2g} q_{ph}. \tag{3.11}$$

where,

[2] We refer readers to (DINIZ; MACEIRA, 2008); (QUINTERO, 2013); (FREDO; FINARDI; DE MATOS, 2019).

$\underline{H}_h, \overline{H}_h$: set of the constraints that belong to the HPF of hydro h;

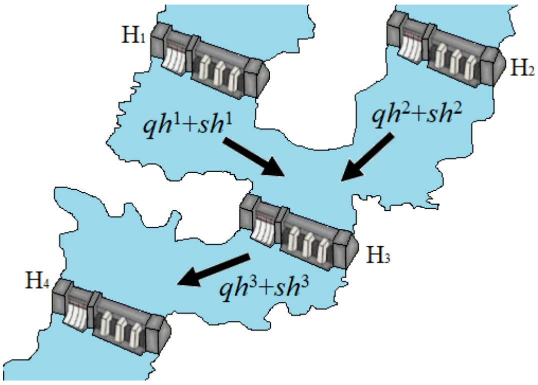
g: constraint g belongs to $\mathcal{G}$;

$CP_{0g},\ldots, Cp_{3g}$: constants related to the constraint g.

3.1.1.2 Hydro balance equation

The hydro balance equation establishes equality between the water inflows and the water outflows in a reservoir, based on the water conservation law. For instance, for a hydro plant with a reservoir, the final storage volume equals the sum of the initial volume and the inflows minus the turbined outflow and the spillage. In real-life systems, several hydro plants can be in a cascaded configuration. The turbined outflow and the spillage of a hydro plant directly impact the amount of water that arrives at the downhill plant. For instance, consider the cascaded configuration shown in Figure 12. Note that the turbined outflow and spillage in H_1 and H_2 are inputs for the H_3. In turn, the outflow of H_3 arrives in H_4. Also, a water delay time between consecutive hydro plants is usually considered in the models when such delay is longer than Δp.

Figure 12 – Cascade configuration.



Source: Author.

The water balance equation for a hydro plant with a reservoir and a run-of-river plant, disregarding the effects of evaporation and infiltration, are presented below in (3.12) and (3.13), respectively:

$$v_{p-1,h} - v_{ph} + \mathrm{K}_p \left[y_{ph} - q_{ph} - sh_{ph} \right] + \sum_{m \in \mathcal{M}} \mathrm{K}_\tau (q_{\tau m} + sh_{\tau m}) = 0, \quad h \in \mathcal{V} \tag{3.12}$$

$$\mathrm{K}_p \left[y_{ph} - q_{ph} - sh_{ph} \right] + \sum_{m \in \mathcal{M}} \mathrm{K}_\tau (q_{\tau m} + sh_{\tau m}) = 0, \quad h \in \mathcal{R} \tag{3.13}$$

where,

$\mathcal{V} \quad \mathcal{H}$: set of the hydro plants with a reservoir in operation during period p;

$\mathcal{R} \quad \mathcal{H}$: set of the run-of-river plants in operation during period p;

$\mathcal{M} \quad \mathcal{H}$: set of the hydro plants m upstream of the hydro h;

y_{ph}: incremental inflow to hydro h, during period p;

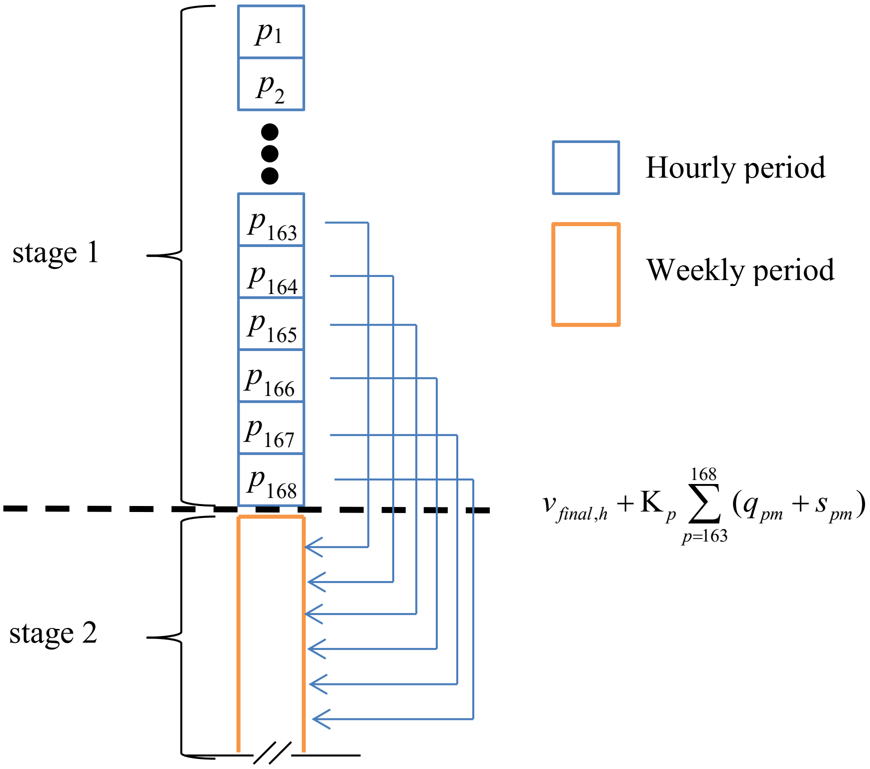
K_p: constant to convert a given flow rate in m^3/s in an equivalent volume in hm^3, during period p;

τ: period in which the upstream outflows of hydro h are decided, $\tau = p - \tau_{mh}$;

τ_{mh}: water delay time between hydro m and hydro h.

This work disregards the water delay time when period p is longer than τ_{mh}. In this case, the outflow of a hydro plant m upstream in period p is available to the hydro h in the same period. Otherwise, τ_{hm} influences the HB equations in (3.12) and (3.13). Another simplification considered in this work consists of including the delayed outflows available in a subsequent stage t+1, at the end of stage t, following the approach in (DINIZ; SOUZA, 2014). Figure 13 presents an example of the first stage with an hourly discretization and a second stage with weekly time steps. In this example, the outflows arrive from hydro m to hydro h in τ_{mh} = 6 hours. In this case, the hydro m outflows of periods p_{163}, p_{164}, p_{165}, p_{166}, p_{167}, and p_{168} are considered in the first-stage cost-to-go functions to carry out a valuation of the water.

Figure 13 – Approach of outflows with the water delay time between stages.



Source: Author.

Finally, the bounds of the volume, the turbined outflow, and the spillage are presented below:

$$\underline{\mathrm{V}}_h \leq v_{ph} \leq \overline{\mathrm{V}}_h, \qquad \forall h \in \mathcal{H} \tag{3.14}$$

$$\underline{\mathrm{Q}}_h \leq q_{ph} \leq \overline{\mathrm{Q}}_h, \qquad \forall h \in \mathcal{H} \tag{3.15}$$

$$0 \leq sh_{ph}, \qquad \forall h \in \mathcal{H} \tag{3.16}$$

where,

$\underline{\mathrm{V}}_h, \overline{\mathrm{V}}_h$: minimum (max.) volume of the hydro *h* (hm^3);

$\underline{\mathrm{Q}}_h, \overline{\mathrm{Q}}_h$: minimum (max.) turbined outflow of the hydro *h* (m^3/s).

3.1.2 Thermal plants

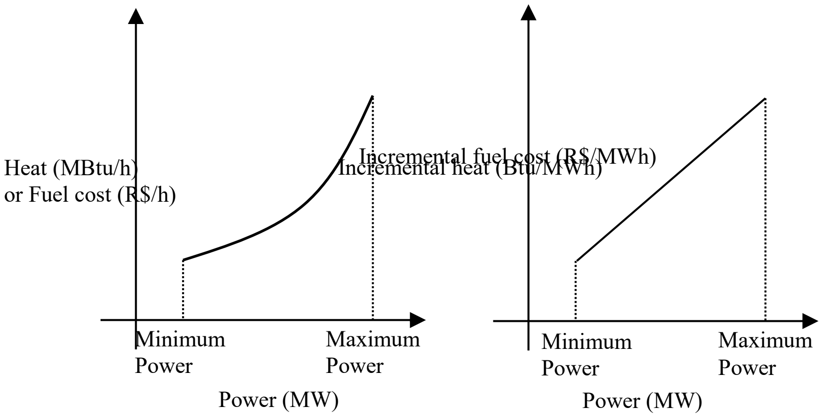
The thermal plants produce electrical power from chemical or nuclear energy of certain elements known as fuels. The operation is based on the generation of saturated water vapor, using the heat produced with some fuel as the primary force for a turbine, which gives movement to the electric generator rotor (FLYNN, 2003). Depending on the fuel, there are three groups:

i. thermal plants that use fossil fuels, such as coal, fuel oil, or natural gas;
ii. thermal plants that use fissile material as uranium or plutonium, which release heat energy through nuclear fission;
iii. thermal plants that use organic material, animal, or plant origin, such as biomass.

In the context of the MTGS problem, the thermal plant modeling is carried out considering two principal aspects: (*i*) the costs associated with the generation process and (*ii*) the operational constraints related to technical limits.

Thermal plant costs are directly related to the costs of the fuels used in the generation process. The relation between the fuel input and the electrical power output can be represented as a quadratic function (WOOD; WOLLENBERG, 1984). A typical input-output curve of a thermal plant and the incremental cost characteristic is shown in Figure 14. The input is expressed as the heat energy requirements in (MBtu/h) or, multiplied by the fuel cost (R$/MBtu), the equivalent total fuel cost per hour (R$/h) is obtained.

Figure 14 – Input-output curve of a thermal plant and the incremental heat (cost) characteristic.



Source: Adapted from (WOOD; WOLLENBERG, 1984).

Mathematically, the input-output function in (R\$/h) for a thermal unit $f \in \mathcal{F}_p$ in period p is defined below:

$$Fc_{pf}(pf_{pf}) = \mathrm{FC}_{0,f} + \mathrm{FC}_{1,f}\, pf_{pf} + \mathrm{FC}_{2,f}(pf_{pf})^2, \tag{3.17}$$

where,

$\mathcal{F}_p$:	set of thermal plants in operation during period p, $\mathcal{F}_p = \{f\text{: } f \in \mathbb{Z}, 1 \leq f \leq nf_p\}$;
nf_p:	number of thermal plants operating in period p;
pf_{pf}:	generation of thermal f during period p (MWh);
Fc_{pf}:	cost function of thermal f during period p (R\$/h);
$\mathrm{FC}_{0:2,f}$	constants related to the cost function.

Equation (3.17) must be linearized to enable its implementation in large-scale problems, obtaining an approximated cost function for each thermal plant f. The linear relation between the generated thermal power with the total cost is given by the term known as incremental cost CF_f, shown in (3.18):

$$Fa_{pf}(pf_{pf}) = \mathrm{CF}_f \cdot pf_{pf}, \tag{3.18}$$

where,

Fa_{pf}:	the approximation cost function of thermal f during period p (R\$/h);
CF_f:	the incremental cost of thermal f (R\$/MWh).

Additionally, the generation limits of thermal f during period p are represented by the constraint in (3.19):

$$\underline{\mathrm{F}}^f \leq pf_{pf} \leq \overline{\mathrm{F}}^f, \qquad \forall f \in \mathcal{F} \tag{3.19}$$

where,

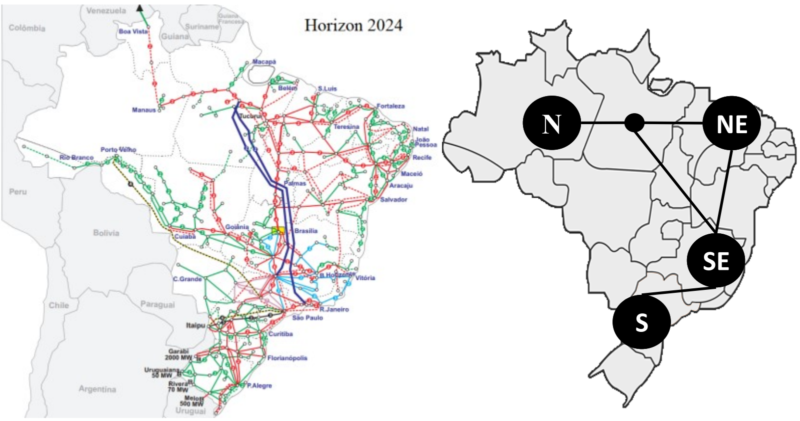
$\underline{\mathrm{F}}_f, \overline{\mathrm{F}}_f$: minimum (max) generation of thermal plant f (MW).

3.1.3 Electrical subsystems

One of the main characteristics of an electrical system is the incapacity to store sizeable electrical energy quantities. The constant balance between the electrical generation and the consumption (load) must be guaranteed. The generation plants and the consumer centers typically are in different geographic positions, requiring a transmission system composed of transmission lines and electrical substations (bars) connected in a redundant form to increase the reliability system.

In the context of the MTGS problem, the equations that represent the transmission system are simplified, and the total number of system bars considered is significantly reduced. Figure 15 shows a representation of the Brazilian electrical system divided into four equivalent subsystems: north (N), northeast (NE), southeast (SE), and south (S).

Figure 15 – The Brazilian transmission system and equivalent subsystems.



Source: Adapted from (“ONS - Operador Nacional do Sistema Elétrico”, 2020b).

The set of the subsystems s considered in the model is defined as:

$$\mathcal{S} \qquad \in \mathbb{Z} \qquad ıs\}, \tag{3.20}$$

where,

ns: the number of subsystems.

The modeling aims at satisfying the load requirements of each subsystem with its generation and the possibility of power interchange from other subsystems. An input power represents power interchanges in the importing subsystem and output power in the exporting subsystem. The set of power interchanges is defined below:

$$\mathcal{I} \qquad \mathbb{Z} \qquad i\}, \tag{3.21}$$

where,

ni: the number of power interchanges.

The quantity of power interchange must satisfy the limits directly related to the transmission capacity that belongs to a subsystem *s*. In this way, the limits of the interchange *i* during period *p* are represented by the constraint (3.22):

$$\underline{\mathrm{I}}_i \le pi_{pi} \le \bar{\mathrm{I}}_i, \quad i \in \mathcal{I} \tag{3.22}$$

where,

pi_{pi}: power interchange *i* during period *p* (MW);

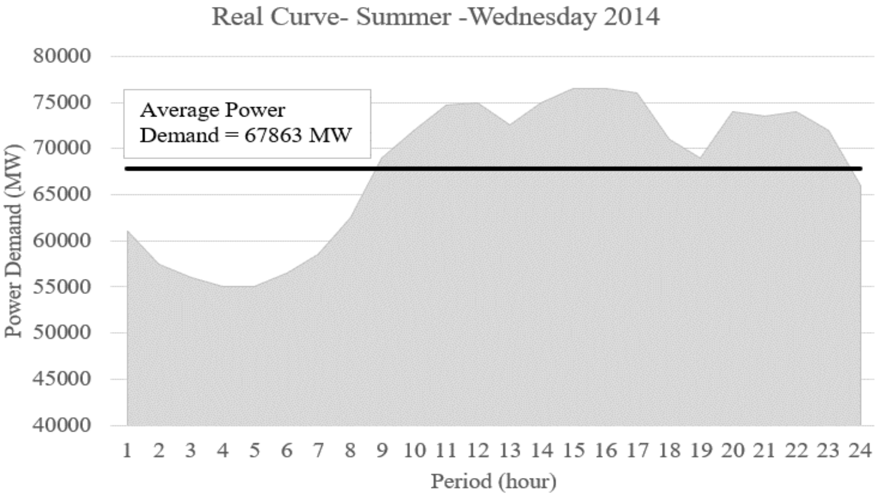
$\underline{\mathrm{I}}_i, \bar{\mathrm{I}}_i$: minimum (max) power capacity of the interconnection *i* (MW).

Regarding the load requirements, note that this is an important parameter since it determines the amount of power required to keep the balance generation-load during all horizon planning in each subsystem *s*.

3.1.3.1 Power demand

The power demand is directly related to the consumer characteristics that can be broadly classified as residential, commercial, or industrial. The requirements of each type include the total consumption of electricity over a period *p*. For that, in planning decision models, the average power demanded during a period *p* is calculated. For instance, consider a typical power demand curve in Brazil for 24 hours and an average consumption of the daily stage, as shown in Figure 16.

Figure 16 – Typical power demand curve.



Source: Adapted from (EPE, 2015).

In this work, the power demand pl_{ps} is considered deterministic for all periods and subsystems. The power demand is defined as the sum of the power demand and the estimated internal transmission losses, discounting the power generation produced from the small hydro plants and wind plants of all the buses representing in the subsystem *s*.

3.1.3.2 Deficit

Hydrothermal systems with a thermal capacity lower than the load requirements cannot guarantee a 100% uninterrupted service. In this case, in dry periods, the stored energy added to the thermal generation is insufficient to satisfy the load requirements, resulting in a power deficit.

The deficit is modeled through a fictitious thermal plant with the capacity to meet all the load. This assumption makes the MTGS model with a relatively complete recourse, as explained in Section 2.2. The deficit variable must be associated with a cost that estimates the social damage of a non-satisfying load. For instance, different deficit levels are considered in the Brazilian model (DINIZ et al., 2018). The set of the deficit levels *d* for stage *t* is defined in (3.23):

$$\mathcal{D} \qquad d \in \mathbb{Z} \qquad nd_p\}, \tag{3.23}$$

where,

nd_p: number of deficit levels in period *p*.

The capacity limits of each deficit level are indicated in (3.24). Each deficit level has a different cost CD$_d$ and limit.

$$\underline{\mathrm{D}}_d \le pd_{pd} \le \overline{\mathrm{D}}_d, \tag{3.24}$$

where,

CD$_d$: incremental cost of the deficit on deficit level *d*;

pd_{pd}: deficit on level *d*, during period *p* (MW);

$\underline{\mathrm{D}}_d, \overline{\mathrm{D}}_d$: minimum (max) deficit on the deficit level *d*.

3.1.3.3 Load requirement

The load requirement constraint is modeled by the equation (3.25), considering the representation of the electrical system elements presented in the last sections. The Lagrange multiplier of the load requirement equation represents the marginal cost of each subsystem *s*.

$$\sum_{h\in\mathcal{H}} ph_{ph} + \sum_{\mathcal{F}} pf_{pf} + \sum_{\mathcal{D}} pd_{pd} + \sum_{\mathcal{I}} pi_{pi} - \sum_{\mathcal{I}} pi_{pi} = pl_{ps}, \forall s \in \mathcal{S} \tag{3.25}$$

$\mathcal{H}$: set of hydro plants in operation during period p in the power subsystem s, $\mathcal{H} \quad \mathcal{H}$;

$\mathcal{F}$: set of thermal plants in operation during period p in the power subsystem s, $\mathcal{F} \quad \mathcal{F}$;

$\mathcal{D}$: set of deficit levels in the power subsystem s, $\mathcal{D} \quad \mathcal{D}$;

$\mathcal{I}$: set of the power interconnection lines arriving at subsystem s, $\mathcal{I} \quad \mathcal{I}$;

$\mathcal{I}$: set of the power interconnection lines coming to subsystem s, $\mathcal{I} \quad \mathcal{I}$.

3.1.3.4 Load levels

The load levels are equivalent to the demand in period *p* by loads with different depths and durations. The set of load levels *l* for period *p* is defined below.

$$\mathcal{L} \qquad \mathbb{Z} \qquad l_p\}, \tag{3.26}$$

where,

nl_p: number of load levels for period *p*.

Three load levels are considered in Brazil: low, medium, and peak, representing the periods from lowest to highest power demand.

To illustrate the load levels' implementation in a planning model, consider the average load of a weekly period equal to 30000 MWa, which must be represented in 3 load levels, $nl_p = 3$. For that, the time duration TL_{pl} of each load level *l* in a period *p* must ensure equation (3.27). For instance, $\mathrm{TL}_{p1} = 0.2$, $\mathrm{TL}_{p2} = 0.5$ and $\mathrm{TL}_{p3} = 0.3$.

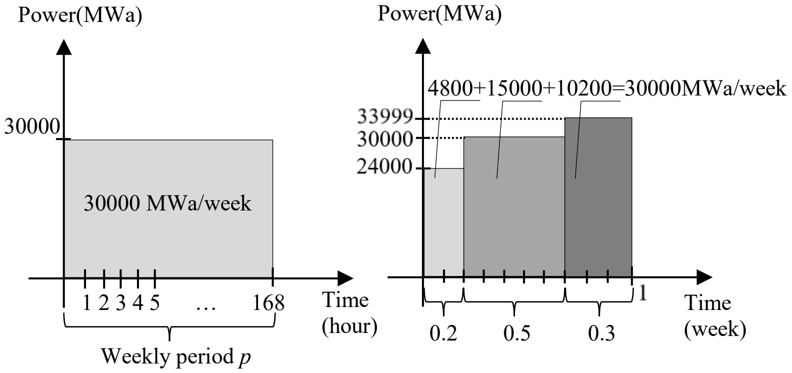
$$\sum_{l \in \mathcal{L}} \mathrm{TL}_{pl} = 1. \tag{3.27}$$

Each load level's depth can be defined as a proportion pu_{pl} regarding the average load defined in period p. For example, consider pu_{p1} = 0.8000, pu_{p2} = 1.0000 and pu_{p3} = 1.1333. Note that the equivalence presented in (3.28) must be satisfied, as follows:

$$pl_{ps} = \sum_{l \in \mathcal{L}} (pl_{ps} \cdot \mathrm{TL}_{pl} \cdot pu_{pl}), \quad \forall s \in \mathcal{S} \tag{3.28}$$

Graphically, Figure 17 presents the equivalence of the average load and the load levels considered in the example:

Figure 17 – Load level example.



Source: Author.

The inclusion of the load levels in the model implies redefining the hydro balance equations presented in Section 3.1.1.2, as it is carried out in (3.29) - (3.30). A variable for representing the turbined outflow and spillage per level load is required. This work considers one equation per level load for the run-of-river plants to avoid using all the available inflow in the period p at a specific load level.

$$v_{ph} - v_{p-1,h} + \sum_{l \in \mathcal{L}} \mathrm{K}_{pl} \left(q_{phl} + sh_{phl} - y_{ph} \right) - \sum_{\mathcal{M}} \sum_{\mathcal{L}} \mathrm{K}_{\tau l} (q_{\tau ml} + sh_{\tau ml}) = 0, \ \forall h \in \mathcal{V} \tag{3.29}$$

$$\mathrm{K}_{pl}(q_{phl} + sh_{phl} - y_{ph}) - \sum_{m \in \mathcal{M}} \mathrm{K}_{\tau l}(q_{\tau ml} + sh_{\tau ml}) = 0, \qquad \forall h \in \mathcal{R} \qquad \mathcal{L} \tag{3.30}$$

where,

q_{phl}: turbined outflow of hydro *h*, during period *p* and load level *l* (m^3/s);

sh_{phl}: spillage of hydro *h*, during period *p* and load level *l* (m^3/s);

K_{pl}: constant to convert flow to volume during period *p* and load level *l*, i.e., $\mathrm{K}_{pl} = \mathrm{K}_p \, \mathrm{TL}_{pl}$.

The turbined outflow and spillage limits for each load level are defined in (3.31) - (3.32) respectively:

$$\underline{\mathrm{Q}}_h \le q_{phl} \le \overline{\mathrm{Q}}_h, \quad \forall l \in \mathcal{L} \qquad \mathcal{H} \tag{3.31}$$

$$0 \le sh_{phl}, \qquad \forall l \in \mathcal{L} \qquad \mathcal{H} \tag{3.32}$$

The constraints to represent the HPF must be applied for each load level *l* considered, as it is presented in (3.33) and variable limits in (3.34), increasing nl_p times the size of the MTGS model.

$$ph_{phl} \ge \mathrm{CP}_{0g} + \mathrm{CP}_{1g} v_{phl} + \mathrm{CP}_{2g} q_{phl} + \mathrm{CP}_{3g} sh_{phl}, \quad \forall l \in \mathcal{L} \qquad \mathcal{G} \tag{3.33}$$

$$\underline{\mathrm{H}}_h \le ph_{thl} \le \overline{\mathrm{H}}_h, \quad \forall l \in \mathcal{L} \qquad \mathcal{H} \tag{3.34}$$

where,

ph_{phl}: generation of hydro *h* during period *p* and load level *l* (MW).

The load requirement modeling considering load level *l* is redefined for each subsystem in (3.35).

$$\begin{aligned} &\mathrm{TL}_{pl} \sum_{h\in\mathcal{H}} ph_{phl} + \sum_{\mathcal{F}} pf_{pfl} + \sum_{\mathcal{D}} pd_{pdl} + \\ &+ \sum_{i\in\mathcal{I}} pi_{pil} - \sum_{\mathcal{I}} pi_{pil} = \mathrm{TL}_{pl} \cdot pl_{psl}, \qquad \forall l \in \mathcal{L} \qquad\qquad \mathcal{S} \end{aligned} \tag{3.35}$$

where,

pf_{phl}: generation of thermal f during period p and load level l (MW);

pd_{pdl}: deficit on level d, during period p and load level l (MW);

pi_{pil}: power interchange i during period p and load level l (MW);

pl_{psl}: load of the subsystem s, period p, and load level l (MW).

Finally, the limits of the thermal generation, deficit, and power interchange, considering the load level approach, are altered as indicated below:

$$\underline{\mathrm{F}}_f \le pf_{pfl} \le \mathrm{TL}_{pl} \cdot \overline{\mathrm{F}}_f, \quad \forall l \in \mathcal{L} \qquad \mathcal{F} \tag{3.36}$$

$$\underline{\mathrm{D}}_d \le pd_{pdl} \le \mathrm{TL}_{pl} \cdot \overline{\mathrm{D}}_d, \quad \forall l \in \mathcal{L} \qquad \mathcal{D} \tag{3.37}$$

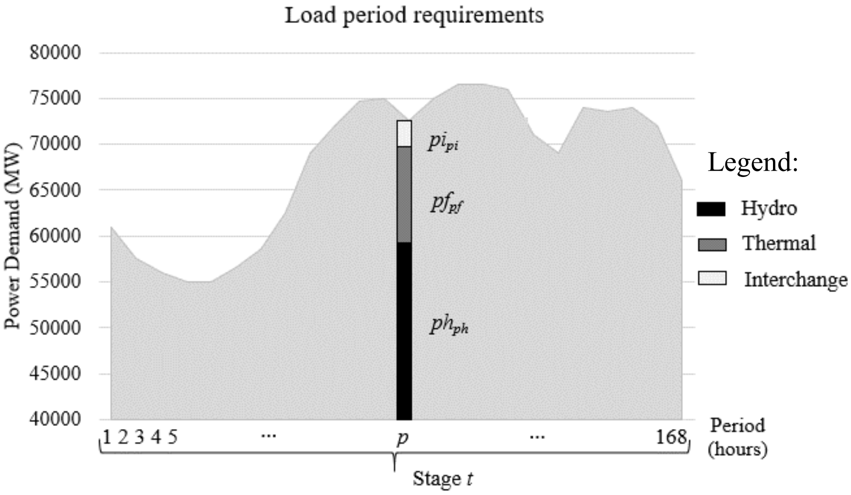
$$\underline{\mathrm{I}}_i \le pi_{pil} \le \mathrm{TL}_{pl} \cdot \overline{\mathrm{I}}_i, \quad \forall l \in \mathcal{L} \qquad \mathcal{I} \tag{3.38}$$

3.1.3.5 Hourly periods

Another approach to model the demand in a stage t for the MTGS employs the load curve, wherein this case, the modeling captures the gradual change of consumption over the periods. A decision variable and model constraints are required by each hourly period, increasing the associated PL model. For instance, consider a weekly stage and a system with the following dimension: ns = 4, nh = 152, nf = 135, ni = 5 and nd = 4. For a weekly discretization and considering nl = 3, the number of variables and constraints is 1506 and 338, respectively. Otherwise, using an hourly time-step of the same week stage, the number of variables equals 91,897, and the constraints are 26,544, representing an increase of approximately 60 times.

Figure 18 shows a representation of the generation required to satisfy the load requirement in the period *p*.

Figure 18 – Representation of the load period approach.



Source: Author.

Source: Author.

In this work, an hourly discretization is implemented only for the first-stage problem. This discretization allows us to model the water delay time when $\tau_{hm} \geq 1$ and thermal UC constraints.

3.2 THE MTGS PROBLEM

The proposed MTGS problem is modeled as a multi-period stochastic optimization problem in which the inflow in each hydro plant is the random variable. A scenario tree with finite realizations represents the stochastic problem. The deterministic equivalent model is presented in (3.39) - (3.45):

$$\min \sum_{t=1}^{T} \left\{ \sum_{n \in \mathcal{N}} \left\{ pa_{tn} \cdot dv_t \cdot \sum \left[\sum_{\mathcal{F}} (\mathrm{CF}_f \, pf_{tpfn}) + \sum_{\mathcal{D}} (\mathrm{CD}_d \, pd_{tpdn}) \right] \right\} \right\} + \theta_T, \tag{3.39}$$

s.t:

$$v_{tphn} + \mathrm{K}_{tp} \left[q_{tphn} + sh_{tphn} \right] - \sum_{m \in \mathcal{M}} \mathrm{K}_{t\tau} (q_{tpmn} + sh_{tpmn}) = v_{ft(p),p-1,h,fn(n,p)} + \mathrm{K}_{tp} y_{tphn}, \forall p, h \in t, \mathcal{V} \tag{3.40}$$

$$\mathrm{K}_{tp}\left[q_{tphn}+sh_{tphn}-y_{tphn}\right]-\sum_{m\in\mathcal{M}}\mathrm{K}_{t\tau}(q_{\tau pmn}+sh_{\tau pmn})=0, \qquad \forall p,h\in t,\mathcal{R} \tag{3.41}$$

$$\sum_{h\in\mathcal{H}} ph_{tphn}+\sum_{\mathcal{F}} pf_{tpfn}+\sum_{\mathcal{D}} pd_{tpdn}+\sum_{\mathcal{I}} pi_{tpin}-\sum_{\mathcal{I}} pi_{tpin}=pl_{tps}, \ \ \forall p,s\in t,\mathcal{S} \tag{3.42}$$

$$ph_{tphn}\geq \mathrm{CP}_{0g}+\mathrm{CP}_{1g}v_{tphn}+\mathrm{CP}_{2g}q_{tphn}+\mathrm{CP}_{3g}sh_{tphn}, \ \forall p,g\in t,\mathcal{G} \tag{3.43}$$

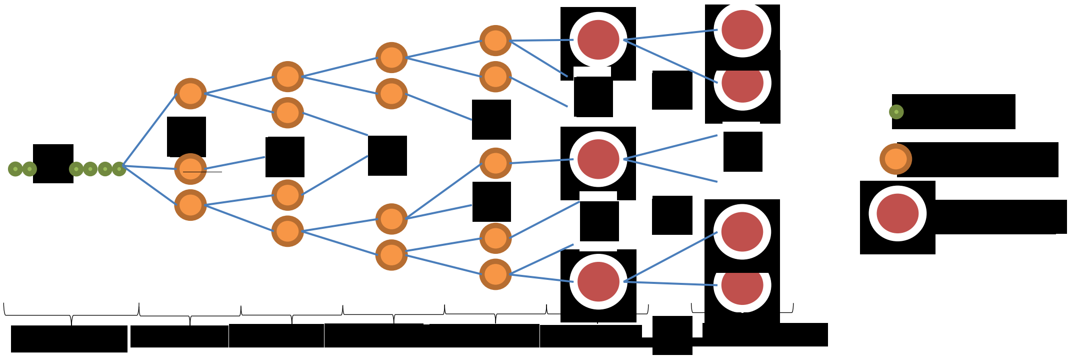
$$\begin{aligned}
&\underline{\mathrm{F}}_f\leq pf_{tpfn}\leq \overline{\mathrm{F}}_f, && \forall p,f\in t,\mathcal{F}\\
&\underline{\mathrm{D}}_d\leq pd_{tpdn}\leq \overline{\mathrm{D}}_d, && \forall p,d\in t,\mathcal{D}\\
&\underline{\mathrm{I}}_i\leq pi_{tpin}\leq \overline{\mathrm{I}}_i, && \forall p,i\in t,\mathcal{I}\\
&\underline{\mathrm{H}}_h\leq ph_{tphn}\leq \overline{\mathrm{H}}_h,\ \ \underline{\mathrm{Q}}_h\leq q_{tphn}\leq \overline{\mathrm{Q}}_h,\ \ 0\leq sh_{tphn}, && \forall p,h\in t,\mathcal{H}\\
&\underline{\mathrm{V}}_h\leq v_{tphn}\leq \overline{\mathrm{V}}_h, && \forall p,h\in t,\mathcal{H}
\end{aligned} \tag{3.44}$$

$$\theta_T\geq \beta_{T+1}^{j\top}{}_{\prime\, TPhn}+\alpha_{T+1}^{j}, \qquad \forall n,h,j\in\mathcal{N}\ \ \mathcal{V}\ \ \mathcal{J} \tag{3.45}$$

The objective function (3.39) comprises the weighted sum of the thermal and power deficit cost of all periods p and scenario tree nodes $n\in\mathcal{N}$. Equations (3.40) and (3.41) represent the water balance equation of hydro plants with reservoir and run-of-river plants, respectively. The water delay time of the upstream hydro plants m relatively to h is represented by τ. Equation (3.42) represents the load requirements for all $p\in t$. A piecewise linear model (3.43) represents the hydro production function in each hydro plant. Constraints (3.44) represent the bounds of the variables and, finally, the coupling with the LTGS problem is represented in (3.45). This work obtains the final water value from a five-year LTGS problem with monthly decisions and the same MTGS system configuration. The LTGS problem is computed via the SDDP algorithm, considering 50 inflow realizations per period, 120 scenarios per iteration with resampling and 12 hours of CPU time.

The proposed MTGS problem considers an hourly discretization for the first week, weekly periods in the remaining weeks of the first month, followed by monthly periods for the rest of the planning horizon, as presented in Figure 19. More than one realization of the random variable is considered from the second week.

Figure 19 – Structure of the proposed MTGS problem.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

The time-step determines the system modeling level that can be considered. For instance, hourly periods in our model include the thermal UC constraints represented by following mixed-integer linear (3.46) - (3.50) (FRANGIONI; GENTILE; LACALANDRA, 2011), (MORALES-ESPAÑA; LATORRE; RAMOS, 2013a), (MORALES-ESPAÑA; LATORRE; RAMOS, 2013b). The dynamic of the process associated with the boiler operation has a high relevance because, in hours, the thermal units can be subject only to a gradual temperature change (WOOD; WOLLENBERG, 1984). Therefore, the modeling must incorporate a period required to start-up or shut down a thermal unit and to variate, in a nominal operation, the thermal generation between consecutive hours.

$$u_{tpfn}\underline{\mathrm{F}}_f \le pf_{tpfn} \le u_{tpfn}\overline{\mathrm{F}}_f, \forall p, f \in \mathcal{P}\ \mathcal{F} \tag{3.46}$$

$$u_{tpfn} - u_{t,p-1,fn} = up_{tpfn} - ud_{tpfn}, \quad \forall p, f \in \mathcal{P}\ \mathcal{F} \tag{3.47}$$

$$\sum_{p'=p-\mathrm{UP}_f+1}^{p} up_{tp'fn} \le u_{tpfn}, \sum_{p'=p-\mathrm{DW}_f+1}^{p} ud_{tp'fn} \le 1 - u_{tpfn}, \forall p, f \in \mathcal{P}\ \mathcal{F} \tag{3.48}$$

$$pf_{tpfn} - pf_{t,p-1,fn} - u_{t,p-1,fn}\Delta\mathrm{UP}_f + u_{t,p-1,fn}\underline{\mathrm{F}}_f \le \underline{\mathrm{F}}_f, \quad \forall p, f \in \mathcal{P}\ \mathcal{F} \tag{3.49}$$

$$pf_{t,p-1,fn} - pf_{tpfn} - u_{tpfn}\Delta\mathrm{DW}_f + u_{tpfn}\underline{\mathrm{F}}_f \le \underline{\mathrm{F}}_f, \forall p, f \in \mathcal{P}\ \mathcal{F} \tag{3.50}$$

$$u_{tpfn}, up_{tpfn}, ud_{tpfn} \in [0,1] \tag{3.51}$$

where,

pf_{pf}: generation of thermal f during the period p (MW);

u_{pf}: binary variable equals 1 whether the thermal plant f is online; otherwise, it is equal to 0;

up_{pf}: binary variable that takes the value of 1 if thermal plant f of period p is turned on, or 0 otherwise;

ud_{pf}: binary status variable that takes the value of 1 if thermal plant f of period p is turned off, or 0 otherwise;

UP_f: minimum number of periods p that the thermal plant must remain online (h);

DW_f: minimum number of periods p that the thermal plant must remain offline;

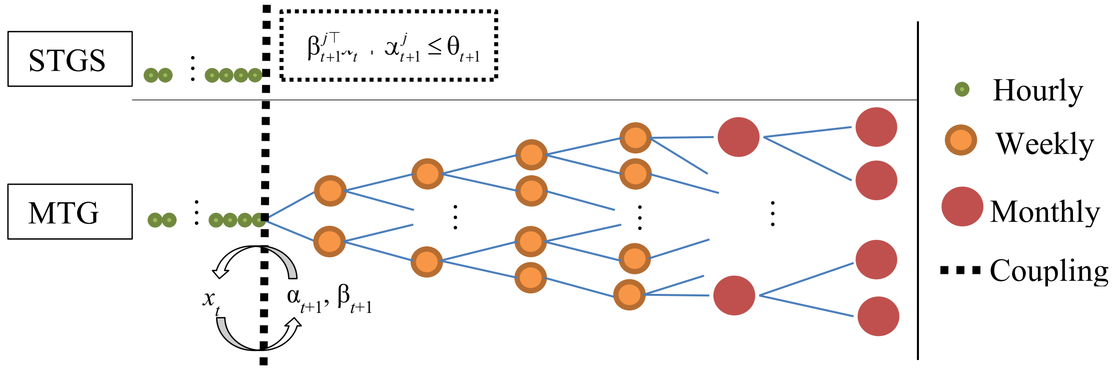
$\Delta\mathrm{UP}_f$: ramp-up rate of thermal f (MW/h);

$\Delta\mathrm{DW}_f$: ramp-down rate of thermal f (MW/h).

Equation (3.46) represents the thermal generation limits. The 3-binary model is shown in (3.47). In turn, (3.48) is related to the classical minimum uptime and downtime constraints. Finally, the maximum variation rate (increase/decrease) of thermal generation in nominal operative condition between two consecutive periods are modeled by the ramp-up and ramp-down constraints presented in (3.49) and (3.50), respectively. On the other hand, for weekly and monthly periods, we include a power load level approach.

Before presenting our solution strategy, we would like to highlight that in the end of the solution procedure, it is necessary to output a cost-to-go function at the end of the first week, as illustrated in Figure 20. The STGS problem uses this cost-to-go function for evaluating the future value of the water in the reservoirs.

Figure 20 – Coupling MTGS-STGS


STGS
$\beta^{j\top}_{t+1}x_t + \alpha^{j}_{t+1} \leq \theta_{t+1}$
MTG
x_t
$\alpha_{t+1}, \beta_{t+1}$
Hourly
Weekly
Monthly
Coupling

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

The next chapter presents the solution strategies proposed for solving the MTGS problem.

4 SOLUTION METHODS

Due to the large-scale characterization of the MTGS problem and, especially considering a multi-stage approach for representing the inflow uncertainty, it is necessary to employ efficient solution methods. This work explores the performance of different techniques to compute the cost-to-go functions for the MTGS problem. We consider two different approaches: (i) an original regularized NBD employing the central path Chebyshev centers of (ELZINGA; MOORE, 1975), and (ii) a two-stage reformulation of the multi-stage MTGS, which is efficiently solved via the Level Bundle method.

Firstly, the NBD is presented in Section 4.1. Approach (i) is detailed in Section 4.2, whereas (ii) is investigated in Section 4.3.

4.1 NESTED BENDERS DECOMPOSITION (NBD)

As discussed in Chapter 2, a multi-stage stochastic linear program is formulated as in (2.24)-(2.26). The NBD principle consists of approximating the recourse function $\mathcal{Q}$ $_{+1]})$ of the underlying multi-stage stochastic program iteratively through optimality cuts of the form $\beta_{t+1}^{c\top} \cdot \alpha_{t+1}^{c} \leq r_{t+1}$. For each iteration k, a trial point x_t^k is obtained and coupled with its subsequent subproblems related to stage t + 1. This procedure, known as the forward step, is carried out for t = 1,…, T. In the backward step, the obtained trial points are used to construct linear approximations of the cost-to-go-function. This step is based on the dual solution of the equivalent dynamic programming equations (2.24) and computation of the recourse function sub-gradient around x_t^k. To exemplify this step, consider the stochastic linear subproblem for stage t = T - 1 in (4.1):

$$Q_{T-1}(x_{T-2}^{k}, \xi_{[T-1]}) = \begin{cases} \min\limits_{x_t \geq 0} & c_{T-1}^{\top}{}_{T-1} + \mathcal{Q} \qquad\qquad {}_{]}) \\ s.t. \, A_{T-1}x_{T-1}^{k} = b_{T-1} - B_{T-1}x_{T-2}^{k} \end{cases} \tag{4.1}$$

where the recourse function $\mathcal{Q}$ $\qquad {}_{]}) = \sum_{s=1}^{S} p_s Q_T^s(x_{T-1}^k, \xi_{[T]})$ and

$$Q_T^s(x_{T-1}^k, \xi_{[T]}) = \begin{cases} \min\limits_{x_T \geq 0} \; c_T^\top x_T \\ s.t. \quad A_T x_T = b_T^s - B_T^s x_{T-1}^k \end{cases} = \begin{cases} \max\limits_{\lambda_T} \lambda_T^{s,k\top}(b_T - B_T^s x_{T-1}^k) \\ s.t. \;\; A_T^\top \lambda_T \leq c_T. \end{cases} \tag{4.2}$$

Using the dual solution of (4.2), the resulting recourse function is presented below:

$$\mathcal{Q} \qquad]) = \sum_{s=1}^{S} p_s \lambda_T^{s,k\top}(b_T - B_T^s x_{T-1}^k). \tag{4.3}$$

The subgradient of (4.3) is $\beta^k = -\sum_{s=1}^{S} p_s B_T^{s\top} \lambda_T$. Convexity of $\mathcal{Q}$ yields the following lower approximation:

$$\begin{aligned} \mathcal{Q} \qquad & \mathcal{Q} \qquad \beta^{k\top}(x_{T-1} - x_{T-1}^k) \\ &= \sum_{s=1}^{S} p_s \lambda_T^{s,k\top}(b_T \quad B_T x_{T-1}) \quad \sum_{s=1}^{S} p_s \lambda_T^{\quad\top} B_T(x_{T-1} \quad x_{T-1}) \\ &= \sum_{s=1}^{S} p_s \lambda_T^{s,k\top}(b_T - B_T^s x_{T-1}). \end{aligned} \tag{4.4}$$

Where $\beta_T = -\sum_{s=1}^{S} p_s \lambda^{s,k\top} B_T$ is the cut slop, and $\alpha_T = \sum_{s=1}^{S} p_s \lambda_T^{s,k\top} b_T$ is the constant cut term. The NBD master subproblem for stage *t* is defined as:

$$\begin{cases} \min\limits_{x_t \geq 0, \theta_{t+1} \geq 0} \; c_t^\top x_t + \theta_{t+1} \\ s.t. \quad A_t x_t = b_t - B_t x_{t-1}^k, \\ \qquad\;\; \beta_{t+1}^{c\top} x_t + \alpha_{t+1}^c \leq \theta_{t+1}, \quad \forall j \in \mathcal{J} \end{cases} \tag{4.5}$$

In each iteration *k* of the forward step, a feasible and implementable policy $x_t^k(\xi_{[t]})$ up to stage *t* and an upper bound $\bar{z}^k$ for the optimal value for the problem (2.26) are obtained. The upper bound associate with the policy $x^k(\xi)$ is defined as:

$$\bar{z}^k = \mathbb{E}\left[\,{}^{T}_{t=1} \qquad \xi_{[t]})\right], \tag{4.6}$$

In the backward step, a lower bound $\underline{z}^k$ for the optimal value of (2.23) is found as the optimal value of the first stage subproblem (4.7):

$$\underline{z}^k = \begin{cases} \min\limits_{x_1 \geq 0, \theta_2 \geq 0} c_1^\top x_1 + \theta_2 \\ s.t. \quad A_1 x_1 = b_1, \\ \qquad \beta_2^{j\top} x_1 + \alpha_2^j \leq \theta_2, \qquad \forall j \in \mathcal{J} \end{cases} \tag{4.7}$$

The NBD process stops when the current optimality gap, defined by $gap^k = \bar{z}^k - \underline{z}^k$, satisfies $gap^k \leq$ Tol, for a defined tolerance Tol ≥ 0. Otherwise, the iteration counter ($k \leftarrow k+1$) is increased, and another forward-backward iteration is carried out, updating the cutting-plane models. This procedure is summarized in the NBD algorithm (BIRGE, 1985) presented below:

Algorithm 1 – NBD

Step 0: Select a tolerance tol ≥ 0. Define $k = 1$ and $\mathcal{J}$, for t =1, …, T;

Step 1: Forward Step. For t =1,…, T, solve (4.5) for each realization of the scenario tree to obtain $x_t^k = x_t^k(\xi_{[t]})$. For t=1, obtain the lower bound according to (4.7). Compute $\bar{z}^k$ as (4.6);

Step 2: Stopping criterion. Set $gap^k = \bar{z}^k - \underline{z}^k$. If gap^k < tol, then stop. The optimal value of the problem is $\bar{z}^k$, and the obtained cost-to-functions are the set of cuts $(\alpha_t^j, \beta_t^j)_{j \in \mathcal{J}}$ for t = 1,…, T. Otherwise, go to Step 3;

Step 3: Backward Step. Use the Lagrange multiplies of (4.2) to calculate α_T^k and β_T^k. Set $\mathcal{J} \quad \mathcal{J}$. For t= T–1, …, 2, solve (4.5) to calculate α_t^k and β_t^k. Set $\mathcal{J} \quad \mathcal{J}$.

Step 4: Loop: Set $k = k$+1 and return to Step 1**.**

Algorithm 1 is a sort of cutting-plane method (KELLEY, 1960), which is well-known for being non-monotonic and having the following shortcomings: (*i*) slow convergence (instability); and (*ii*) inability of exploring good trial points obtained during the iteration. As an attempt to accelerate the convergence process in practice, we modify the cuts in the forward step by employing some ideas related to the Chebyshev center of the NBD subproblems' feasible sets without compromising convergence analysis. This is the subject of the following section, which presents original material.

4.2 NBD WITH CHEBYSHEV CENTERS

The NBD computational performance is highly dependent on the trial points defined in the forward step since it determines the regions where the cost-to-go functions are constructed. To obtain an improvement, other regions of the cost-to functions can be explored. For that, we propose to define implementable and feasible policies with Chebyshev centers. This section is based on (BELTRÁN et al., 2020).

4.2.1 Chebyshev center and related optimization methods

Let $\mathcal{P}$ $\mathfrak{R}^n : \beta^{j\top} x \; \alpha^j \leq 0, j \in J\}$ be a nonempty and bounded polyhedron. Let x_c and σ be, respectively, the center and the largest radius of the ball $B(x_c, \sigma) := \{x_c + d : \| d \| \leq \sigma\}$ inscribed in $\mathcal{P}$. The point x_c is known as the Chebyshev center $\mathcal{P}$. Such a center depends on the considered norm ||·|| and it is the point deepest inside the polyhedron (w.r.t. the norm). It can be shown that computing the Chebyshev center of $\mathcal{P}$ amounts to solving the following LP:

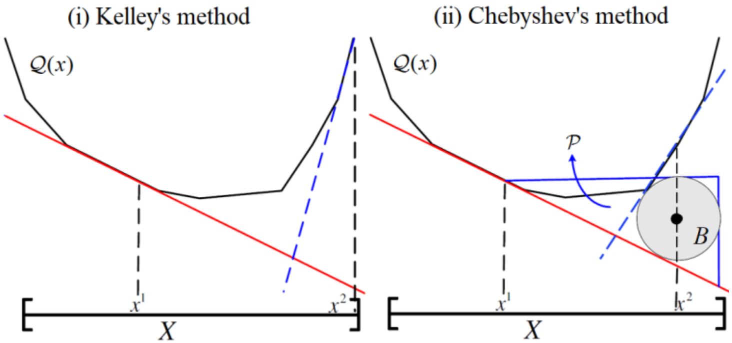
$$\max_{x,\sigma} \sigma \text{ s.t. } \beta^{j\top} x_t \; \alpha^j + \sigma \cdot \left\| \beta^j \right\|_* \leq 0, \quad \forall j \in \mathcal{J} \tag{4.8}$$

where $\|\cdot\|_*$ is the dual norm of ||·||. Let $\min_{x \in X} \mathcal{Q}$ be a deterministic problem with $\mathcal{Q}$ $\mathfrak{R}$ $\mathfrak{R}$ a convex function and $X \subset \mathfrak{R}^n$ a bounded polyhedron. Under these assumptions, (ELZINGA; MOORE, 1975) proposes a minimizing algorithm that defines iterates as the Chebyshev center of the polyhedron:

$$\mathcal{P} \qquad \mathbb{R} \qquad \mathcal{J} \tag{4.9}$$

where $\bar{z}$ is a given upper bound on the problem's optimal value and $\beta^{j\top} x + \alpha^{j}$ the j-th cut of $\mathcal{Q}$. Figure 21 shows an illustration of the Chebyshev centers. Suppose that the function $\mathcal{Q}$ is approximated by cuts (linearizations) around iterates. Kelley's method defines a trial point by minimizing the cutting-plane model over *X*. Instead, the Chebyshev approach defines a new iterate as the center of the largest ball *B* inscribed in $\mathcal{P}$, which implies constructing the next cut in another region of $\mathcal{Q}$.

Figure 21 – Example of Kelley and Chebyshev iterates.


(i) Kelley's method

(ii) Chebyshev's method

Source: (BELTRÁN et al., 2020).

Computationally, the method of (ELZINGA; MOORE, 1975) can be implemented similarly to Kelley's method and can improve the convergence speed. Following these lines, (OUOROU, 2009) proposes to combine Chebyshev centers with bundle methods, and this idea has been extended in (DE OLIVEIRA, 2017) for particular classes of nonsmooth convex optimization problems. All these works make use of a reliable upper bound on the considered deterministic problem's optimal value. In contrast, in MSLP, such bounds are stochastic because they depend on each node of the scenario tree and decisions made in previous stages. This upper bound estimation hinders a direct application of (ELZINGA; MOORE, 1975) to multi-stage stochastic programming. In what follows, we revisit and adapt the ideas of (ELZINGA; MOORE, 1975) to deal with MSLPs.

4.2.2 The Chebyshev approach

The NBD with Chebyshev centers – NBD-CC – proposed below employs the backward procedure of the NBD but modifies the forward step by replacing (4.5) with the following LP, which defines Chebyshev centers:

$$\begin{cases} \max\limits_{x_t, s_t, \sigma_t, \omega_t} \ \sigma_t \\ s.t. \quad A_t x_t = b_t - B_t x^k_{t-1}, \\ \qquad \sigma_t + s_t \le \bar{z}\left(\xi_{[t]}\right) \\ \qquad (c_t + \beta^j_{t+1})^\top x_t \ ' \ \alpha^j_{t+1} + \varpi_t \left\|(1, c_t + \beta^j_{t+1})\right\|_* \le s_t, \quad \forall j \in \mathcal{J} \\ \qquad \sigma_t \le \varpi_t,\ x_t \ge 0,\ \varpi_t \ge 0,\ \sigma_t \in \mathbb{R} \qquad \mathbb{R} \end{cases} \tag{4.10}$$

The following result, whose proof is given in Appendix A of (BELTRÁN et al., 2020), shows that (4.10) is a well-defined LP. Moreover, a relation between this LP and the Chebyshev center is also established.

Proposition 1: *Consider* (4.10) *with a given finite* $\bar{z}(\xi_{[t]})$ *and a point* $\bar{x}_{t-1}$. *Then:(i) if LP* (4.5) *is solvable, so is* (4.10). *(ii)* (Kelley iterate). *If* $\bar{z}(\xi_{[t]}) \le \underline{Q}_t(x_{t-1}, \xi_t)$ *then, the x*$_t$*-solution of* (4.10) *also solves* (4.5). *(iii)* (Chebyshev iterate) *If* $\bar{z}(\xi_{[t]}) > \underline{Q}_t(x_{t-1}, \xi_t)$ *then x*$_t$*-solution of* (4.10) *is the center of the largest ball inscribed in the polyhedral set*

$$\begin{Bmatrix} x_t \in \mathbb{R} & (\xi_{[t]}) \\ \theta_{t+1} \in \mathbb{R} & {}_1)^\top x_t \ ' \ \alpha^j_{t+1} \le \theta_{t+1}, \forall j \in \mathcal{J} \end{Bmatrix} \tag{4.11}$$

(iv) LP (4.10) *is equivalent, in terms of solution x*$_t$*, to the following simplified LP:*

$$\begin{cases} \min\limits_{x_t \geq 0, \theta_{t+1} \geq 0, \omega_t \geq 0} & c_t^\top x_t + \theta_{t+1} \\ s.t. & A_t x_t = b_t - B_t x_{t-1}^k, \\ & \beta_{t+1}^{j\top} x_t + \alpha_{t+1}^j + \varpi_t \left\|(1, c_t + \beta_{t+1}^j)\right\|_* \leq \theta_{t+1}, \quad \forall j \in \mathcal{J} \\ & \bar{z}_n^k(\xi_{[t]}) - (c_t^\top x_t + \theta_{t+1}) \leq \omega_t. \end{cases} \tag{4.12}$$

Compared to (4.5), the LP (4.12) only has one more variable and one additional constraint. As a result, the computational burden to solve (4.12) is comparable to the one used for solving (4.5). However, the formulation (4.12) has a shortcoming compared to (4.5): (4.12) makes use of an estimated upper bound on the tree node's value $Q(\bar{x}_{t-1}, \xi_t)$. Preliminary numerical experiments described below show that estimating a reasonable upper bound $\bar{z}(\xi_{[t]})$ is not an easy task. Several strategies for defining $\bar{z}(\xi_{[t]})$ (e.g., the sample average of the costs issued by the tree node ξ_t) have shown unpromising: the computed value $\bar{z}(\xi_{[t]})$ is larger than expected, in which case the Chebyshev center could be any feasible point of (4.5). A simple and straightforward strategy to obtain iterates different from Kelley's method is to fix the inscribed ball's radius: this amounts to replacing the constraint $\bar{z}(\xi_{[t]}) - (c_t^\top x_t + \theta_{t+1}) \leq w_t$ in (4.12) with $\bar{\sigma}_t \leq w_t$, in which $\bar{\sigma}_t \geq 0$ ideally estimates $Q(\bar{x}_{t-1}, \xi_t) - \underline{Q}(\bar{x}_{t-1}, \xi_t) \geq 0$. Notice that $\bar{\sigma}_t \leq w_t$ must be active at the solution. Accordingly, we can remove the decision variable w_t and write a variant (4.12) as follows:

$$\begin{cases} \min\limits_{x_t \geq 0, \theta_{t+1} \geq 0} & c_t^\top x_t + \theta_{t+1} \\ s.t. & A_t x_t = b_t - B_t x_{t-1}^k, \\ & \beta_{t+1}^{j\top} x_t + \alpha_{t+1}^j + \bar{\sigma}_t \left\|(1, c_t + \beta_{t+1}^j)\right\|_* \leq \theta_{t+1}, \forall j \in \mathcal{J} \end{cases} \tag{4.13}$$

Note that (4.13) has the same dimension and number of constraints as (4.5). The parameter $\bar{\sigma}_t$ can be updated with several simple heuristics, which must guarantee that $\bar{\sigma}_t \to 0$ if $k \to \infty$ and $\bar{\sigma}_t = 0$ at the convergence. The NBD-CC algorithm is presented below:

Algorithm 2 – NBD-CC

Step 0: Select a tolerance tol ≥ 0. Define k = 1, $\mathcal{J}$ and choose a rule to define $\bar{\sigma}_t$, for t =1, …, T;

Step 1: Forward Step. For t =1,…, T-1 and tree nodes in these periods, solve (4.13) to define $x_t^k = x_t^k(\xi_{[t]})$. Set $x_T^k = x_T^k(\xi_{[T]})$ for all nodes belong to stage T as solutions of the subproblem (4.5). For t=1, obtain the lower bound according to (4.7). Compute $\bar{z}^k$ as (4.6);

Step 2: Stopping criterion. Set $gap^k = \bar{z}^k - \underline{z}^k$. If gap^k < tol, then stop. The optimal value of the problem is $\bar{z}^k$, and the obtained cost-to-functions are the set of cuts $(\alpha_t^j, \beta_t^j)_{j\in\mathcal{J}}$ for t = 1,…; T. Otherwise, go to Step 3;

Step 3: Backward Step. Use the Lagrange multipliers of (4.2) to calculate α_T^k and β_T^k. Set $\mathcal{J}$ $\mathcal{J}$. For t= T−1, …, 2, solve (4.5) to calculate α_t^k and β_t^k. Set $\mathcal{J}$ $\mathcal{J}$.

Step 4: Loop: Set k = k+1, update $\bar{\sigma}_t \geq 0$ according to a given rule, and return to Step 1**.**

Note that the backward step is unaltered regarding the NBD one. The Chebyshev strategy only focuses on modifying the trial points obtained in the forward step. A practical interpretation of the Chebyshev centers and how the parameter $\bar{\sigma}_t$ influences the solution of LP (4.13) is shown in the next subsection.

4.2.3 A practical interpretation of the Chebyshev centers

In this section, we exemplify the effect of $\bar{\sigma}_t$ in (4.13). Initially, note that[3] $\bar{\sigma}_t \left\|(1, c_t + \beta_{t+1}^j)\right\|_*$ only affects the intercept α_{t+1}^j; therefore, the original slope β_{t+1}^j is preserved. A natural question is how this new intercept affects the next trial point x. To this end, consider a reduced system with one hydro and five thermal plants. The hydro has a 1,000 m^3/s limit in the turbined outflow (q_t) and 4,000 hm^3 maximum volume (v_t). The hydro production function is given by $ph_t = q_t$. The initial volume is 2,800 hm^3, and the spillage (s_t) is a slack variable. The generation of each thermal plant (pf_{tf}) is limited to 200 MWa[4]. The unitary variable costs are 10, 20, 25, 30, and 40 R$/MWa. In this problem, a 1000 MWa load must be met over a

[3] We use the Euclidean norm, i.e. $\left\|(1, c_t + \beta_{t+1}^j)\right\|_* = \sqrt{1 + \left\|c_t + \beta_{t+1}^j\right\|_2^2}$.

[4] Unit of energy related to the constant operation of a MW capacity for a period.

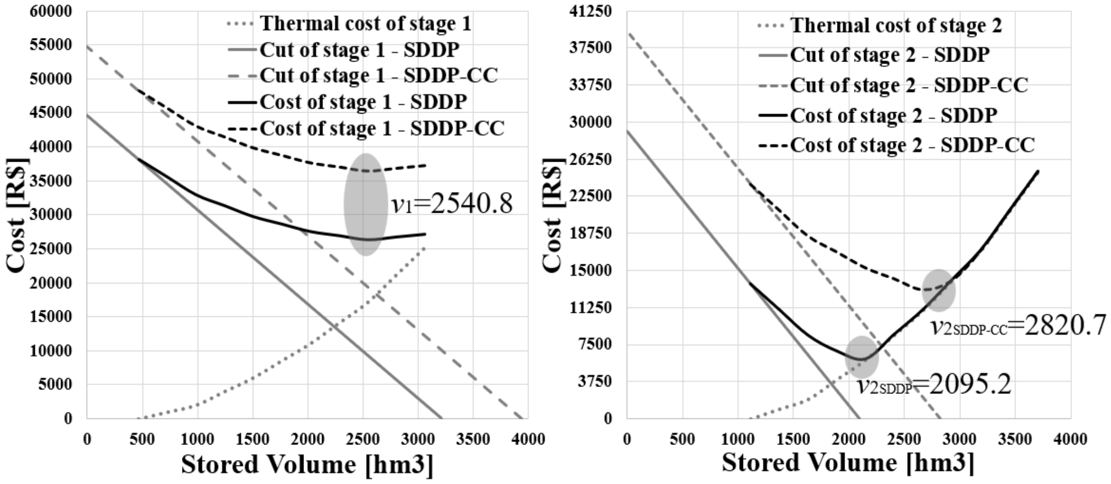
four-month horizon. Finally, a stage-wise scenario tree with common samples and five inflow (ξ_t^n) realizations (m^3/s) per month is considered, with values $\xi_1{}^1 = 100$, $\xi_2{}^n$ = [450 150 300 100 200], $\xi_3{}^n$ = [100 250 180 130 270] and $\xi_4{}^n$ = [400 300 500 200 600]. The month t NBD-CC subproblem[5] is presented in (4.14), $\bar{\sigma}_t$ equals zero for the NBD.

$$\begin{cases} \min \ 10pf_{t1} + 20pf_{t1} + 25pf_{t3} + 30pf_{t4} + 40pf_{t5} + \theta_{t+1} \\ \text{s.t. } pf_{t1} + pf_{t2} + pf_{t3} + pf_{t4} + pf_{t5} + q_1 = 1{,}000 \\ \qquad v_t + \mathrm{K}_t \cdot (q_t + s_t) = v_{t-1} + \mathrm{K}_t \cdot y_{tn} \\ \qquad \beta_{t+1}^{j\top} v_t + \alpha_{t+1}^{j} + \bar{\sigma}_t \left\| (1, c_t + \beta_{t+1}^{j}) \right\|_* \leq \theta_{t+1},\ j \in \mathcal{J} \\ \qquad 0 \leq pf_{tf} \leq 200,\ 0 \leq q_t \leq 1{,}000,\ s_t \geq 0, 0 \leq v_t \leq 4{,}000, \theta_{t+1} \in \Re. \end{cases} \tag{4.14}$$

Solving all of the scenarios in the forward step via NBD, we obtained an exact solution with R$ 34,051.40 cost in 9 iterations. The NBD-CC obtains the same solution in 6 iterations. The initial iteration and the first cut built in the NBD and NBD-CC are identical. In contrast, the NBD-CC defines different trial points from the second iteration, when the cost-to-go function is equal to zero in the solution. Note that for $t = 1$, $\theta_{t+1}>0$ thus $\bar{\sigma}_t$ only includes an offset in the LP cost, maintaining the same trial point that the NBD (Figure 22-left). On the other hand, for $t = 2$, using the NBD algorithm, a solution with θ_{t+1} equals zero is reached. Thus, the $\bar{\sigma}_t$ effect is on modifying the LP solution (4.14), obtaining more stored volume (36%) with the NBD-CC (Figure 22-right). In this case, when the convergence criterion is satisfied, the trial points issued by the NBD-CC speed-up the convergence rate of the algorithm by visiting other regions of the cost-to-go function.

[5] The constant $\mathrm{K}_0 = 2.592$ converts flow to volume considering a 1-month time step.

Figure 22 – Comparison of the NBD and NBD-CC objective function and resulting trial point.



(a) Stage 1.

(b) Stage 2.

Source: (BELTRÁN et al., 2020).

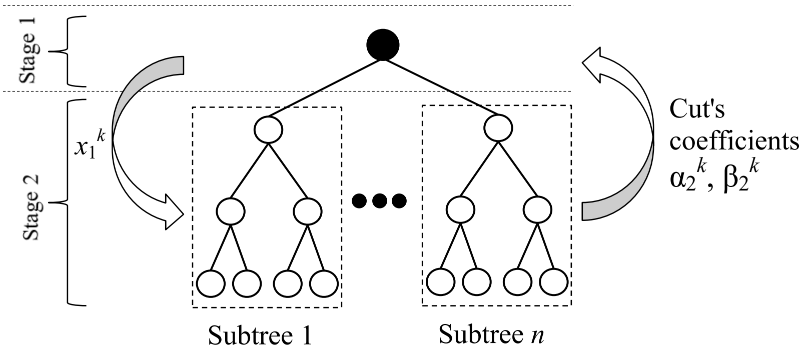
According to the explanations and results presented in Appendix A, the NBD-CC method presents an unsatisfactory performance to solve the considered MTGS problem. However, several tests in (BELTRÁN et al., 2020), indicated in Appendix A, show that the Chebyshev approach is a suitable strategy to deal with LTGS problems when equipped with the SDDP algorithm.

The following section presents another proposal, which has proven successful in dealing with the underlying MTGS.

4.3 TWO-STAGE SOLUTION METHODS

This section is based on (BELTRÁN; FINARDI; DE OLIVEIRA, 2021), where a two-stage reformulation of the considered multi-period MTGS problem is investigated. The key aspect is to simulate a two-stage problem employing deterministic equivalent LPs (subtrees) from the second week of the planning horizon, as depicted in Figure 23.

Figure 23 – Two-stage diagram.



Source: Author.

Such a reformulation is motivated by the excellent numerical performance of some optimization techniques for two-stage stochastic programming. As shown in Figure 23, methods for two-stage stochastic programs obtain a trial point x_1^k by solving the first-stage problem, and afterward, an oracle is called. An oracle is a process in which the two-stage subproblems (2.21) are computed to obtain an approximation of the cost-to-go-function (cut's coefficients α_2^k and β_2^k) around the trial point x_1^k.

Among the two-stage solution methods applicable to the proposed reformulation of the MTGS are the L-Shaped (VAN SLYKE; WETS, 1969), which is the NBD applied to two-stage stochastic programs, and bundle methods. This work considers the extended level bundle method described in the sequel.

4.3.1 Level bundle methods

The Level Decomposition (LD) is the level bundle method combined with the two-stage decomposition applied to two-stage stochastic programs (WOLF et al., 2014). Level bundle methods were proposed in (LEMARÉCHAL; NEMIROVSKII; NESTEROV, 1995) as variants of the proximal bundle methods. Further developments have been proposed by (KIWIEL, 1995), (FABIÁN, 2000), (DE OLIVEIRA; SAGASTIZÁBAL, 2014), (WOLF et al., 2014), (DE OLIVEIRA, 2016). As our two-stage model has binary variables, we need to employ the Extended Level Decomposition (ELD). The ELD defines trial points as projections of given stability centers $\hat{x}^k$ onto level sets of the cutting-plane model. To accomplish this task, the ELD uses a level parameter f_{lev}^k, which is iteratively updated. For that, the ELD applied to the considered MTGS problem solves the following (mixed-binary) first-stage master problem:

$$\begin{aligned} \min_{x \in X} \ & \frac{1}{2}\left\|x_1 - \hat{x}^k\right\|_\Diamond \\ \text{s.t.} \ & (c+\beta^j)^\top x_1 + \alpha^j \leq f_{lev}^k, \ \forall j \in \mathcal{J} \end{aligned} \tag{4.15}$$

In this notation, $\|\cdot\|_\Diamond$ can be l_1, l_∞, or l_2 norms[6], and X is a mixed-binary set. The master problem can be infeasible if $f_{lev}{}^k$ is less than the optimal value of (4.15). In this case, $f_{lev}{}^k$ must be updated until (4.15) is feasible. In this work, before computing (4.15), (4.7) is previously solved at each iteration to obtain a lower bound of the problem that assures $f_{lev}{}^k \geq \underline{z}$. The ELD algorithm is presented below.

Algorithm 3 – Extended Level Decomposition

Step 0: Select a tolerance tol > 0, $\kappa \in (0,1)$, $\Delta_{\text{ref}} = \infty$ and $k = 0$. Solve (4.7) without cuts to define $x_{\text{best}} = x_1^0$ and $\underline{z}^k$. Call an oracle to compute $\bar{z}^k$, α^0, β^0 and set $\mathcal{A}$;

Step 1: Set $\Delta^k = \bar{z}^k - \underline{z}^k$. If $\Delta^k \leq \bar{z}^k \cdot \text{tol}$, return x_{best}, $\bar{z}^k$ and stop. Otherwise, go to step 2;

Step 2: If $\Delta^k \leq \kappa\, \Delta_{\text{ref},}$ then $\Delta_{\text{ref}} = \Delta^k$ and $\hat{x}^k = x_{\text{best}}$;

Step 3: Compute $f_{lev}{}^k = \bar{z}^k - \kappa\, \Delta^k$ and try to find $x_1{}^{k+1}$ by solving (4.15) with $\mathcal{J}$ $\mathcal{A}$. If (4.15) is infeasible, set $\underline{z}^k = f_{lev}^k$ and return to Step 1;

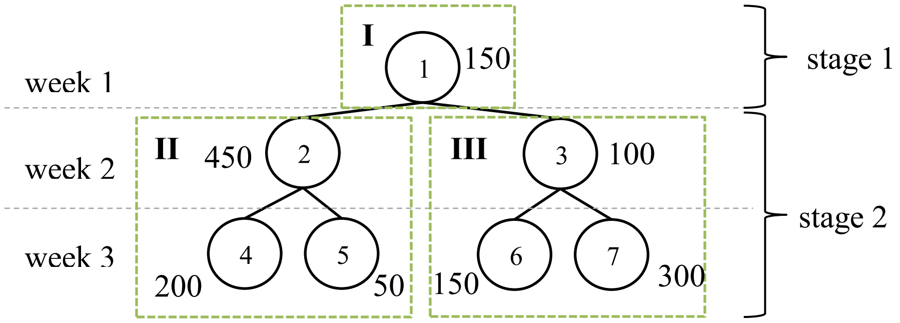
Step 4: Compute $f^{\,k+1} = c^\top x_1{}^{k+1} + Q(x_1{}^{k+1})$, β^{k+1}, α^{k+1} and define $\mathcal{A}$ $+1\}$. If $f^{k+1} < \bar{z}^k$, $\bar{z}^k = f^{k+1}$ and $x_{\text{best}} = x_1{}^{k+1}$. Set $k = k$+1.and return to Step 1.

[6] For the l_2 norm, $\|\cdot\|_2^2$ is usually used.

4.4 A PRACTICAL EXAMPLE OF L-SHAPED AND ELD METHODS

We present a practical example to illustrate the main differences between LS (a two-stage version of NBD) and ELD methods. To this end, consider a reduced system with five thermal plants and one hydro plant. Each thermal plant's maximum generation is $\overline{\mathrm{F}}_f = 250$ MWa[7] with a unitary variable cost CF_f equals 10, 20, 30, 40, and 50 R$/MWa, respectively. Regarding the hydro plant, the turbined outflow is limited to 1400 m^3/s, and the maximum volume is 3,000 hm^3. The initial volume is 1,500 hm^3, and the spillage is a slack variable. The hydro production function is given by $ph_{tp} = q_{tp}$. Finally, this hydrothermal system must meet a 1,400 MWa load over a three-week horizon considering an equiprobable inflow scenario tree presented in Figure 24. The values are in m^3/s, and the node number is inside the circle. Additionally, a two-stage decomposition is carried out, resulting in 3 subproblems: for stage 1, subproblem I associated with node 1, and, for stage 2, subproblems II and III related to nodes 2, 4, 5 and nodes 3, 6, 7 respectively.

Figure 24 – Inflow scenario tree and two-stage decomposition.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

In terms of the subproblem structures, the only difference between the LS and ELD algorithms is subproblem I. In general terms, the ELD first-stage subproblem is harder to compute, but the trial points obtained per iteration can accelerate the convergence process. Subproblem I – LS is presented in (4.16) and subproblem I – ELD using l_2 norm in (4.17), where constant $\mathrm{K}_{t,p}$ = 0.6048 converts flow to volume considering a 1-week step. For

[7] Unit of energy related to the constant operation of a MW capacity for a period.

simplicity, we present the variables with only the sub-indices for stage *t*, period *p*, and thermal plant *f*: pf_{pf}, q_p, s_p, v_p, r_t.

Subproblem I – LS:

$$
\left\{
\begin{array}{l}
\min \;\; 10pf_{11} + 20pf_{12} + 30pf_{13} + 40pf_{14} + 50pf_{15} + \theta_2 \\
\text{s.t.} \;\; pf_{11} + pf_{12} + pf_{13} + pf_{14} + pf_{15} + q_1 = 1{,}400 \\
\qquad v_1 + \mathrm{K}_1 \cdot (q_1 + s_1) = 1{,}500 + \mathrm{K}_1 \cdot 150 \\
\qquad \beta_2^{j\top} v_1 \cdot \alpha_2^j \le \theta_2, \; j \in \mathcal{J} \\
\qquad 0 \le pf_{1f} \le 250, \; 0 \le q_1 \le 1{,}400, \; s_1 \ge 0, 0 \le v_1 \le 3{,}000, \theta_2 \in \Re.
\end{array}
\right.
\tag{4.16}
$$

Subproblem I – ELD:

$$
\left\{
\begin{array}{l}
\min 0.5[(pf_{11} - \hat{pf}_{11})^2 + (pf_{12} - \hat{pf}_{12})^2 + (pf_{13} - \hat{pf}_{13})^2 + (pf_{14} - \hat{pf}_{14})^2 + \\
\qquad + (pf_{15} - \hat{pf}_{15})^2 + (q_1 - \hat{q}_1)^2 + (s_1 - \hat{s}_1)^2 + (v_1 - \hat{v}_1)^2] \\
\text{s.t.} \;\; pf_{11} + pf_{12} + pf_{13} + pf_{14} + pf_{15} + q_1 = 1{,}400 \\
\qquad v_1 + \mathrm{K}_1 \cdot (q_1 + s_1) = 1{,}500 + \mathrm{K}_1 \cdot 150 \\
\qquad \beta_2^{j\top} v_t \cdot \alpha_2^j + 10pf_{11} + 20pf_{12} + 30pf_{13} + 40pf_{14} + 50pf_{15} \le f_{lev}^k, \; j \in \mathcal{J} \\
\qquad 0 \le pf_{1f} \le 250, \; 0 \le q_1 \le 1{,}400, \; s_1 \ge 0, 0 \le v_1 \le 3{,}000.
\end{array}
\right.
\tag{4.17}
$$

The second-stage structure is exemplified with subproblem II in (4.18). Subproblem III is only necessary to update the node inflows 3, 6, and 7. To denote the associated node *n*, an additional sub-index is used: $pf_{pf,n}$, $q_{p,n}$, $s_{p,n}$, $v_{p,n}$.

Second-stage subproblem:

$$\left\{\begin{array}{l}
\min\ \ 10pf_{21,2}+20pf_{22,2}+30pf_{23,2}+40pf_{24,2}+50pf_{25,2}+\\
+\frac{1}{2}(10pf_{31,4}+20pf_{32,4}+30pf_{33,4}+40pf_{34,4}+50pf_{35,4}+10pf_{31,5}+20pf_{32,5}+30pf_{33,5}+40pf_{34,5}+50pf_{35,5})\\
\text{s.t. } pf_{21,2}+pf_{22,2}+pf_{23,2}+pf_{24,2}+pf_{25,2}+q_{2,2}=1{,}400\\
\quad v_{2,2}+\mathrm{K}_{21}\cdot(q_{2,2}+s_{2,2})=v_1+\mathrm{K}_{21}\cdot 450\\
\quad pf_{31,4}+pf_{32,4}+pf_{33,4}+pf_{34,4}+pf_{35,4}+q_{3,4}=1{,}400\\
\quad v_{3,4}+\mathrm{K}_{22}\cdot(q_{3,4}+s_{3,4})=v_{2,2}+\mathrm{K}_{22}\cdot 200\\
\quad pf_{31,5}+pf_{32,5}+pf_{33,5}+pf_{34,5}+pf_{35,5}+q_{3,5}=1{,}400\\
\quad v_{3,5}+\mathrm{K}_{22}\cdot(q_{3,5}+s_{3,5})=v_{2,2}+\mathrm{K}_{22}\cdot 50\\
\quad 0\leq pf_{pf,n}\leq 250,\ 0\leq q_{p,n}\leq 1{,}400,\ s_{p,n}\geq 0{,}0\leq v_{p,n}\leq 3{,}000.
\end{array}\right. \tag{4.18}$$

Initially, the LS algorithm computes (4.16) obtaining $\underline{z}$ = 0 and v_1 = 744 hm^3. This volume is coupled in the second-stage to compute subproblems II and III, achieving an upper bound $\bar{z}$ = 18,795.64 R$/h without satisfying the convergence criterion. Therefore, a first cut is constructed with parameters β_2^1= 45.47 R$/h×hm^3 and α_2^1 = 52,625 R$/h, and they are added to (4.16). In this way, a new volume v_1 = 1,046.4 hm^3 is obtained, and the process continues until $\bar{z}$ and $\underline{z}$ are sufficiently close to one another.

Regarding the ELD method, the first iteration is the same as the LS method, from such iteration $\bar{z}$ = 18,795.64 R$/h, $\mathcal{J}$ ={1}, and x$_{\text{best}}$ is initialized as the solution of (4.16). In the second iteration, it is necessary to update f_{lev} = $\bar{z} - \kappa\,(\bar{z} - \underline{z})$, in this example, κ = 0.9. However, if f_{lev} < $\underline{z}$, (4.17) is infeasible. To deal with this condition, this work previously solves (4.16) obtaining a lower bound of the problem that guarantees $f_{lev} \geq \underline{z}$; if $\Delta^k = \bar{z}^k - \underline{z}^k$ is lower than tolerance, the algorithm stops. Otherwise, the subproblem (4.17) is computed through an update on f_{lev} = 13,170.64, obtaining a new volume v_1 = 996 hm^3. This volume is then sent to the oracle, which computes $\bar{z}$, and a new cut is constructed. Moreover, if the new trial point makes $\bar{z}$ decreases significantly, x$_{\text{best}}$ is updated as the solution of (4.17). Table 5 presents a comparison of $\underline{z}$, $\bar{z}$ and v_1 between LS and ELD methods per iteration. Note that the trial points obtained by subproblem I – ELD reduce the number of oracle calls compared to subproblem I – LS.

Table 5 – LS and ELD results per iteration.

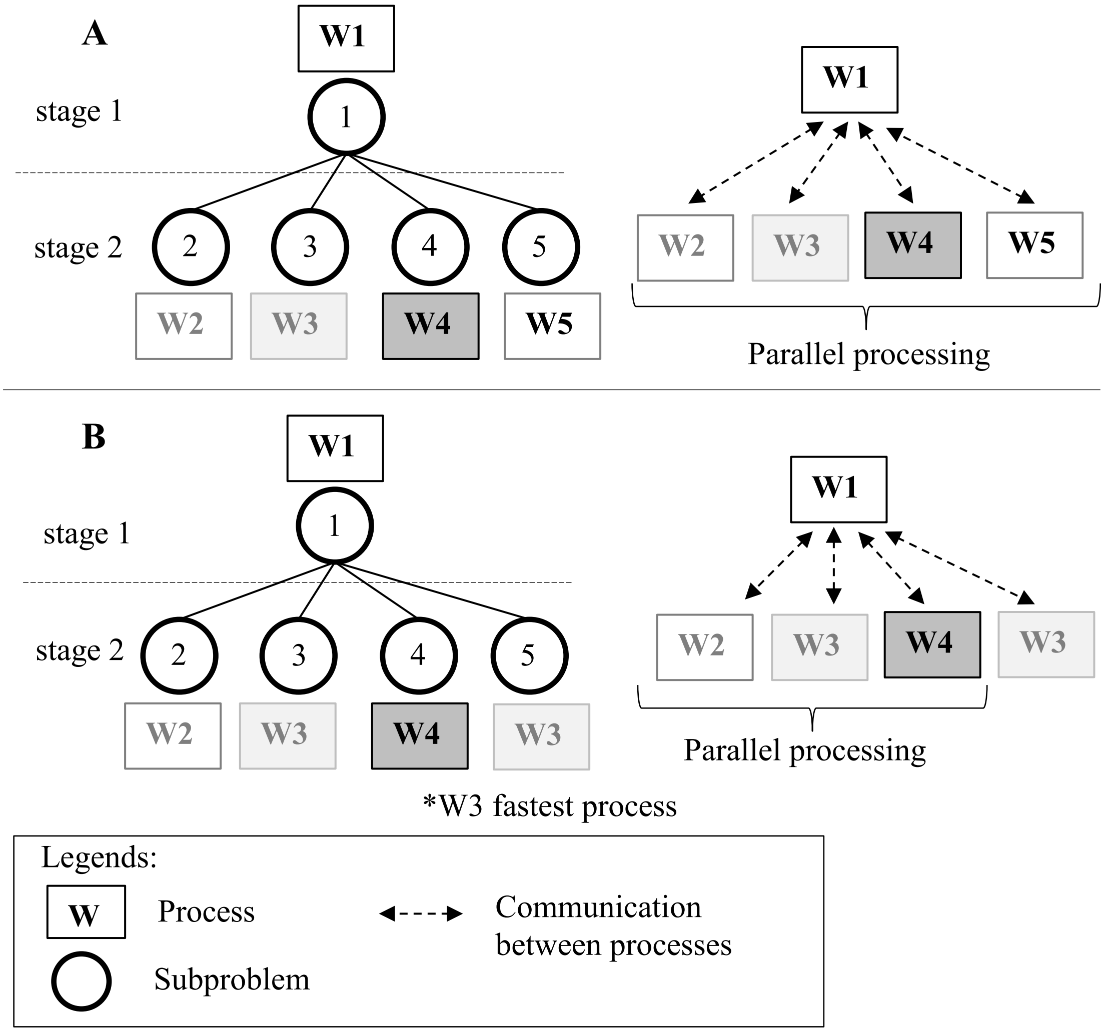
Number of	**LS**			**ELD**		
oracle calls	$\underline{z}$ (R$/h)	$\overline{z}$ (R$/h)	v_1 (hm^3)	$\underline{z}$ (R$/h)	$\overline{z}$ (R$/h)	v_1 (hm^3)
1	0.00	18,795.64	744.00	0.00	18,795.64	744.00
2	12,545.64	15,097.22	1,046.40	12,545.63	14,896.83	996.00
3	14,472.22	15,071.43	895.20	14,896.83	14,896.83	-
4	14,771.83	14,896.83	967.68	-	-	-
5	14,896.83	14,896.83	997.92	-	-	-

Source: Author.

4.5 TWO-STAGE STRUCTURE WITH PARALLEL COMPUTING

For the two-stage decomposition, a parallel processing strategy is implemented. In this setting, there is a master process that has two objectives: (*i*) solving the first-stage subproblem obtaining a trial point x_1^k and (*ii*) coordinating other processes (denominated slaves) to compute the second-stage subproblems at the point x_1^k. The slave processes transfer $Q(x_1^k)$, β_2^k, and α_2^k to the master process and wait until a new task is assigned. Figure 25 presents a problem that is decomposed into 5 subproblems with two parallel processing conditions: (A) the number of processes equals the number of subproblems and (B) the number of processes is smaller than the number of subproblems. In (A), the oracle time is given by the slower slave process's required time. On the other hand, the oracle time can be longer in case (B) since it is necessary to wait for the fastest process (W2, W3, or W4) to finish its first task. Then, subproblem 5 is allocated to the fastest one (in the example, consider W3 as the fastest process). Computational resources are shared between processes only when second-stage subproblems are computed. To compute the first-stage MILP, all computing resources are available to the master process.

Figure 25 – Two-stage parallel computing structure.



Source: Author.

5 NUMERICAL ASSESSMENT

This section presents the computational results obtained with the solution methods described in Chapter 4. For that, we use an instance of the Brazilian MTGS problem with a two-month planning horizon. As presented in Chapter 3, the discretization considers hourly periods in the first week, weekly periods until completing the first month, and a monthly period for the second month. The problem includes 152 hydro plants, 135 thermal plants, 4 subsystems, and 4 power deficit levels. For subproblems with weekly and monthly periods, 3 power load levels are considered. Subproblems with hourly time-steps include the classical thermal UC constraints. In the 25 hydro plants, which represent 70% of the hydro capacity, a concave piecewise HPF with 3 hyperplanes per plant is considered. For other plants, constant productivity modeling is assumed. At the end of the planning horizon, the final water value is obtained from a five-year LTGS problem with monthly decisions and the same system configuration. Table 6 presents the resulting number of variables and constraints concerning the time-step subproblems. The weekly/monthly subproblem size is a function of the number of nodes |N|. Also, this work considers scenario trees with non-common samples. The scenario tree generation is handled via an independent model[8] and a river basin representation (LIMA; POPOVA; DAMIEN, 2014), considering the inflows' historical data.

Table 6 – Dimension of the subproblems[9].

Δp	Number of variables (NV)	Number of constraints (NC)
hourly	162,960 (68,040 binary)	160,104
Weekly/monthly	1555 |N|	565 |N|

Source: Author.

We have used servers Xeon Servers CPU with 3.47 GHz, 32 GB RAM, and 24 cores. All MILPs and MIQPs are computed using Gurobi, called from environment C++. For the two-stage decomposition methods, a parallel processing strategy is used to compute the second-stage subproblems. The results are divided as follows: initially, several cases are presented to compare the two and the multi-stage decompositions' performance. The idea is to assess the regularization effect obtained from the construction of tighter cuts per iteration. A

[8] We refer readers to (BELTRÁN, 2015) Section 4.1 for details on an inflow generation independent model.

[9] Volume variables for run-of-river hydro plants are not considered. Also, hydro plants with a constant productivity modeling are represented directly in the load requirement constraints by $CP_2 \times q_h$.

computational analysis between the L-Shaped and ELD algorithms is carried out using single and multi-cut versions.

5.1 TIGHT-CUT EFFECT

The way in which a problem is decomposed impacts the solution method performance. As discussed in Section 2.3, the smaller the number of decomposition stages, the tighter the obtained cut per iteration. Table 7 reports the results of solving 5 scenario trees (only the inflow values change) with 6S and 2S decompositions. In this test, the water delay time is only considered in the first week. The table also presents two kinds of tree structure, which results in 400 and 2,700 scenarios. The number of variables (NV) and constraints (NC) is reported, considering the UC in the first stage. The upper bound, the number of iterations, the time per iteration, and the total time are reported in rows z^*, *ite*, t_{ite}, t_{sol}, respectively. A tolerance $tol = 10^{-6}$ is considered for the convergence of all the cases. The resulting problems are solved by applying Algorithm 1-NBD. On the other hand, Table 8 presents the number and dimension of subproblems per iteration required by 6S and 2S decompositions.

Table 7 – Comparison between 6S and 2S decompositions using the NBD algorithm.

Tree Structure	**Parameter**	**Decomposition**									
		6S					**2S**				
		Scenario tree					**Scenario tree**				
		1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
1×10×2×2×2×5 : 400 NV=1,018,210 (68,040 binary) NC=470,854	z^* (10^{11} R$)	1.082	1.018	1.021	0.997	0.995	1.082	1.018	1.021	0.997	0.995
	ite	452	470	424	537	417	25	39	31	30	27
	t_{ite} (h)	0.006	0.006	0.005	0.007	0.005	0.012	0.014	0.014	0.012	0.013
	t_{sol} (h)	3.49	3.54	3.19	4.21	3.06	0.21	0.30	0.27	0.16	0.22
1×10×3×3×3×10 : 2700 NV=4,983,460 (68,040 binary) NC=1,911,604	z^* (10^{11} R$)	1.003	1.024	1.002	0.989	1.017	1.003	1.024	1.002	0.989	1.017
	ite	529	551	442	488	423	32	38	34	35	28
	t_{ite} (h)	0.04	0.04	0.04	0.03	0.03	0.12	0.11	0.12	0.11	0.11
	t_{sol} (h)	19.13	19.60	15.65	16.85	14.20	3.76	4.17	4.07	3.81	2.97

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Table 8 – Number and dimension of subproblems with 6S and 2S decompositions.

Tree Structure	**Iteration step**	**Decomposition**			
		6S		**2S**	
		Number of subproblems	**Dimension**	**Number of subproblems**	**Dimension**
1×10×2×2×2×5 : 400	Forward	1	NV=162,960 (68,040 binary) NC=160,104	1	NV=162,960 (68,040 binary) NC=160,104
		550	NV=1,555 NC=565		
	Backward	150	NV=1,555 NC=565	10	NV=85,525 NC=31,075
1×10×3×3×3×10 : 2700	Forward	1	NV=162,960 (68,040 binary) NC=160,104	1	NV=162,960 (68,040 binary) NC=160,104
		3100	NV=1,555 NC=565		
	Backward	400	NV=1,555 NC=565	10	NV=482,050 NC=175,150

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

The results indicate superior performance of the two-stage decomposition with (on average) 85% time reduction compared to the other decomposition. The critical aspect is the number of iterations that the algorithm required to approximate the cost-to-go functions. Note that the 6S decomposition requires several hundred iterations, whereas 2S decomposition only needs tens. This iteration reduction confirms the significant benefit of obtaining a tight cut at each iteration. In this sense, the greater computational effort to compute 2S subproblems is compensated for the obtained tight cut. The next section presents the NBD and Level methods' performance in a 2S setting, assessing the construction of single and multi-cuts. We recall that the NBD in the two-stage setting is the L-Shaped method (VAN SLYKE; WETS, 1969).

5.2 L-SHAPED VERSUS ELD VIA SINGLE AND MULTI CUTS

As previously presented, the tightness effect of a 2S decomposition provides a significant running time reduction. This section presents a comparative analysis between the LS and ELD algorithms, including single and multi-cut versions. The idea is to further improve the 2S performance by applying the ELD regularization technique. Regarding the ELD algorithm, three premises are taking into account:

(i) A solver can present a high burden while identifying a MILP infeasibility. Therefore, subproblem (4.7) is always solved to compute its optimal value as $\underline{z}^k$

in Algorithm 3, ensuring the ELD master problem's feasibility. The computational burden associated with the ELD first-stage subproblem is necessarily greater than the LS one;

(ii) The stability center in (4.15) is set as the state variable (volumes) of the problem, and the l_1 norm is used to maintain the MILP properties.

(iii) The time required to solve (4.15) is limited to 100 seconds (approximately 4 times the required time (4.7)). When this time is exceeded, x^k is defined as the best solution obtained by the solver.

We set $\kappa = 0.7$ to update the level parameter. Moreover, this work tests a combination of methods LS and ELD. The idea is to obtain LS iterates in the first part of the optimization process, and ELD iterates nearby the convergence. The ELD subproblem's extra effort in the first iterations and the LS instability in the last iterations can be avoided. The assessed methods are defined as follows:

- LS: L-Shaped described in Algorithm 1;
- ELD: Extended level decomposition of Algorithm 3, with state variables as the stability center of (4.15) using l_1 norm;
- LS-ELD: obtains L-Shaped iterates computing (4.7) if gap > $3\times10^{-5}\cdot\overline{z}^k$ [10]; otherwise, solve (4.15) to achieve an ELD iterate, using state variables as the stability center and l_1 norm.

Three structures and four scenario trees per tree structure are considered. Table 9 reports the upper bound $\overline{z}$, iterations *ite*, the average time to compute the first-stage t_{1S}, the average time of the oracle t_{2S}, and the total time t_{sol} for the single and multi-cut versions of LS, ELD, and LS-ELD, respectively. For all the cases, convergence is achieved using a tolerance $tol = 10^{-6}$.

[10] With this value, in this test, approximately half the iterates are LS, while the rest are ELD.

Table 9 –Computational results.

	Tree Structure: 1×10×5×5×5×5: 6250 scenarios – 7811 nodes – NV=12,307,510 (68,040 binary) – NC=4,412,650											
Parameter	**Scenario tree 1**						**Scenario tree 2**					
	Single-cut			**Multi-cut**			**Single-cut**			**Multi-cut**		
	LS	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**
$\overline{z}\ (\times 10^{10}\,\mathrm{R\$})$	9.2919	9.2919	9.2919	9.2919	9.2919	9.2919	9.2820	9.2820	9.2820	9.2820	9.2820	9.2820
ite	26	24	24	17	16	11	39	27	28	18	17	13
t_{1S} **(h)**	0.0066	0.0230	0.0150	0.0061	0.0175	0.0077	0.0069	0.0179	0.0163	0.0067	0.0184	0.0101
t_{2S} **(h)**	0.1288	0.1290	0.1295	0.1300	0.1308	0.1277	0.1331	0.1334	0.1344	0.1325	0.1348	0.1323
t_{sol} **(h)**	3.52	3.65	3.47	2.31	2.37	1.49	5.46	4.08	4.22	2.51	2.61	1.86
Parameter	**Scenario tree 3**						**Scenario tree 4**					
	Single-cut			**Multi-cut**			**Single-cut**			**Multi-cut**		
	LS	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**
$\overline{z}\ (\times 10^{10}\,\mathrm{R\$})$	9.8515	9.8515	9.8515	9.8515	9.8515	9.8515	9.3813	9.3813	9.3813	9.3813	9.3813	9.3813
ite	20	17	17	13	12	13	18	20	14	15	15	10
t_{1S} **(h)**	0.0058	0.0152	0.0091	0.0056	0.0153	0.0100	0.0059	0.0159	0.0081	0.0059	0.0178	0.0103
t_{2S} **(h)**	0.1380	0.1409	0.1422	0.1407	0.1395	0.1394	0.1363	0.1350	0.1343	0.1375	0.1362	0.1353
t_{sol} **(h)**	2.88	2.78	2.56	1.90	1.85	1.94	2.56	3.02	2.00	2.15	2.45	1.44
	Tree Structure: 1×100×1×1×1×50: 5000 scenarios – 5401 nodes – NV=8,397,000 (68,040 binary) – NC=3,051,000											
Parameter	**Scenario tree 1**						**Scenario tree 2**					
	Single-cut			**Multi-cut**			**Single-cut**			**Multi-cut**		
	LS	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**
$\overline{z}\ (\times 10^{10}\,\mathrm{R\$})$	9.4713	9.4713	9.4713	9.4713	9.4713	9.4713	9.4713	9.4713	9.4713	9.4713	9.4713	9.4713
ite	29	23	19	14	14	13	31	21	17	13	14	12
t_{1S} **(h)**	0.0069	0.0200	0.0100	0.0081	0.0243	0.0132	0.0059	0.0164	0.0105	0.0082	0.0195	0.0116
t_{2S} **(h)**	0.1479	0.1481	0.1490	0.1471	0.1475	0.1462	0.1451	0.1447	0.1471	0.1449	0.1486	0.1480
t_{sol} **(h)**	4.49	3.87	3.02	2.17	2.41	2.09	4.69	3.37	2.67	1.99	2.38	1.93
Parameter	**Scenario tree 3**						**Scenario tree 4**					
	Single-cut			**Multi-cut**			**Single-cut**			**Multi-cut**		
	LS	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**
$\overline{z}\ (\times 10^{10}\,\mathrm{R\$})$	9.4332	9.4332	9.4332	9.4332	9.4332	9.4332	9.4656	9.4656	9.4656	9.4656	9.4656	9.4656
ite	21	19	19	13	12	12	21	17	16	16	16	12
t_{1S} **(h)**	0.0058	0.0147	0.0104	0.0082	0.0183	0.0116	0.0059	0.0143	0.0083	0.0087	0.0208	0.0161
t_{2S} **(h)**	0.1529	0.1516	0.1532	0.1540	0.1541	0.1521	0.1760	0.1752	0.1776	0.1525	0.1520	0.1514

t_{sol} (h)	3.33	3.16	3.14	2.11	2.07	1.96	3.82	3.22	2.96	2.58	2.76	2.02
	Tree Structure: 1×5×2×2×2×50: 2000 scenarios – 2076 nodes – NV=3,389,585 (68,040 binary) – NC=1,332,479											
Parameter	**Scenario tree 1**						**Scenario tree 2**					
	Single-cut			**Multi-cut**			**Single-cut**			**Multi-cut**		
	LS	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**
$\overline{z}$ **($\times 10^{10}$R$)**	9.0853	9.0853	9.0853	9.0853	9.0853	9.0853	9.3015	9.3015	9.3015	9.3015	9.3015	9.3015
ite	34	20	24	17	15	14	30	24	20	18	17	13
t_{1S} **(h)**	0.0068	0.0223	0.0145	0.0058	0.0217	0.0095	0.0063	0.0200	0.0100	0.0059	0.0199	0.0103
t_{2S} **(h)**	0.1648	0.1676	0.1595	0.1639	0.1611	0.1663	0.1594	0.1418	0.1550	0.1472	0.1476	0.1480
t_{sol} **(h)**	5.83	3.79	4.17	2.89	2.74	2.45	4.97	3.87	3.30	2.76	2.85	2.06
Parameter	**Scenario tree 3**						**Scenario tree 4**					
	Single-cut			**Multi-cut**			**Single-cut**			**Multi-cut**		
	LS	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**
$\overline{z}$ **($\times 10^{10}$R$)**	9.5742	9.5742	9.5742	9.5742	9.5742	9.5742	9.4956	9.4956	9.4956	9.4956	9.4956	9.4956
ite	21	22	14	19	15	14	23	17	16	18	13	15
t_{1S} **(h)**	0.0058	0.0186	0.0086	0.0058	0.0138	0.0087	0.0063	0.0164	0.0101	0.0061	0.0155	0.0083
t_{2S} **(h)**	0.1636	0.1688	0.1599	0.1547	0.1568	0.1561	0.1550	0.1606	0.1537	0.1583	0.1515	0.1589
t_{sol} **(h)**	3.56	4.12	2.34	3.05	2.55	2.28	3.71	3.06	2.61	2.96	2.18	2.51

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

In general, the difference of the optimal upper bound via LS, ELD, and LS-ELD is less than 1×10^{-6} for all the cases. As expected, the t_{1S} is longer for ELD cases than LS, approximately 2.82 times, due to the extra effort to solve (4.15) at each iteration. For the LS-ELD strategy, the t_{1S} difference decreases 1.65 times (regarding the LS case) as the LS first-stage subproblem is computed in the first part of the optimization process. On the other hand, ELD and LS-ELD reduce by 12% and 23%, respectively, the number of iterations compared to LS. This regularization effect obtains a time reduction concerning the LS algorithm if the iterative extra first-stage burden of ELD techniques is less than the LS's extra oracle calls. The regularization benefit is proportional to the oracle computational burden, i.e., the more difficult the 2S subproblems, the more effective is the reduction of an iteration. In our cases, t_{2S} is on average 12 times longer than t_{1S}. The ELD method obtains time reductions in 58% of all the cases and LS-ELD in 96%.

The multi-cut version of all algorithms surpasses the single-cut performance. On average, 33% less of t_{sol} is required by the multi-cut approach. The number of extra variables and constraints required by the multi-cut version is irrelevant to the first-stage subproblem size. Also, more accurate information per iteration is obtained. Table 10 reports the time percentage difference t_{diff} between the multi and single-cut versions of LS, ELD, and LS-ELD cases. Negative values indicate superior multi-cut performance.

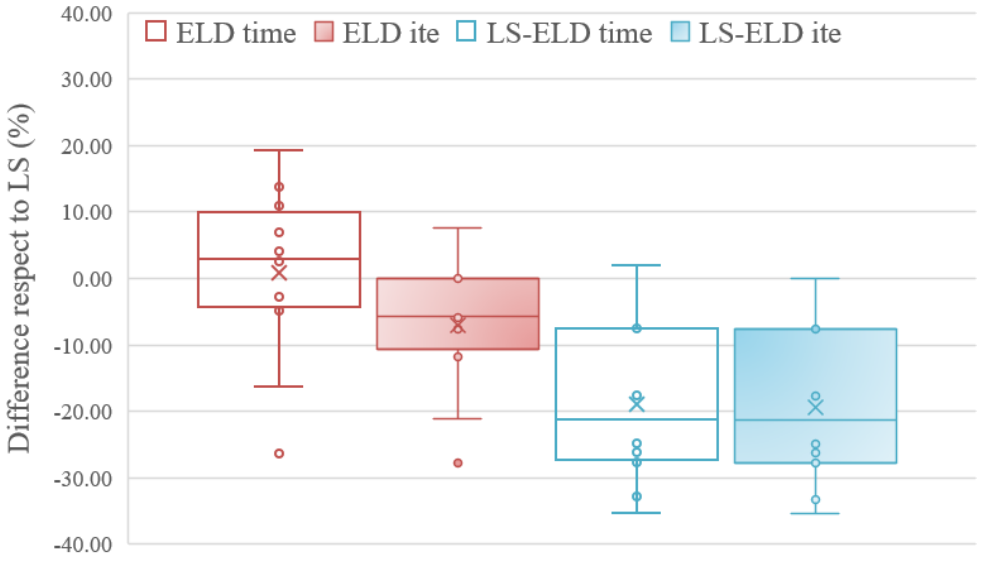
Table 10 – The time percentage difference between single and multi-cut versions.

	$t_{\text{diff}} = 100/[(t_{\text{sol\_Single}} - t_{\text{sol\_Multi}})\ t_{\text{sol\_Single}}]$ (%)											
Tree Structure	**Scenario Tree**											
	1			**2**			**3**			**4**		
	LS	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**	**LS**	**ELD**	**LS-ELD**
1×10×5×5×5×5	-34.3	-35.1	-56.9	-54.1	-36.2	-56.0	-33.8	-33.6	-24.2	-16.0	-19.0	-27.7
1×100×1×1×50	-51.6	-37.7	-30.9	-57.5	-29.6	-27.7	-36.7	-34.5	-37.4	-43.5	-14.4	-31.9
1×5×2×2×2×50	-50.5	-27.6	-41.2	-44.6	-26.4	-37.5	-14.3	-38.1	-2.6	-20.2	-16.6	-4.0

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

The regularized single-cut methods obtain larger computing times than the LS with multi-cuts for all the cases. From this point, the LS multi-cut results are used as the computational benchmark. Figure 26 presents the t_{sol} and *ite* percentage difference between multi-cut versions of ELD and ELD-LS regarding the LS multi-cut. Twelve comparative points per regularization technique are obtained. Negative values indicate reductions regarding the LS multi-cut.

Figure 26 – Computational comparison between LS, ELD, and LS-ELD multi-cut cases.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Note that LS-ELD multi-cut method surpasses the LS one performance, obtaining on average 19% t_{sol} reductions in 92% of all cases. The ELD multi-cut presents a trend in the iterations' reduction. Nevertheless, this reduction is insufficient to compensate for the extra effort to compute the regularized first-stage subproblem, obtaining a better t_{sol} only in 42% of the cases. The latter suggests the interest of getting trial points via subproblems with the low computational burden (e.g., LS subproblem) when the process is far away from convergence. After constructing a classical cutting-plane model, a regularization technique can be activated to concentrate the construction of the cost-to-go functions near the convergence region. This strategy has two useful advantages: (*i*) a $t_{1\mathrm{S}}$ extra effort is avoided in first iterations and, (*ii*) more accurate information is available to update ELD parameters as $\hat{x}^k$ and $f_{lev}{}^k$. The latter can explain why LS-ELD presents iteration reductions more significantly than the ELD. Considering that the ELD method updates $\hat{x}^k$ and $f_{lev}{}^k$ with information far away from the solution at the beginning of the optimization process, the initial ELD iterations concentrates in uninteresting regions of the cost-to-go functions. On the other hand, and similarly to the LS, the approach LS-ELD does not benefit from a good starting point.

As presented in Table 9, the required time to solve the second-stage subproblems $t_{2\mathrm{S}}$ is the most considerable computational effort of the two-stage decomposition methods. Considering that computational resources are limited, the computational effort $t_{2\mathrm{S}}$ increases in the proportion of two parameters: (*i*) the number of nodes of the subtrees and (*ii*) the quantity of second-stage subproblems computed in a parallel processing setting, since it determines the allocated resources per parallel process. Table 11 reports the computing time required to solve different subtree structures by increasing the number of two-stage subproblems solved concurrently. Note that if the number of nodes significantly increases, e.g., subtree structure of case 3, the computing time $t_{2\mathrm{S}}$ hinders the two-stage decomposition performance in an iterative process. To avoid this condition, the growth of the node's amount must be controlled by increasing the number of realizations only in the last SP periods or by applying scenario tree reduction techniques. This work considers that $t_{2\mathrm{S}}$ is suitable for the oracle if shorter than 0.15 hours.

Table 11 – Computation time of second-stage subproblems.

Case	Subtree structure	t_{2S} (h)		
		Number of two-stage subproblems solved concurrently		
		2	5	10
1	1×1×1×1×50: 54 nodes	0.0063	0.0080	0.0150
2	1×5×5×5×5: 781 nodes	0.0401	0.0736	0.1308
3	1×10×10×10×10: 11,111 nodes	3.0206	5.9807	7.2493

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

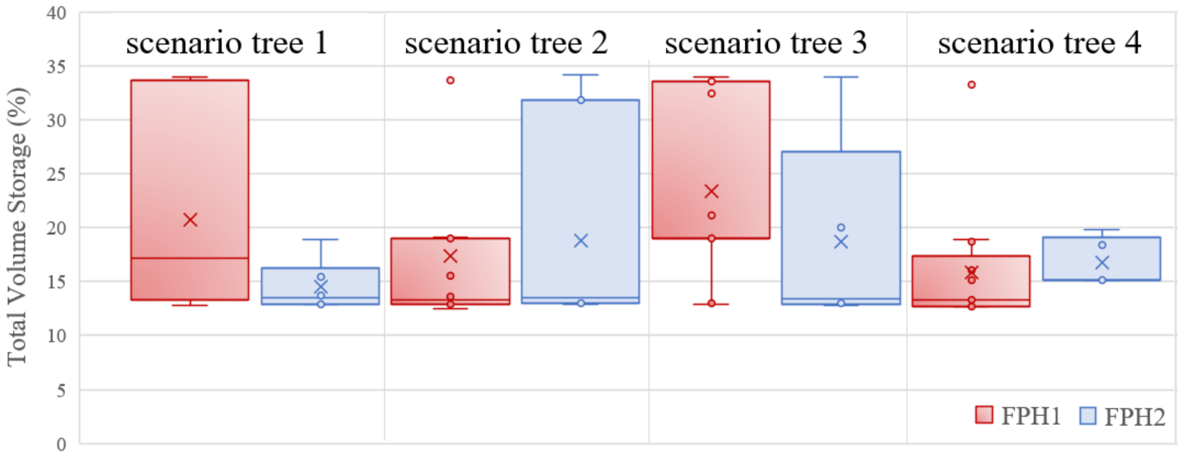
5.3 PRACTICAL APPLICATION OF THE LS-ELD MULTI-CUT

In previous sections, we presented the computational gains of employing tight cuts and LS-ELD multi-cut. Since one of the main objectives of the MTGS is to provide the best possible operating policy to the STGS problem, the CPU time reduction allows improvements on the MTGS modeling. For instance, we have included a piecewise HPF with an average of 9 hyperplanes for all the hydro plants in the first week of the optimization model. The idea is to generate trial points that are even more consistent with the STGS modeling. For this example, we use four scenario trees with a 1×10×5×5×5×5 structure. Figure 27 presents a comparison between the system storage volume at each iteration of the following modeling cases:

i. FPH1: 25 hydro plants, which represent 70% of the hydro capacity, with 3 hyperplanes for representing the concave piecewise HPF per plant;

ii. FPH2: 9 hyperplanes for representing the concave piecewise HPF per plant.

Figure 27 – Comparison of iterative storage volume of FPH1 and FPH2.



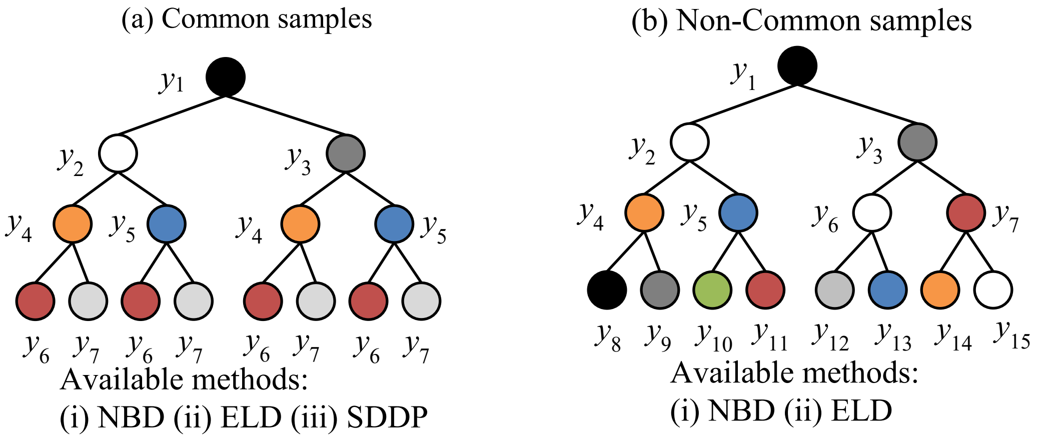
Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

An FPH modeling improvement impacts the hydro plants' operation, requiring different turbined outflow, spillage, and storage decisions to obtain a determined hydro production level. Other volume regions are visited among the iterative process, in which the cost-go-functions of the STGS-MTGS coupling are approximated. This operational policy is expected to value the water's future cost more accurately since it is constructed in more interesting regions for the STGS operation. Regarding the computational effort to include the FPH2, an increase on average 2.3 times to solve the first-stage problem is obtained. This extra effort can be compensated with the proposed LS-ELD strategy, obtaining on average 11% t_{sol} reductions regarding the LS for this more constrained problem.

5.4 TWO-STAGE DECOMPOSITION VERSUS SDDP METHOD

For the proposed MTGS problem, this section compares the performance of two-stage decomposition methods and the SDDP algorithm (indicated in Appendix B). As mentioned in Section 5.2, the SDDP algorithm can only solve scenario trees with common samples. Figure 28 presents an illustration of a scenario tree with common and non-common samples. Note that all nodes belonging to a period p have the same realizations y_n in the descendent nodes in the common sample setup. It should be noted that the scenario trees with non-common samples employed in Sections 5.1, 5.2, and 5.3 cannot be solved via the SDDP method.

Figure 28 – Scenario trees with common (a) and non-common (b) samples.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

This study performs the SDDP algorithm under the following premises:

- Parallel programming with 10 processes is applied to divide the number of scenarios per iteration;
- The SDDP algorithm stops when the lower bound obtained via LS-ELD multi-cut method is attained or a determined limit time of 6 hours is reached;
- To improve the SDDP performance, the cut selection of work (DE MATOS; PHILPOTT; FINARDI, 2015) is applied.

Table 12 presents the computational results of solving 3 scenario trees with common samples via LS and LS-ELD multi-cut. Similar to the non-common sample case in Section 5.2, LS-ELD multi-cut method presents a higher performance with time reductions on average 32% regarding the LS multi-cut. Therefore, in this study, the lower bounds $\underline{z}_{\text{LS-ELD}}$ obtained via LS-ELD multi-cut, equal to the upper bound, are used as a benchmark of the SDDP performance.

Table 12 – Two-stage decomposition performance solving a scenario tree with a common sample.

	Tree Structure: 1×10×5×5×5×5: 6250 scenarios – 7811 nodes – NV=11,915,492 (68,040 binary) – NC=3,495,652					
Parameter	**Scenario tree 1**		**Scenario tree 2**		**Scenario tree 3**	
	Multi-Cut		**Multi-Cut**		**Multi-Cut**	
	LS	**LS-ELD**	**LS**	**LS-ELD**	**LS**	**LS-ELD**
$\underline{z}$ **(×10^{10}R$)**	10.1860	10.1860	9.0933	9.0933	9.4956	9.4956
ite	25	15	19	12	24	18
t_{1S} **(h)**	0.0065	0.01126	0.0074	0.0113	0.0064	0.0108
t_{2S} **(h)**	0.1296	0.1279	0.1222	0.1234	0.1173	0.1166
t_{sol} **(h)**	3.40	2.07	2.46	1.61	2.97	2.30

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Concerning the SDDP performance, Table 13 reports the $\underline{z}$ obtained via SDDP and the percentage difference $\underline{z}_{diff}$ regarding the lower bound $\underline{z}_{\text{LS-ELD}}$, using the same CPU time than the LS-ELD multi-cut convergence (reported in Table 12), i.e., 2.07, 1.61, and 2.30 hours for scenario trees 1, 2 and 3 respectively. Negative values indicate worse SDDP performance. The iterations, the average time per iteration, and the total time until attaining the stopping criterion are reported in rows *ite*, t_{ite}, and t_{tot} in that order. Four cases of the number of scenarios per iteration with resampling are considered: 10, 20, 50, and 100.

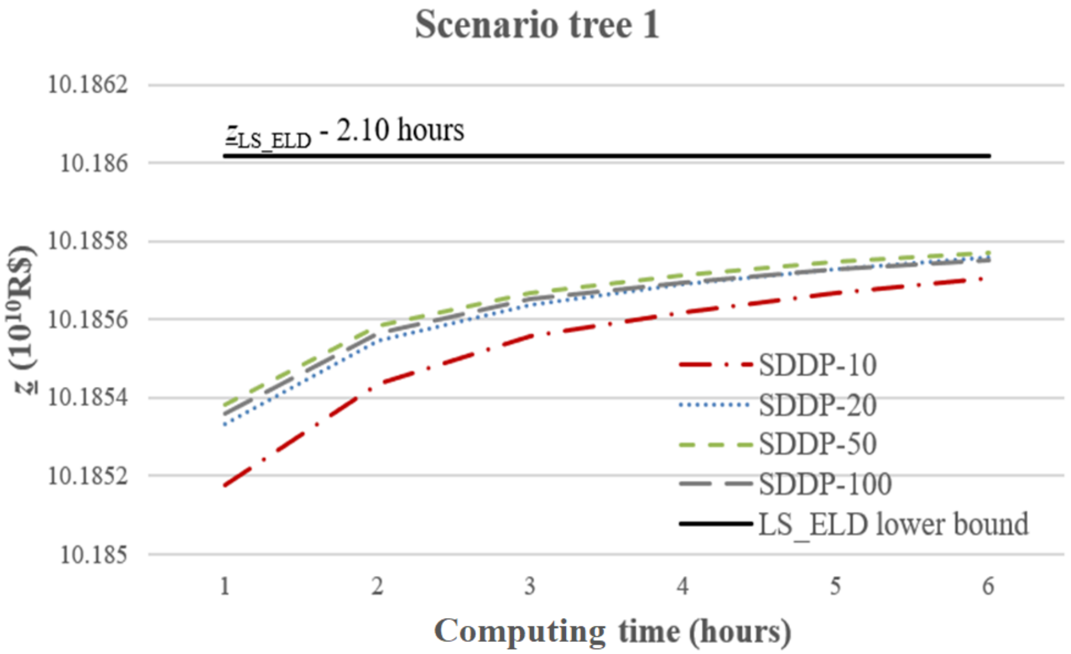
Table 13 –SDDP performance.

	Tree Structure: 1×10×5×5×5×5: 6250 scenarios – 7811 nodes – NV=11,915,492 (68,040 binary) – NC=3,495,652											
Parameter	**Scenario tree 1**				**Scenario tree 2**				**Scenario tree 3**			
	Number of scenarios per iteration				**Number of scenarios per iteration**				**Number of scenarios per iteration**			
	10	**20**	**50**	**100**	**10**	**20**	**50**	**100**	**10**	**20**	**50**	**100**
$\underline{z}$ **(×10^{10}R$)**	10.1855	10.1856	10.1856	10.1856	9.0931	9.0932	9.0932	9.0932	9.4952	9.4953	9.4953	9.4953
$\underline{z}_{diff}$ **(×10^{-3}%)**	-5.55	-4.53	-4.17	-4.29	-2.29	-1.73	-1.48	-1.67	-4.20	-3.38	-3.01	-3.60
ite	552	390	202	113	500	358	189	107	554	393	195	111
t_{ite} **(h)**	0.0109	0.0154	0.0297	0.0538	0.012	0.0168	0.0319	0.0577	0.0108	0.0153	0.0304	0.0546
t_{tot} **(h)**	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0

Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

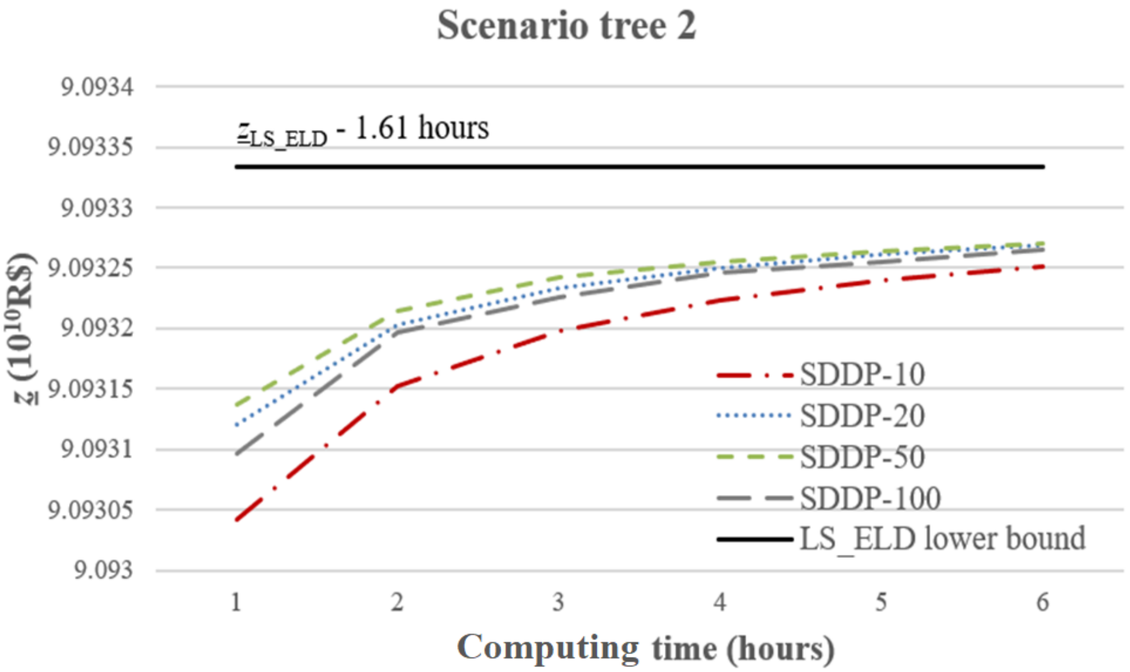
Note that, employing the same CPU time, the $\underline{z}$ obtained via SDDP is lower than $\underline{z}_{\text{LS-ELD}}$ in all cases. For instance, scenario tree 1 with 50 scenarios per iteration presents a $\underline{z}$ percentage difference of -4.53×10^{-3}, in other words -4.26×10^{6} R$. Such difference represents a continuous thermal generation over 1 week of 253 MW with a unitary variable cost of 100 R$/MWh. When the number of scenarios per iteration increases, more state variables are considered to construct the cost-to-go functions; however, the SDDP computational effort per iteration gradually increases. For this study, the best SDDP results are obtained considering 50 scenarios per iteration. When setting 100 scenarios, t_{ite} increases on average 80% without any SDDP performance gain. Figure 29, Figure 30, and Figure 31 present the lower bound progress obtained via the SDDP method compared to $\underline{z}_{\text{LS-ELD}}$ for scenario trees 1, 2, and 3. None of the SDDP cases reach the $\underline{z}_{\text{LS-ELD}}$ in 6 hours, confirming the cut tightness effect obtained from two-stage decomposition methods.

Figure 29 – SDDP performance – Scenario tree 1.



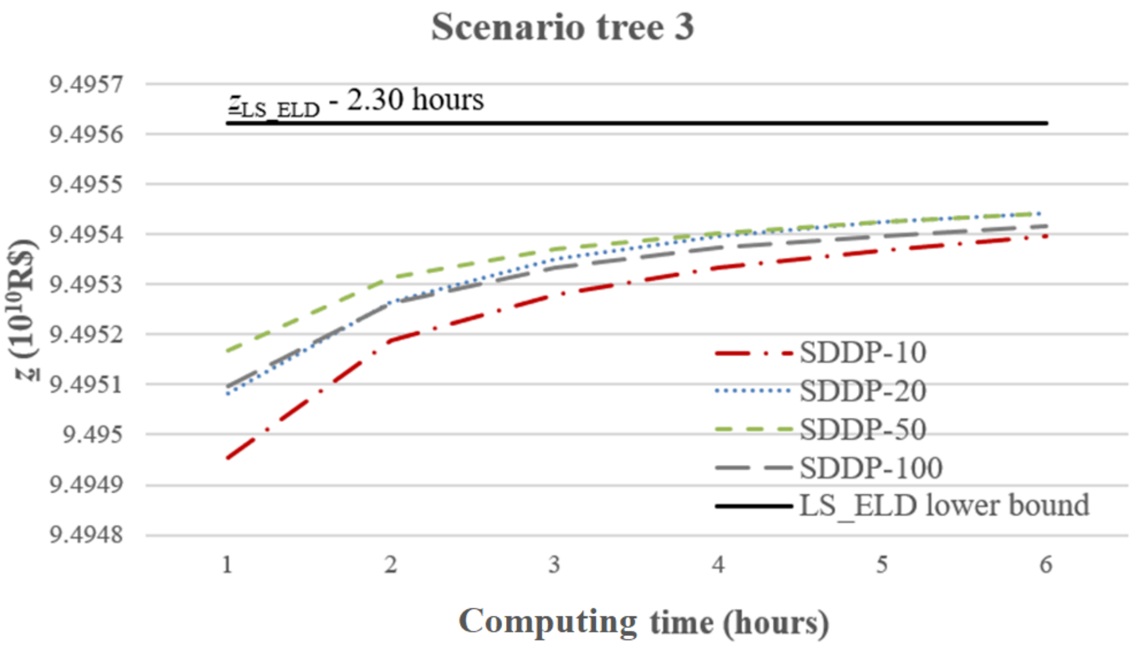
Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Figure 30 – SDDP performance – Scenario tree 2.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Figure 31 – SDDP performance – Scenario tree 3.



Source: (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

5.5 OPTIMIZATION PROCESS: EFFECT OF THE THERMAL UC CONSTRAINTS IN THE FIRST-STAGE

In this section, we compare the first-stage decision computed by the following cases:

(i) Formulation 1: considering a load level approach for the first week;

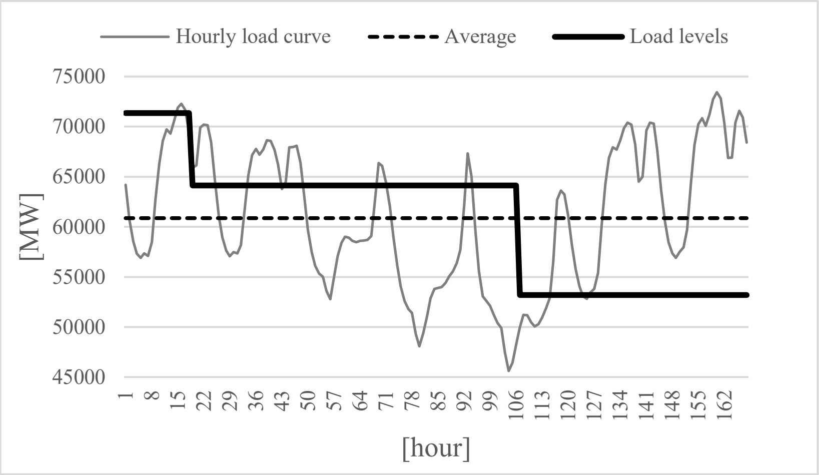
(ii) Formulation 2: solving the MTGS problem with an hourly discretization in the first week including the thermal UC constraints.

Moreover, the following conditions are considered:

- The same scenario trees;
- for $t = 1$, the hourly inflow and curve demand for 168 hours are used to obtain the equivalent load level values.

This work considers the load curve presented in Figure 32, which corresponds with the week's hourly load from 30/12/2017 to 05/01/2018 (ONS, 2018).

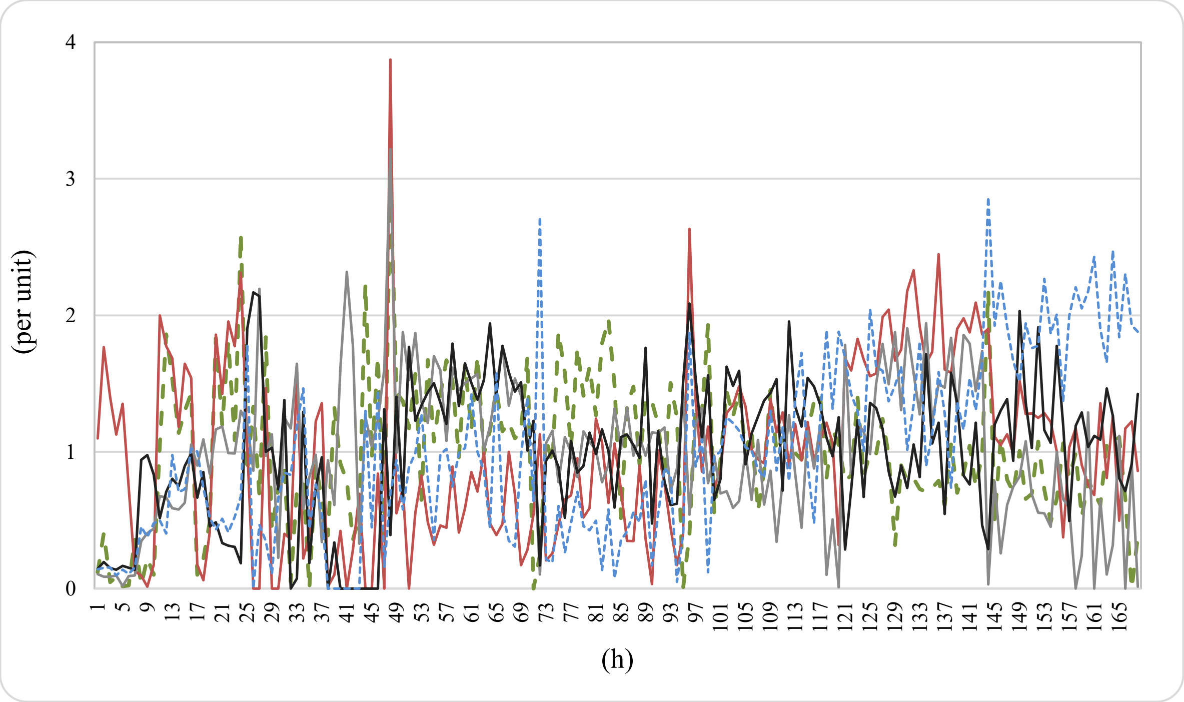
Figure 32 – Hourly load (MW).



Source: Author.

The first-stage weekly inflow is proportionally distributed per hour, considering the week's historical daily inflow from 30/12/2017 to 05/01/2018 (ONS, 2018). For instance, Figure 33 shows the inflow variability considered during a week for 5 hydro plants.

Figure 33 – Inflow value per unit along first-stage



Source: Author.

Table 14 reports the scenario tree considered by the L-Shaped method with a relative tolerance of 10^{-4} (%).

Table 14 – Scenario tree characteristics.

Tree structure	**#scenarios**	**#nodes**	**#stages**
1×10×3×4×4×10	4,800	5,441	6 (weekly stages in the first month and a monthly in the last stage)

Source: Author.

Table 15 presents the optimization procedure results for different conditions of the initial volume and first-stage inflow. The latter is presented as a percentage of the original first-stage inflow realization. The final volume, objective function value, optimality gap, and the number of iterations are reported in columns v_1, $\bar{z}^k$, *gap* and #*k*.

Table 15 – Optimization process results.

First-stage formulation	**First-stage inflow (%)**	v_0 **(%)**	v_1 **(%)**[11]	$\bar{z}^k$ **(1×10^{10} R$)**	***gap*** **(1×10^{-6})**	***k***
Load level	150	70	74.40	1.1823	0.83	70
UC			74.27	1.1902	0.92	60
Load level	150	50	55.33	3.0012	0.70	29
UC			55.11	3.0185	0.95	23
Load level	150	30	35.97	5.6414	0.25	4
UC			35.73	5.6605	0.50	5
Load level	100	70	71.84	1.4075	0.81	28
UC			71.61	1.4112	0.83	26
Load level	100	50	52.69	3.4419	0.65	17
UC			52.48	3.4503	0.71	13
Load level	100	30	33.20	6.1088	0.88	9
UC			33.04	6.1175	0.10	9
Load level	70	70	70.47	1.5569	0.94	47
UC			70.45	1.5595	0.81	50
Load level	70	50	50.88	3.7262	0.84	22
UC			50.70	3.7319	0.88	18
Load level	70	30	31.19	6.4075	0.61	7

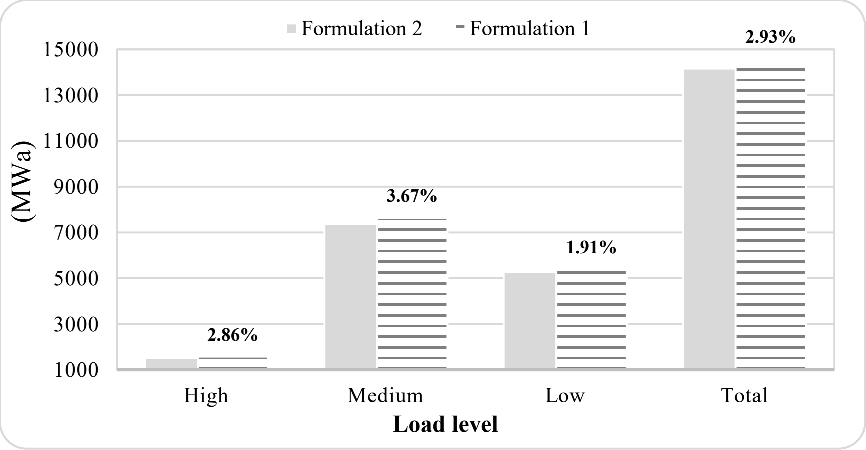
[11] For formulation 2, v_1 corresponds the final volume at hour 168.

First-stage formulation	**First-stage inflow (%)**	v_0 **(%)**	v_1 **(%)**[11]	$\overline{z}^k$ **(1×10^{10} R$)**	***gap*** **(1×10^{-6})**	k
UC			31.13	6.4138	0.84	11
Load level	50	70	69.11	1.6661	0.89	63
UC			69.10	1.6681	0.83	48
Load level	50	50	49.53	3.9302	0.77	20
UC			49.54	3.9343	0.95	16
Load level	50	30	29.97	6.6195	0.79	6
UC			30.06	6.6234	0.17	5

Source: Author.

Formulation 2 presents an average increase cost of 0.24% and an average final volume decrease of 0.22% concerning formulation 1. Modeling aspects as the water delay time, load curve, and thermal plants' constraints lead to more realistic dynamics of the system, such as a gradual increase/decrease of the thermal generation or a real availability of the water at a certain period in a reservoir. In this sense, formulation 1 has a trend of optimistic decisions since all thermal capacity is available at any time, and the outflow water is immediately available in downstream plants. These conditions are not necessarily true to satisfy the load requirements. In general, formulation 1 uses more thermal generation in the first-stage, and consequently the future cost associated with the LTGS problem and the optimal cost decrease. Figure 34 presents the thermal generation per load level, considering inflow 50% and v_0 = 50%. The percentage increase of the thermal generation associated with formulation 1 regarding formulation 2 is reported in the figure. The first-stage thermal generation supplied by formulation 1 is 2.93% higher than formulation 2.

Figure 34 – First-stage thermal generation per load level.

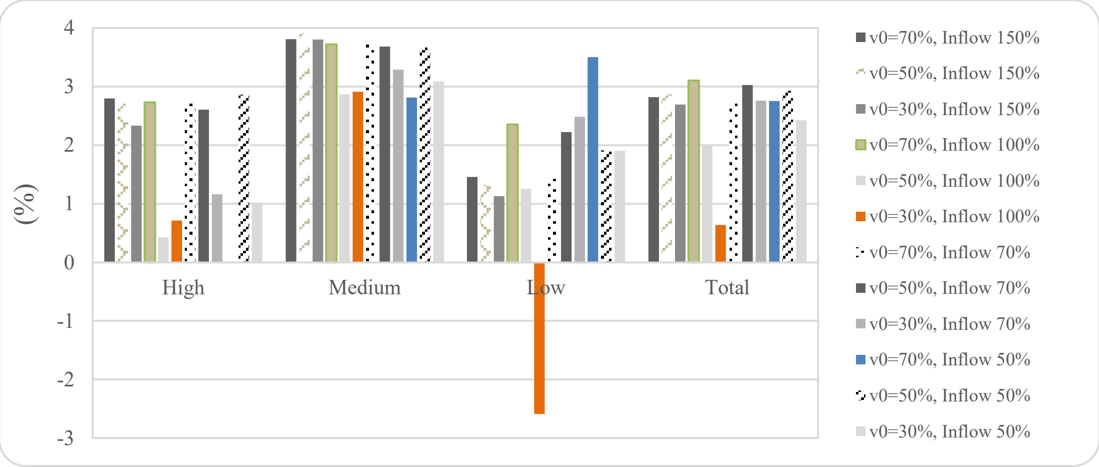


Source: Author.

For all cases, the percentage difference of first-stage thermal generation per load level between formulation 1 and 2 are presented in Figure 35, which are calculated as follows:

$$\%\text{diff} = \frac{(\text{Formulation 1 value} - \text{Formulation 2 value})}{\text{Formulation 2 value}} \times 100\%,$$

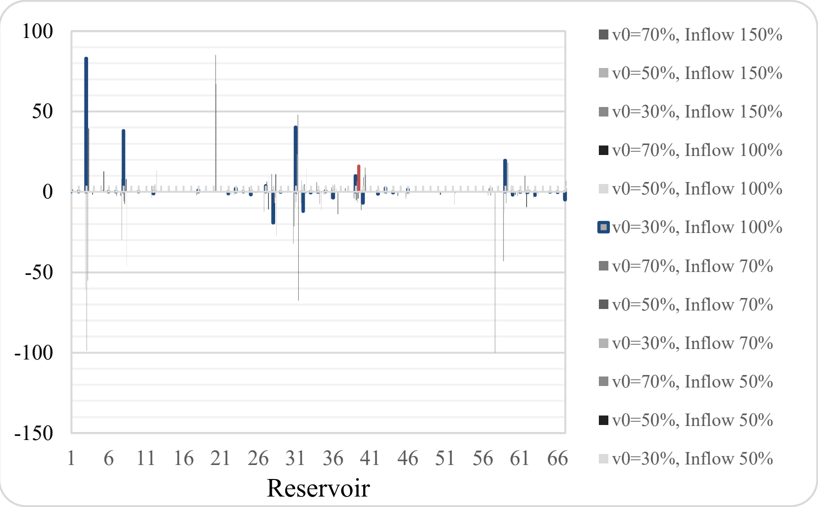
Figure 35 –Difference between first-stage thermal generation per load level.



Source: Author.

For several initial volume and inflow conditions, the load level modeling tends to use more thermal generation. This fact leads to construct the first-stage cost-to-go function in favorable volume regions. Nevertheless, these storage levels or water allocation may not be achieved in practice. Figure 36 presents the final volume difference in all reservoir of the system issued by solving formulation 1 and 2. Positive values indicate more storage by formulation 2; otherwise, formulation 1 saves more water.

Figure 36 –Final volume difference regarding the storage capacity.



Source: Author.

Significant water allocation differences between formulation 1 and 2 solutions are obtained. The thermal UC generation constraints and the water delay modeling time in the first stage can obtain a different valuation of each reservoir's water opportunity cost. As the main conclusion, a more detailed first-stage of the MTGS problem can help obtain more accurate cost-to-go functions, improving the guidelines that link the STGS problem. These results are tested in the next section with an out-of-sample simulation of the resulting operational policies.

5.6 OUT-OF-SAMPLE SIMULATION: EFFECT OF THE THERMAL UC CONSTRAINTS IN THE FIRST-STAGE

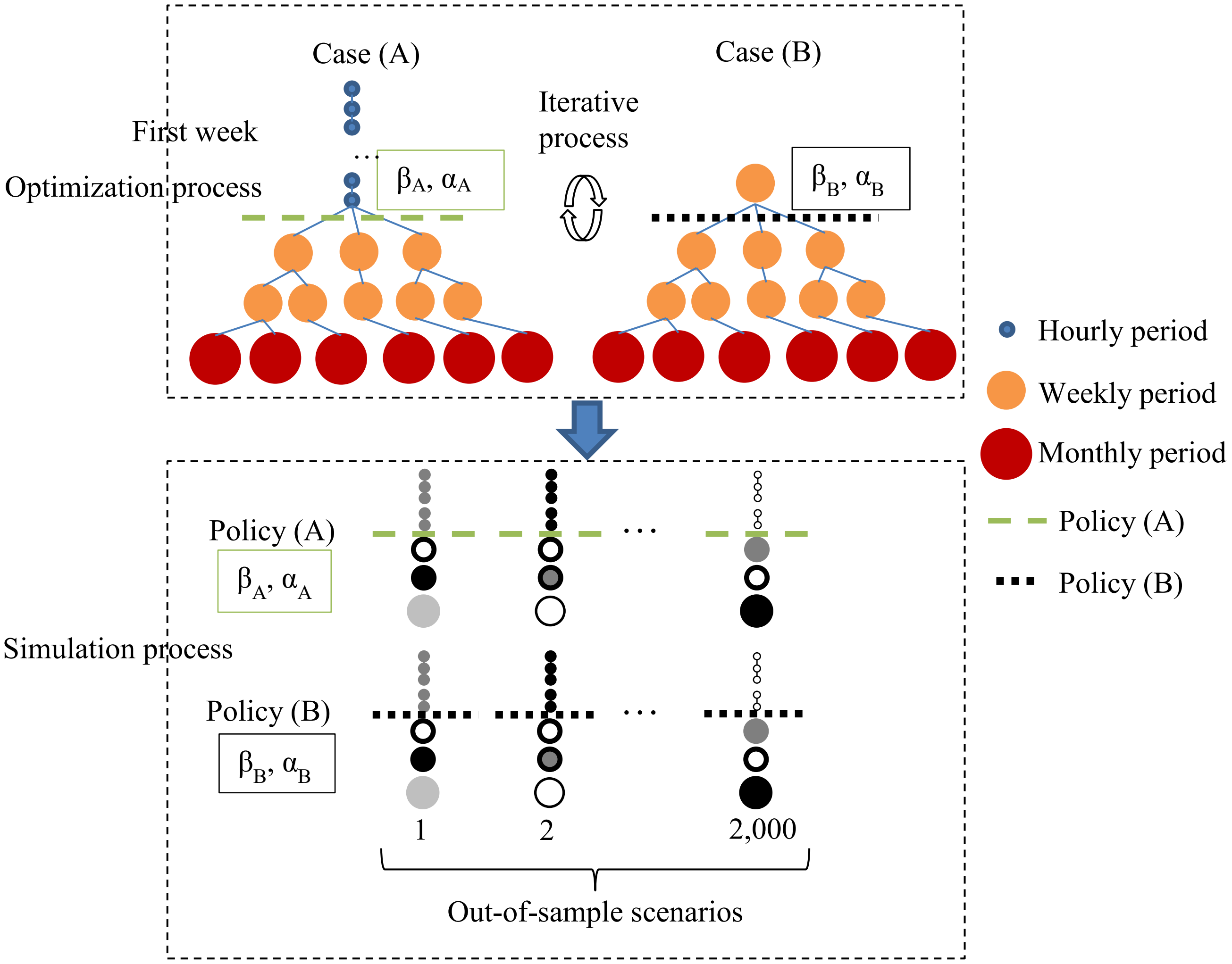
This section presents an out-of-sample simulation of different operational policies obtained from two first-week models: (A) an hourly discretization including the thermal UC constraints, (B) a weekly discretization with a load-level approach. The main characteristics of this study are presented below:

- The considered scenario tree in the optimization process has the following structure: 1×10×5×5×5×5: 6250 scenarios;

- The three different scenario trees are equally considered for both cases (A) and (B);
- An out-of-sample simulation of the resulting policies with the same 2,000 scenarios is carried out for cases (A) and (B).

This simulation process considers the first-week modeling with an hourly discretization and the UC constraints to ensure a fair comparison and a shared basis between the operational policies obtained from the different models. That is because UC modeling is more realistic. The simulation process couples the cost-to-go functions obtained by computing the MTGS for cases (A) and (B) at the end of the first week. The idea is to determine the performance of these operational policies. For that, a set of possible out-of-sample inflow realizations is considered. Figure 37 exemplifies the considered simulation process in this work.

Figure 37 – Simulation process.



Source: Author.

Table 16 and Table 17 report the expected operational total cost and the expected operational first-stage cost, respectively. For cases A and B, these results are obtained by simulating 3 different operational policies related to computing 3 different scenario trees. The simulation reveals that the cost-to-go functions in case B have a trend for a higher thermal operation in the first week, which leads to an increase of the first-stage cost (on average 1.4%) regarding case A. Despite this rise, the expected total operational cost decreases slightly, 0.02% on average. Also, Figure 38 presents a comparison of the total storage volume series for all the simulated scenarios. Note that the operational policies in cases A and B have a water use difference for wet scenarios. These results validate the hypothesis that the cost-to-go function's construction is carried out in different regions when different first-stage models are considered.

Table 16 – Simulation results – Expected operational total cost.

Average total cost (×10^{10}R$)			
Case	Scenario tree		
	1	2	3
A- UC	9.610	9.610	9.604
B- Load levels	9.609	9.607	9.601
difference (%)	-0.01	-0.03	-0.03

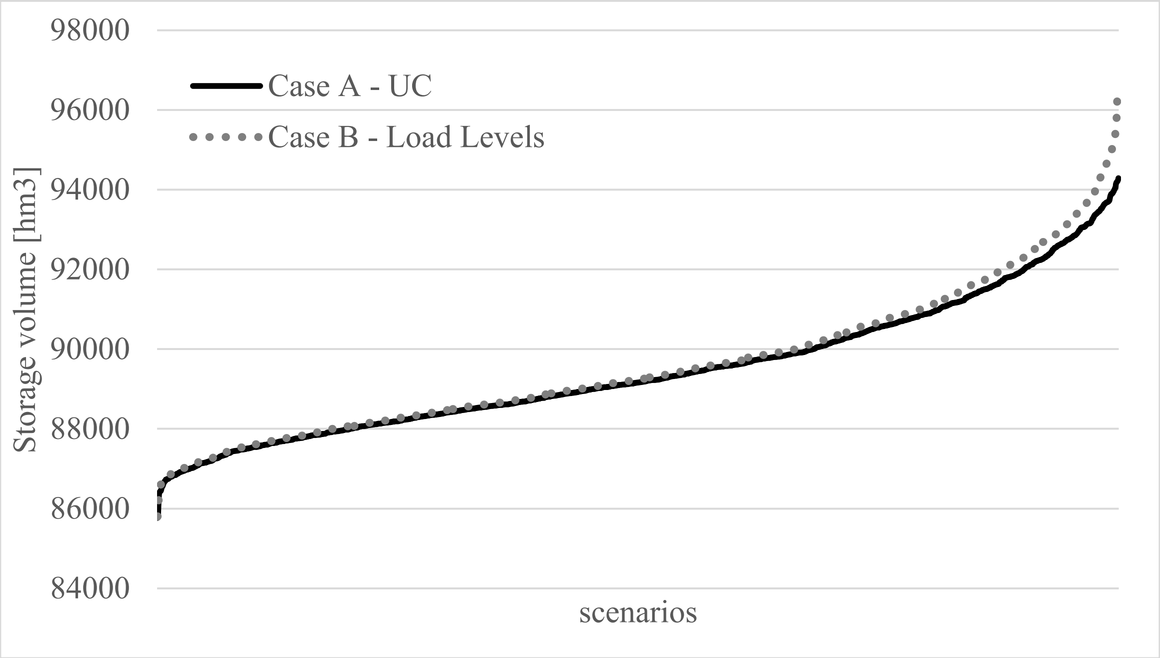
Source: Author.

Table 17 – Simulation results – Expected operational first-stage cost.

Average first-stage cost (×10^8R$)			
Case	Scenario tree		
	1	2	3
A- UC	1.8555	1.8779	1.6254
B- Load levels	1.8738	1.9056	1.6521
difference (%)	0.99	1.48	1.64

Source: Author.

Figure 38 – Simulation results - total storage volume series.



Source: Author.

This out-of-sample simulation only presents the differences in the static condition of the system. To assess the benefits of more detailed modeling, a rolling-horizon approach is required to observe the system's dynamic evolution. This process is not an easy task since a combined execution configuration of the LTGS and MTGS models is needed during an appropriate horizon. The required rolling-horizon execution is out of this work's scope and is left for future research.

6 CONCLUSIONS

In general terms, the STGS-MTGS coupling quality depends significantly on the MTGS problem's characteristics, especially on modeling details and on the scenario tree considered to represent the stochastic process. One of the ideas of this work is to make the modeling of the coupling stage (in our case, the first week of the MTGS) come closer to the modeling level of the STGS problem. In this way, the cost-to-go-functions obtained in the MTGS solution are better approximated regarding the STGS problem. For instance, the inclusion of the thermal UC constraints and the water delay time in the first week of the MTGS present more realistic conditions: reduced thermal generation flexibility (in the order of 3%) and real availability of the water at a certain period in each reservoir. These factors impact the valuation of the water opportunity cost for each reservoir and, consequently, the economic signal about the water's location to obtain a benefit in the future. A comparison between different models for the first-stage problem reveals a significant difference in the quantity and the location of the storage water.

This work considers a scenario tree for the MTGS problem that includes the inflow stochasticity from the second week of the planning horizon to improve the stochastic process representation. Indeed, this scenario tree structure results in a challenging large-scale optimization problem. To deal with this large problem, the first idea of this work was the use of Chebyshev centers in the NBD method by controlling the next iterates through a linear master subproblem. For the considered instances of the MTGS problem, the NBD-CC method did not perform better than the NBD method because of the inherent coupling with the LTGS problem. However, we found out that the Chebyshev approach is a promissory tool for solving other stochastic problems, such as the LTGS problem. These findings have been published in the journal paper (BELTRÁN et al., 2020).

We explore decomposing the resulting problem in different stages since the decomposition impacts the solution methodologies' performance. One of the most important conclusions of this work is the following: the proposed two-stage decomposition of the multi-period MTGS problem is advisable as long as the computational burden to solve the resulting subproblems (which are themselves smaller multi-periods problems) remains acceptable. Such a strategy results in tight cuts that accelerate the iterative process. For the proposed MTGS problem, the two-stage decomposition presents a higher performance regarding a multi-stage decomposition, obtaining time reductions in the order of 85%. The tight cut effect reveals a considerable reduction of the required number of iterations by the solution methods. In our case, a multi-stage decomposition requires in the order of several hundred iterations, whereas a two-stage decomposition only needs tens. In this sense, the extra effort to solve larger subproblems in the second stage is largely compensated by the obtained tight cuts and the reduction of the resulting number of subproblems per iteration. The two-stage decomposition performance was also tested using scenario trees with common samples, comparing the two-stage methods taking into account the SDDP algorithm behavior. The results reveal that the lower bound obtained from the SDDP does not attain the optimal value obtained from the two-stage methods (even using 3 times more computing time).

To assess the potential improvement of the two-stage decomposition performance, this work compares the use of single-cut and multi-cut versions of the cost-to-functions. The results disclose that the multi-cut version strengthens the tightness effect of the two-stage decomposition with time reductions in the order of 30% when compared to the single-cut version. The reason for this behavior is that, in the single-cut version, all the optimal simplex multipliers related to the second-stage subproblems are aggregated to generate one cut. On the other hand, the multi-cut version includes each multiplier individually, quantifying how a trial point impacts every second-stage subproblem. Naturally, the number of constraints used to represent the cost-to-go functions is higher in the multi-cut version. In our case, these extra constraints are not relevant due to the size of the MILP and, therefore, the extra first-stage computational burden does not compromise the solution method performance.

In the two-stage setting, the use of regularization techniques is straightforward. This work applies the level bundle method, which is known for having a good performance when dealing with binary variables. The results show that higher computational performance is achieved when a combination of methods LS and ELD is carried out. The idea is to apply the regularization technique when the iterative process is closer to the solution region, avoiding the LS method's possible oscillations. Time reductions in the order of 19% are observed. In practice, this computational gain can improve the MTGS modeling to construct the cost-to-go function in interesting regions for the STGS problem. Given that the solvers can spend a considerable time detecting a MILP infeasibility, this work recommends updating the level parameter with the LS master problem's optimal value, thereby ensuring the feasibility of the ELD master problem. A limited CPU time to solve the ELD master can be adopted in the same spirit to avoid spending a high computing time in one iteration. In this condition, in the worst case, the trial point used in the oracle can be the one from the LS master problem (provided the recommendation to deal with the ELD infeasibilities is adopted). We care to mention that the two-stage solution methods have a limitation when applied to the proposed MTGS: the computing time required in the second-stage subproblems might be excessive. The aforementioned effort depends on the subproblems size (according to the number of tree nodes), the parallel processes computing simultaneously, and the available computational resources. Also, concerning the two-stage versus multi-stage decompositions to solve the MTGS problem, the main results obtained in this Thesis have been published in (BELTRÁN; FINARDI; DE OLIVEIRA, 2021).

Finally, this work's next step would be to assess the proposed MTGS problem's benefits using a rolling-horizon study that couples all the chain scheduling models STGS, MTGS, and LTGS. Naturally, any modeling or methodological improvement, such as the stochastic representation of alternative power sources (such as wind or solar power), is bound to improve the STGS- MTGS coupling. Methodologically, the application of the cut tightness concepts can be explored in other solution methods such as the SDDP algorithm. Furthermore, by constructing unbalanced scenario trees yielding subproblems of different sizes for the proposed two-stage decomposition, the extended level decomposition can benefit from asynchronous management of slaves and master programs. This idea, investigated for UC problems in (COLONETTI; FINARDI; DE OLIVEIRA, 2020), can be particularly useful in the MTGS context due to the flexibility of choosing the subproblems size.

REFERENCES

BAYRAKSAN, G.; MORTON, D. P. Assessing solution quality in stochastic programs. **Mathematical Programming**, v. 108, n. 2, p. 495–514, 1 set. 2006.

BELLMAN, R. E. **Dynamic Programming**. [s.l.] Dover Publications, Incorporated, 2003.

BELTRÁN, F. **Redução de cenários via distância aninhada aplicada ao problema do planejamento da operação energética**. [s.l.] Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Elétrica, 2015.

BELTRÁN, F.; FINARDI, E. C.; DE OLIVEIRA, W; FREDO, G. L. M. Improving the performance of the stochastic dual dynamic programming algorithm using Chebyshev centers. **Optimization and Engineering**, 7 set. 2020.

BELTRÁN, F.; DE OLIVEIRA, W.; FINARDI, E. C. Application of Scenario Tree Reduction Via Quadratic Process to Medium-Term Hydrothermal Scheduling Problem. **IEEE Transactions on Power Systems**, v. 32, n. 6, p. 4351–4361, nov. 2017.

BELTRÁN, F.; FINARDI, E. C.; DE OLIVEIRA, W. Two-stage and multi-stage decompositions for the medium-term hydrothermal scheduling problem: A computational comparison of solution techniques. **International Journal of Electrical Power & Energy Systems**, v. 127, p. 106659, 1 maio 2021.

BEN-TAL, A.; NEMIROVSKI, A. Non-euclidean restricted memory level method for large-scale convex optimization. **Mathematical Programming**, v. 102, n. 3, p. 407–456, 1 jan. 2005.

BIRGE, J. R. Aggregation bounds in stochastic linear programming. **Mathematical Programming**, v. 31, n. 1, p. 25–41, 1 jan. 1985.

BIRGE, J. R.; LOUVEAUX, F. **Introduction to Stochastic Programming**. New York, NY: Springer New York, 2011.

BIRGE, J. R.; LOUVEAUX, F. V. A multicut algorithm for two-stage stochastic linear programs. **European Journal of Operational Research**, v. 34, n. 3, p. 384–392, 1 mar. 1988.

CARPENTIER, P. L.; GENDREAU, M.; BASTIN, F. **Midterm Hidro Generation Scheduling Under Uncertainty Using the Progressive Hedging Algorithm**. [s.l.] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, 2012.

CARPENTIER, P. L.; GENDREAU, M.; BASTIN, F. Managing Hydroelectric Reservoirs Over an Extended Horizon Using Benders Decomposition With a Memory Loss Assumption. **IEEE Transactions on Power Systems**, v. 30, n. 2, p. 563–572, mar. 2015.

COLONETTI, B.; FINARDI, E. C.; DE OLIVEIRA, W. A Mixed-Integer and Asynchronous Level Decomposition with Application to the Stochastic Hydrothermal Unit-Commitment Problem. **Algorithms**, v. 13, n. 9, p. 235, set. 2020.

CUNHA, S. H. F.; PRADO, S.; COSTA, J. P. **Modelling of the variable productivity of hydro plants based on a hydro production function**. . In: XII SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS (IN PORTUGUESE). Brazil: 1997

DA COSTA, J. P.; DE OLIVEIRA, G. C.; LEGEY, L. F. L. **Reduced Scenario Tree Generation for Mid-term Hydrothermal Operation Planning**. 2006 International Conference on Probabilistic Methods Applied to Power Systems. **Anais**... In: 2006 INTERNATIONAL CONFERENCE ON PROBABILISTIC METHODS APPLIED TO POWER SYSTEMS. jun. 2006

DE MATOS, V. L.; MORTON, D. P.; FINARDI, E. C. Assessing policy quality in a multistage stochastic program for long-term hydrothermal scheduling. **Annals of Operations Research**, v. 253, n. 2, p. 713–731, 1 jun. 2017.

DE MATOS, V. L.; PHILPOTT, A. B.; FINARDI, E. C. Improving the performance of Stochastic Dual Dynamic Programming. **Journal of Computational and Applied Mathematics**, v. 290, p. 196–208, 15 dez. 2015.

DE OLIVEIRA, W. et al. Optimal scenario tree reduction for stochastic streamflows in power generation planning problems. **Optimization Methods and Software**, v. 25, n. 6, p. 917–936, 1 dez. 2010.

DE OLIVEIRA, W. Regularized optimization methods for convex MINLP problems. **TOP**, v. 24, n. 3, p. 665–692, 1 out. 2016.

DE OLIVEIRA, W. Target radius methods for nonsmooth convex optimization. **Operations Research Letters**, v. 45, n. 6, p. 659–664, 1 nov. 2017.

DE OLIVEIRA, W.; SAGASTIZÁBAL, C. Level bundle methods for oracles with on-demand accuracy. **Optimization Methods and Software**, v. 29, n. 6, p. 1180–1209, 2 nov. 2014.

DE OLIVEIRA, W.; SAGASTIZÁBAL, C.; SCHEIMBERG, S. Inexact Bundle Methods for Two-Stage Stochastic Programming. **SIAM Journal on Optimization**, v. 21, n. 2, p. 517–544, 1 abr. 2011.

DELFINO, A. R. **Outer-approximation algorithms for nonsmooth convex MINLP problems with chance constraints**. [s.l.] Federal University at Paraná, 2018.

DEMPSTER, M. A. H.; THOMPSON, R. T. **Evpi-Based Importance Sampling Solution Procedures for Multistage Stochastic Linear Programmes on Parallel Mimd Architectures**. Rochester, NY: Social Science Research Network, 27 nov. 1997. Disponível em: <https://papers.ssrn.com/abstract=37767>. Acesso em: 24 nov. 2020.

DINIZ, A. L. et al. **Improvement in the hydro plants production function for the mid-term operation planning model in hydrothermal systems**. . In: ENGOPT 2008 - INTERNATIONAL CONFERENCE ON ENGINEERING OPTIMIZATION. Brazil: 2008

DINIZ, A. L. et al. **Short/Mid-Term Hydrothermal Dispatch and Spot Pricing for Large-Scale Systems-the Case of Brazil**. 2018 Power Systems Computation Conference (PSCC). **Anais**... In: 2018 POWER SYSTEMS COMPUTATION CONFERENCE (PSCC). jun. 2018

DINIZ, A. L. et al. A combined SDDP/Benders decomposition approach with a risk-averse surface concept for reservoir operation in long term power generation planning. **Annals of Operations Research**, v. 292, n. 2, p. 649–681, 1 set. 2020.

DINIZ, A. L.; MACEIRA, M. E. P. A Four-Dimensional Model of Hydro Generation for the Short-Term Hydrothermal Dispatch Problem Considering Head and Spillage Effects. **IEEE Transactions on Power Systems**, v. 23, n. 3, p. 1298–1308, ago. 2008.

DINIZ, A. L.; MACEIRA, M. E.; TERRY, L. A. **Modelling of the production function of hydro plants in the medium and short term models for interconnected systems planning**. . In: IX SYMP. SPECIALISTS IN ELECTRIC OPERATIONAL AND EXPANSION PLANNING—SEPOPE. Brazil: 2004

DINIZ, A. L.; SOUZA, T. M. Short-Term Hydrothermal Dispatch With River-Level and Routing Constraints. **IEEE Transactions on Power Systems**, v. 29, n. 5, p. 2427–2435, set. 2014.

DOS SANTOS, M. L. L. et al. Practical aspects in solving the medium-term operation planning problem of hydrothermal power systems by using the progressive hedging method. **International Journal of Electrical Power & Energy Systems**, Power Systems Computation Conference (PSCC) 2008. v. 31, n. 9, p. 546–552, 1 out. 2009.

DUPAČOVÁ. **Portfolio Optimization and Risk Management via Stochastic Programming**. S.l.: Osaka University Publishing Co., 2009.

ELZINGA, J.; MOORE, T. G. A central cutting plane algorithm for the convex programming problem. **Mathematical Programming**, v. 8, n. 1, p. 134–145, 1 dez. 1975.

FABIÁN, C. Bundle-type Methods for Inexact Data. **Central European Journal of Operations Research**, v. 8, p. 35–55, 2000.

FINARDI, E. C. et al. Stochastic hydro-thermal unit commitment via multi-level scenario trees and bundle regularization. **Optimization and Engineering**, v. 21, n. 2, p. 393–426, 1 jun. 2020.

FINARDI, E. C.; DECKER, B. U.; DE MATOS, V. L. An Introductory Tutorial on Stochastic Programming Using a Long-term Hydrothermal Scheduling Problem. **Journal of Control, Automation and Electrical Systems**, v. 24, n. 3, p. 361–376, 1 jun. 2013.

FLYNN, D. **Thermal Power Plant Simulation and Control**. [s.l.] IET, 2003.

FOSSO, O. B. et al. Generation scheduling in a deregulated system. The Norwegian case. **IEEE Transactions on Power Systems**, v. 14, n. 1, p. 75–81, fev. 1999.

FRANGIONI, A.; GENTILE, C.; LACALANDRA, F. Sequential Lagrangian-MILP approaches for Unit Commitment problems. **International Journal of Electrical Power & Energy Systems**, v. 33, n. 3, p. 585–593, 1 mar. 2011.

FREDO, G. L. M.; FINARDI, E. C.; DE MATOS, V. L. Assessing solution quality and computational performance in the long-term generation scheduling problem considering different hydro production function approaches. **Renewable Energy**, v. 131, p. 45–54, 1 fev. 2019.

GIL, E.; BUSTOS, J.; RUDNICK, H. Short-term hydrothermal generation scheduling model using a genetic algorithm. **IEEE Transactions on Power Systems**, v. 18, n. 4, p. 1256–1264, nov. 2003.

GOEL, V.; GROSSMANN, I. E. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. **Computers & Chemical Engineering**, v. 28, n. 8, p. 1409–1429, 15 jul. 2004.

GONÇALVES, R. E. C. et al. Comparing stochastic optimization methods to solve the medium-term operation planning problem. **Computational &amp; Applied Mathematics**, v. 30, n. 2, p. 289–313, 2011.

GONÇALVES, R. E. C.; FINARDI, E. C.; SILVA, E. L. DA. Applying different decomposition schemes using the progressive hedging algorithm to the operation planning problem of a hydrothermal system. **Electric Power Systems Research**, v. 83, n. 1, p. 19–27, 1 fev. 2012.

HELSETH, A.; FODSTAD, M.; MO, B. Optimal Medium-Term Hydropower Scheduling Considering Energy and Reserve Capacity Markets. **IEEE Transactions on Sustainable Energy**, v. 7, n. 3, p. 934–942, jul. 2016.

HJELMELAND, M. N. et al. Nonconvex Medium-Term Hydropower Scheduling by Stochastic Dual Dynamic Integer Programming. **IEEE Transactions on Sustainable Energy**, p. 1–1, 2018.

HOCHREITER, R.; PFLUG, G. C. Financial scenario generation for stochastic multi-stage decision processes as facility location problems. **Annals of Operations Research**, v. 152, n. 1, p. 257–272, 1 jul. 2007.

Hydro Dams for Large-Scale Electricity Supply | ClimateTechWiki. Disponível em: <http://www.climatetechwiki.org/technology/hydro_large>. Acesso em: 1 nov. 2017.

INFANGER, G. **Planning Under Uncertainty: Solving Large-scale Stochastic Linear Programs**. [s.l.] Boyd & Fraser, 1994.

JARDIM, D. L. D. D.; MACEIRA, M. E. P.; FALCAO, D. M. **Stochastic streamflow model for hydroelectric systems using clustering techniques**. 2001 IEEE Porto Power Tech Proceedings (Cat. No.01EX502). **Anais**... In: 2001 IEEE PORTO POWER TECH PROCEEDINGS (CAT. NO.01EX502). set. 2001

JOHANNESEN, A.; FLATABØ, N. Scheduling methods in operation planning of a hydro-dominated power production system. **International Journal of Electrical Power & Energy Systems**, v. 11, p. 189–199, 1 jul. 1989.

KALL, P.; WALLACE, S. W. **Stochastic programming**. [s.l.] Wiley, 1994.

KELLEY, JR., J. The Cutting-Plane Method for Solving Convex Programs. **Journal of the Society for Industrial and Applied Mathematics**, v. 8, n. 4, p. 703–712, 1 dez. 1960.

KIWIEL, K. C. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. **Mathematical Programming**, v. 69, n. 1–3, p. 89–109, 1 jul. 1995.

KOLOMVOS, G.; SAHARIDIS, G. K. D. Accelerating techniques on nested decomposition. **Optimization Methods and Software**, v. 32, n. 3, p. 455–469, 4 maio 2017.

KOVACEVIC, R. M.; PICHLER, A. Tree approximation for discrete time stochastic processes: a process distance approach. **Annals of Operations Research**, v. 235, n. 1, p. 395–421, 7 set. 2015.

LEMARÉCHAL, C.; NEMIROVSKII, A.; NESTEROV, Y. New variants of bundle methods. **Mathematical Programming**, v. 69, n. 1–3, p. 111–147, 1 jul. 1995.

LIMA, L. M. M.; POPOVA, E.; DAMIEN, P. Modeling and forecasting of Brazilian reservoir inflows via dynamic linear models. **International Journal of Forecasting**, v. 30, n. 3, p. 464–476, 1 jul. 2014.

LU, J. et al. Risk Analysis Method of Cascade Plants Operation in Medium Term Based on Multi-Scale Market and Settlement Rules. **IEEE Access**, v. 8, p. 90730–90740, 2020.

MACEIRA, M. et al. **Chain of Optimization Models for Setting the Energy Dispatch and Spot Price in the Brazilian System**. . In: PSCC'02. Sevilla, Spain: 2002

MACEIRA, M. E. P. et al. **Twenty Years of Application of Stochastic Dual Dynamic Programming in Official and Agent Studies in Brazil-Main Features and Improvements on the NEWAVE Model**. 2018 Power Systems Computation Conference (PSCC). **Anais**... In: 2018 POWER SYSTEMS COMPUTATION CONFERENCE (PSCC). jun. 2018

MORALES-ESPAÑA, G.; LATORRE, J. M.; RAMOS, A. **Tight and compact MILP formulation of start-up and shut-down ramping in unit commitment**. 2013 IEEE Power Energy Society General Meeting. **Anais**... In: 2013 IEEE POWER ENERGY SOCIETY GENERAL MEETING. jul. 2013a

MORALES-ESPAÑA, G.; LATORRE, J. M.; RAMOS, A. Tight and Compact MILP Formulation for the Thermal Unit Commitment Problem. **IEEE Transactions on Power Systems**, v. 28, n. 4, p. 4897–4908, nov. 2013b.

MUHLEN, G. V. **Impacto da modelagem da função de produção hidrelétrica no problema da programação diária da operação eletroenergética**. [s.l.] Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Elétrica, 2019.

ONS - Operador Nacional do Sistema Elétrico. Disponível em: <http://ons.org.br:80/paginas/resultados-da-operacao/historico-da-operacao>. Acesso em: 7 dez. 2020a.

ONS - Operador Nacional do Sistema Elétrico. Disponível em: <http://ons.org.br:80/paginas/sobre-o-sin/mapas>. Acesso em: 7 dez. 2020b.

OUOROU, A. A proximal cutting plane method using Chebychev center for nonsmooth convex optimization. **Mathematical Programming**, v. 119, n. 2, p. 239, 1 jul. 2009.

OURANI, K. I.; BASLIS, C. G.; BAKIRTZIS, A. G. **A Stochastic Dual Dynamic Programming model for medium-term hydrothermal scheduling in Greece**. 2012 47th International Universities Power Engineering Conference (UPEC). **Anais**... In: 2012 47TH INTERNATIONAL UNIVERSITIES POWER ENGINEERING CONFERENCE (UPEC). set. 2012

PEREIRA, M. V. F.; PINTO, L. M. V. G. A Decomposition Approach to the Economic Dispatch of Hydrothermal Systems. **IEEE Transactions on Power Apparatus and Systems**, v. PAS-101, n. 10, p. 3851–3860, out. 1982.

PEREIRA, M. V. F.; PINTO, L. M. V. G. Application of Decomposition Techniques to the Mid - and Short - Term Scheduling of Hydrothermal Systems. **IEEE Transactions on Power Apparatus and Systems**, v. PAS-102, n. 11, p. 3611–3618, nov. 1983.

PEREIRA, M. V. F.; PINTO, L. M. V. G. Stochastic Optimization of a Multireservoir Hydroelectric System: A Decomposition Approach. **Water Resources Research**, v. 21, n. 6, p. 779–792, 1 jun. 1985.

PEREIRA, M. V. F.; PINTO, L. M. V. G. Multi-stage stochastic optimization applied to energy planning. **Mathematical Programming**, v. 52, n. 1, p. 359–375, 1 maio 1991.

PÉREZ-DÍAZ, J. I. et al. Medium-term scheduling of a hydropower plant participating as a price-maker in the automatic frequency restoration reserve market. **Electric Power Systems Research**, v. 185, p. 106399, 1 ago. 2020.

PHILPOTT, A. B.; GUAN, Z. On the convergence of stochastic dual dynamic programming and related methods. **Operations Research Letters**, v. 36, n. 4, p. 450–455, 1 jul. 2008.

PHILPOTT, A.; PRITCHARD, G. **EMI-DOASA**. [s.l.] EMI, 2013.

QUINTERO, R. D. **Análise comparativa de diferentes simplificações para a função de produção hidrelétrica no contexto do planejamento da operação energética de curto prazo**. [s.l.] Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Elétrica, 2013.

ROCKAFELLAR, R. T.; WETS, R. J.-B. Scenarios and Policy Aggregation in Optimization Under Uncertainty. **Mathematics of Operations Research**, v. 16, n. 1, p. 119–147, 1 fev. 1991.

ROSEN, K. H. **Discrete Mathematics and its Applications**. [s.l.] MacGraw-Hill International, 1999.

ROTTING, T. A.; GJELSVIK, A. Stochastic dual dynamic programming for seasonal scheduling in the Norwegian power system. **IEEE Transactions on Power Systems**, v. 7, n. 1, p. 273–279, fev. 1992.

RUSZCZYŃSKI, A.; SHAPIRO, A. Handbooks in operations research and management science. In: [s.l.] Elsevier, 2003. v. 10.

SAGASTIZÁBAL, C. Divide to conquer: decomposition methods for energy optimization. **Mathematical Programming**, v. 134, n. 1, p. 187–222, 1 ago. 2012.

SANTOS, T. N. DOS; DINIZ, A. L. A New Multiperiod Stage Definition for the Multistage Benders Decomposition Approach Applied to Hydrothermal Scheduling. **IEEE Transactions on Power Systems**, v. 24, n. 3, p. 1383–1392, ago. 2009.

SANTOS, T. N. et al. Hourly pricing and day-ahead dispatch setting in Brazil: The dessem model. **Electric Power Systems Research**, v. 189, p. 106709, 1 dez. 2020.

SHAPIRO, A. Analysis of stochastic dual dynamic programming method. **European Journal of Operational Research**, v. 209, n. 1, p. 63–72, 16 fev. 2011.

SHAPIRO, A.; DENTCHEVA, D.; RUSZCZYNSKI, A. **Lectures on Stochastic Programming**. [s.l.] Society for Industrial and Applied Mathematics, 2009.

SHAPIRO, A.; NEMIROVSKI, A. On Complexity of Stochastic Programming Problems. In: **Continuous Optimization**. Applied Optimization. [s.l.] Springer, Boston, MA, 2005. p. 111–146.

SHAPIRO, A.; PHILPOTT, A. Tutorial on Stochastic Programming. 2007.

SHERKAT, V. R. et al. Stochastic Long-Term Hydrothermal Optimization for a Multireservoir System. **IEEE Transactions on Power Apparatus and Systems**, v. PAS-104, n. 8, p. 2040–2050, ago. 1985.

SHERKAT, V. R. et al. Modular and flexible software for medium and short-term hydrothermal scheduling. **IEEE Transactions on Power Systems**, v. 3, n. 3, p. 1390–1396, ago. 1988.

SOARES, M. P.; STREET, A.; VALLADÃO, D. M. On the solution variability reduction of Stochastic Dual Dynamic Programming applied to energy planning. **European Journal of Operational Research**, v. 258, n. 2, p. 743–760, 16 abr. 2017.

VAN ACKOOIJ, W. et al. Probabilistic optimization via approximate p-efficient points and bundle methods. **Computers & Operations Research**, v. 77, p. 177–193, 1 jan. 2017.

VAN ACKOOIJ, W.; DE OLIVEIRA, W. Level bundle methods for constrained convex optimization with various oracles. **Computational Optimization and Applications**, v. 57, n. 3, p. 555–597, 1 abr. 2014.

VAN ACKOOIJ, W.; DE OLIVEIRA, W.; SONG, Y. On level regularization with normal solutions in decomposition methods for multistage stochastic programming problems. **Computational Optimization and Applications**, v. 74, n. 1, p. 1–42, 1 set. 2019.

VAN SLYKE, R.; WETS, R. L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. **SIAM Journal on Applied Mathematics**, v. 17, n. 4, p. 638–663, 1 jul. 1969.

WOLF, C. et al. Applying oracles of on-demand accuracy in two-stage stochastic programming – A computational study. **European Journal of Operational Research**, v. 239, n. 2, p. 437–448, 1 dez. 2014.

WOOD, A. J.; WOLLENBERG, B. F. **Power Generation Operation and Control**. [s.l.] John Wiley & Sons, 1984.

XAVIER, L. et al. **Aprimoramento da modelagem da função de produção energética das usinas hidroelétricas no modelo DECOMP: Metodologia e resultados**. . In: SNPTEE. Brazil: 2005

ZAKERI, G.; PHILPOTT, A. B.; RYAN, D. M. Inexact Cuts in Benders Decomposition. **SIAM Journal on Optimization**, v. 10, n. 3, p. 643–657, 1 jan. 2000.

ZOU, J.; AHMED, S.; SUN, X. A. Stochastic dual dynamic integer programming. **Mathematical Programming**, v. 175, n. 1, p. 461–502, 1 maio 2019.

APPENDIX A

A.1 CHEBYSHEV APPLICATION IN AN MTHS PROBLEM

This section presents several results comparing the behavior of the NBD algorithm and the NBD-CC algorithm for solving an MTGS problem. The analysis is carried out by varying the tree scenario structure (number of stages and realizations per stage), the initial volume of the reservoirs, and coupling or non-coupling with the LTGS problem thought cuts in $n \in \mathcal{N}_T$. Only Case 1 is left without such coupling. The tree structure is represented by the number of children nodes of each node. For instance, a tree with 3 children nodes for each node in stages 1 and 2 (nine scenarios) is represented as 1×3×3: 9. The main results obtained by considering different update rules of each case's parameter are reported in Table 18. Note that $\overline{\sigma}_t$ is updated based on the gap^k dynamic since the latter is decreasing during the iterations, tending to zero. The solution method and update rule are indicated in the second and third column, following by the number of iterations to achieve the convergence in the column #*k* and its percentage difference in column #*k* diff. Furthermore, the percentage difference of the algorithm time, the average time per iteration, and the optimal value are presented, respectively, in the column time diff, average time diff, and $\overline{z}^k$ diff. The percentage difference in the assessment parameters is calculated as follows:

$$\%_{\text{diff}} = \frac{(\text{NBD-CC value} - \text{NBD value})}{\text{NBD value}} \times 100\%.$$

Therefore, a negative value indicates a reduction of the considered features (iterations, CPU time) yielded by the NBD-CC algorithm regarding the NBD algorithm and; otherwise, a positive value means an increase.

Table 18 –Comparison between NBD and NBD-CC behavior.

Subcase	Method	Update Rule	*k*	*k* diff [%]	*time* diff [%]	Average *time* diff [%]	diff [%]
Case 1: Structure Tree: 1×5×5×5×5×5:3125/Initial Volume [%] = 50/Without Coupling with the LTGS problem							
1.1	NBD	-	220	-	-	-	-
1.2	NBD-CC	$1000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	231	5.00	-0.76	-5.49	0.00
1.3	NBD-CC	$gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	207	-5.91	-8.43	-2.68	0.00
1.4	NBD-CC	$10 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	176	-20.0	-21.40	-1.75	0.00
1.5	NBD-CC	$100 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	184	-16.4	-21.02	-5.56	0.00
1.6	NBD-CC	$gap^k(\bar{z}^k)^{-1}(1-0.01t) \leq \bar{\sigma}_t$	209	-5.00	-6.58	-1.66	0.00
Case 2: Structure Tree: 1×5×5×5×5×5:3125/Initial Volume [%] = 50/With Coupling with the LTGS problem							
2.1	NBD	-	246	-	-	-	-
2.2	NBD-CC	$1000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	244	-0.81	-2.42	-1.62	0.00
2.3	NBD-CC	$100 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	244	-0.81	1.64	2.47	0.00
2.4	NBD-CC	$10 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	250	1.63	1.63	0.00	0.00
2.5	NBD-CC	$10gap^k(\bar{z}^k)^{-1}(1-0.01t) \leq \bar{\sigma}_t$	244	-0.81	-2.51	-1.71	0.00
2.6	NBD-CC	$100gap^k(\bar{z}^k)^{-1}(0.5t^{-1}) \leq \bar{\sigma}_t$	245	-0.41	1.56	1.98	0.00
2.7	NBD-CC	$10^4 gap^k(\bar{z}^k)^{-1}(0.5t^{-1}) \leq \bar{\sigma}_t$	244	-0.81	-1.14	-0.33	0.00
2.8	NBD-CC	$5000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	244	-0.81	-4.11	-3.32	0.00
Case 3: Structure Tree: 1×10×10×10:1000/Initial Volume [%] = 50/With Coupling with the LTGS problem							
3.1	NBD	-	67	-	-	-	-
3.2	NBD-CC	$10 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	71	5.97	1.33	-4.38	0.00
3.3	NBD-CC	$100 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	69	2.99	1.28	-1.66	0.00
3.4	NBD-CC	$gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	67	0.00	-3.91	-3.91	0.00
3.5	NBD-CC	$1000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	65	-2.99	-4.11	-1.16	0.00
3.6	NBD-CC	$2000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	74	10.45	10.63	0.16	0.00
3.7	NBD-CC	$5000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	70	4.48	0.98	-3.34	0.00
3.8	NBD-CC	$300 \cdot gap^k(\bar{z}^k)^{-1}t \leq \bar{\sigma}_t$	66	-1.49	-7.80	-6.41	0.00
3.9	NBD-CC	$1000 \cdot gap^k(\bar{z}^k)^{-1}t \leq \bar{\sigma}_t$	68	1.49	2.29	0.79	0.00
3.10	NBD-CC	$500 \cdot gap^k(\bar{z}^k)^{-1}t \leq \bar{\sigma}_t$	67	0.00	-5.11	-5.11	0.00
3.11	NBD-CC	$1000 \cdot gap^k(\bar{z}^k)^{-1}t^{-1} \leq \bar{\sigma}_t$	67	0.00	-3.04	-3.04	0.00
Case 4: Structure Tree: 1×20×20×10:4000/Initial Volume [%] = 50/With Coupling with the LTGS problem							
4.1	NBD	-	47	-	-	-	-
4.2	NBD-CC	$100 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	50	6.38	13.28	6.48	0.00
4.3	NBD-CC	$500 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	50	6.38	-3.89	-9.65	0.00
4.4	NBD-CC	$1000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	51	8.51	16.48	7.34	0.00

4.5	NBD-CC	$5000 \cdot gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	49	4.26	4.46	0.20	0.00
Case 5: Structure Tree: 1×20×20×10:4000/Initial Volume [%] = 80/With Coupling with the LTGS problem							
5.1	NBD	-	51	-	-	-	-
5.2	NBD-CC	$gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	61	19.61	21.16	1.30	0.00
5.3	NBD-CC	$0.1gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	65	27.45	34.08	5.20	0.00
5.4	NBD-CC	$0.5gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	54	5.88	9.03	2.97	0.00
5.5	NBD-CC	$10gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	62	21.57	28.56	5.75	0.00
5.6	NBD-CC	$100gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	63	23.53	32.92	7.60	0.00
5.7	NBD-CC	$200gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	63	23.53	18.26	-4.27	0.00
5.8	NBD-CC	$500gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	65	27.45	19.63	-6.14	0.00
5.9	NBD-CC	$5000gap^k(\bar{z}^k)^{-1} \leq \bar{\sigma}_t$	64	25.49	26.43	0.75	0.00

Source: Author.

The results above suggest three general conclusions: (*i*) the optimal value of the problem $\bar{z}^k$ obtained with the NBD and NDB-CC algorithm is equivalent, considering that the percentage difference $\bar{z}^k$ diff equals zero in all cases. (*ii*) The computational effort for solving the LP master subproblem of the NBD and NDB-CC algorithm is comparable. Note that, for the reported cases, the average time diff is in the value range ±10%, even with an appropriate update rule, the NBD-CC LP can be computationally cheaper. Finally, (*iii*) the numerical performance of the approach with Chebyshev centers is superior only when there is no coupling between the MTGS problem with the LTGS problem, i.e., without cuts representing a 5 years future cost in the leaf nodes $n \in \mathcal{N}_T$. Remark that case 1 (without coupling with LTGS problem) obtains considerable reductions of the iteration numbers with the NBD-CC algorithm. For instance, in subcases 1.4 and 1.5, the convergence is achieved with 44 and 36 fewer iterations than the NBD algorithm, reducing the computational effort around 21%. However, subcase 1.2 shows an unsuitable update rule, providing a poor performance (increase in 5% the number of iterations). A dependence of the stage t in the update rule is assessed in subcase 1.6, presenting a minor effect compared with the iteration reduction of analogous subcase 1.3. On the other hand, case 2 is analogous to case 1 but coupled with the LTGS problem, and the results indicate no significant improvement in the convergence performance. Note that for all the considering update rules, including the better ones of case 1, the number of iterations is practically identical to the NBD method. This shortcoming is investigated in cases 3, 4, and 5 for different tree structures and few initial volumes of the system, pointing out that the NBD-CC algorithm has an equal or even worse performance than the NBD algorithm when the MTGS is coupled with the LTGS problem.

To understand this behavior, it is necessary to bear in mind the didactic example presented in Section 4.2.3. When it is incorporated the final water value in the leaf node of the MTGS problem, the stored volume has associated a future cost value greater than zero for all regions of the feasible set solution. Consequently, the effect of $\overline{\sigma}_t$ the variable in this condition is only an offset of the objective function, maintaining the NBD solution, see Figure 22 (a). In conclusion, it turned out that the NBD-CC algorithm is an unpromising method for solving the MTGS problem due to its intrinsic characteristic of coupling with the LTGS problem.

Nevertheless, the Chebyshev master subproblem (4.13) presents benefits for subproblems where the future cost value is equal to zero in the last stage (as the LTGS problem). The results using the Chebyshev approach for solving the LTGS problem within SDDP are presented in Appendix A.2.

A.2 CHEBYSHEV APPLICATION IN AN LTHS PROBLEM

This section presents the results of a Chebyshev application in an LTHS problem published in work (BELTRÁN et al., 2020). This analysis is based on the comparison between the quality of the obtained policies via (*i*) the classical SDDP (CL) and (*ii*) the SDDP with Chebyshev centers (CC). The policy quality assessment follows the guidelines described in Section A.2.1. For CL and CC cases, the same 5 scenario trees are computed, and an out-of-sample simulation of the resulting policies with 2,500 scenarios is carried out. A neutral risk-averse measure is considered. This study is based on real-life instances of the Brazilian case, with individualized decisions per plant over a five-year planning horizon ($T = 60$) with monthly decisions. The problem includes 153 hydro plants (67 reservoirs), 141 thermal plants, 3 load levels, 4 subsystems, 4 deficit levels, and 21 river basins. We make the data available in the link[12]. Accordingly, the LP (4.13), for every stage t and node of the scenario tree, comprises 2,886 variables and 1,459 constraints. The inflow uncertainties are handled via a PAR model (LIMA; POPOVA; DAMIEN, 2014) with order 1 generating a scenario tree containing 50 realizations of $rp_{t,r,b}$ per stage. The forward step considers 120 scenarios per iteration with resampling. The sampled scenarios at each iteration for CL and CC are

[12] https://github.com/OPTE-2020-1017/Improving-the-performance-of-the-SDDP-algorithm-using-Chebyshev-centers.

identical. All cases were stopped after 12 hours of CPU time[13], attaining on average a $\underline{z}$ stabilization of approximately 0.05%. This work's cut selection strategy derives from (DE MATOS; PHILPOTT; FINARDI, 2015).

Regarding the CC case, the proposed heuristic for updating the parameter $\overline{\sigma}_t$ in (A.1) is:

$$\overline{\sigma}_t = k_{\text{CC}} \cdot \frac{\underline{z}^k - \underline{z}^{k-1}}{\sum\limits_{\text{if } nk<10,\, k=0 \text{ else } k=nk-10}^{nk} (\underline{z}^k - \underline{z}^{k-1})}. \tag{A.1}$$

The constant $k_{\text{CC}} \geq 0$ is adjusted for better scalability; when $k_{\text{CC}} = 0$, CL and CC are identical. In this work, the values $k_{\text{CC}} = \{1{,}000;\ 5{,}000;\ 10{,}000;\ 100{,}000\}$ are assessed. This constant is weighed by a proportion of the lower bound improvement w.r.t. the last ten iterations. The main idea is to make $\overline{\sigma}_t$ a function of the lower bound progress. Such parameter increases when the lower bound presents a high increase rate since, at this point, the convergence is not yet achieved, and new regions of the cost-to-go-function can be explored. Otherwise, a stabilization of $\underline{z}$ indicates that new trial points do not improve the model and $\overline{\sigma}_t$ must decrease to satisfy the algorithm convergence requirements. Note that $\overline{\sigma}_t \to 0$ as the algorithm gets to convergence. Several simple updating rules can be formulated by incorporating the problem dynamics and preserving the convergence.

Computationally, a parallel processing strategy is used within servers with 3.33GHz and 32 GB RAM. All LPs were solved using Gurobi, called from environment C++.

A.2.1 Assessing policy quality

In this work, the policy quality is evaluated based on the estimation of the optimality gap (RUSZCZYŃSKI; SHAPIRO, 2003) (BAYRAKSAN; MORTON, 2006) (FINARDI; DECKER; DE MATOS, 2013). The gap is defined as the difference between the objective

[13] According to (SHAPIRO, 2011), an appropriate stopping criterion is the stabilization method, being less optimistic than (PEREIRA; PINTO, 1991). However, the stabilization does not imply that the problem is solved to optimality, it only means that the current policy (cost-to-go-functions) is not improving meaningfully with new trial points.

function values of the continuous problem in an implementable policy $\bar{x}$ and the solution x^*, as given in (A.2):

$$gap = f(\bar{x}) - f(x^*). \tag{A.2}$$

Nevertheless, rarely in large-scale problems, the values of $f(\bar{x})$ and $f(x^*)$ can be computed exactly. In practice, only confidence intervals for the optimality gap are computed. According to (RUSZCZYŃSKI; SHAPIRO, 2003) and (SHAPIRO; PHILPOTT, 2007), a lower estimator of $f(x^*)$ is $\mathbb{E}[\underline{z}^k]$, in which $\mathbb{E}[\underline{z}^k]$ is calculated by solving M problems with independent samples. We denote $\mathbb{E}[\underline{z}^k]$ by *low* and the standard deviation of M samples of $\underline{z}^k$ by σ_{low}. In general, $M \in \{5,10\}$ is sufficient (SHAPIRO; PHILPOTT, 2007) to achieve a reasonable estimative. Although there are other less expensive approaches to determine the lower estimator (BAYRAKSAN; MORTON, 2006), this work considers the most reliable estimation based on several SDDP runs. On the other hand, an upper estimator can be estimated via an out-of-sample simulation, using N scenarios ($N >> M$), of the M cost-to-go functions available after an SDDP run; for more details, see (DE MATOS; MORTON; FINARDI, 2017). Thereupon, the average and standard deviation of M samples of $\bar{z}$ are obtained, denominated as *up* and σ_{up}, respectively.

In this context, a confidence interval (CI) of the optimality gap is given in (A.3), as follows:

$$\text{CI} = (up + t_{M-1,p}\sigma_{up}) - (low - t_{M-1,p}\sigma_{low}). \tag{A.3}$$

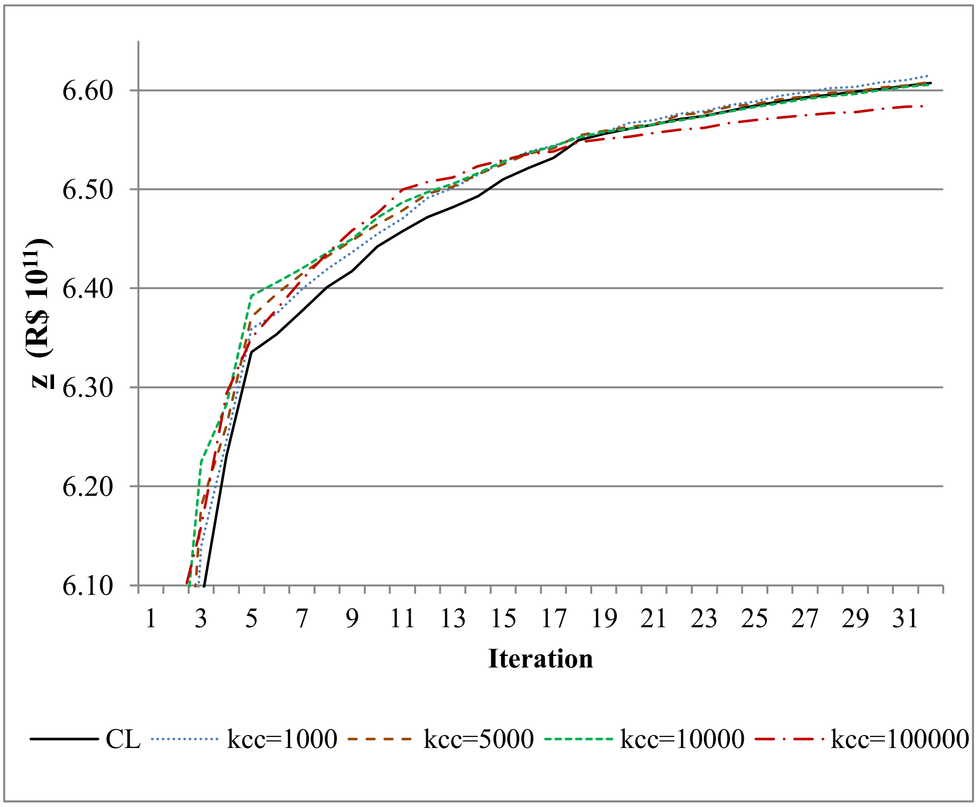
In the interval above, $t_{M\text{-}1,\alpha}$ is the critical value of the *t*-student distribution with M-1 degrees of freedom and a confidence level named p. In this way, the optimality gap is lower than or equal to CI with a probability of $(1-p)$ %.

A.2.2 Iterative process

To compare the iterative process of CL and CC cases, an SDDP run is presented in Figure 39 and Figure 40, evidencing the lower bound progress by iteration and by computing time, respectively. Note that the most considerable $\underline{z}$ performance differences between CL and

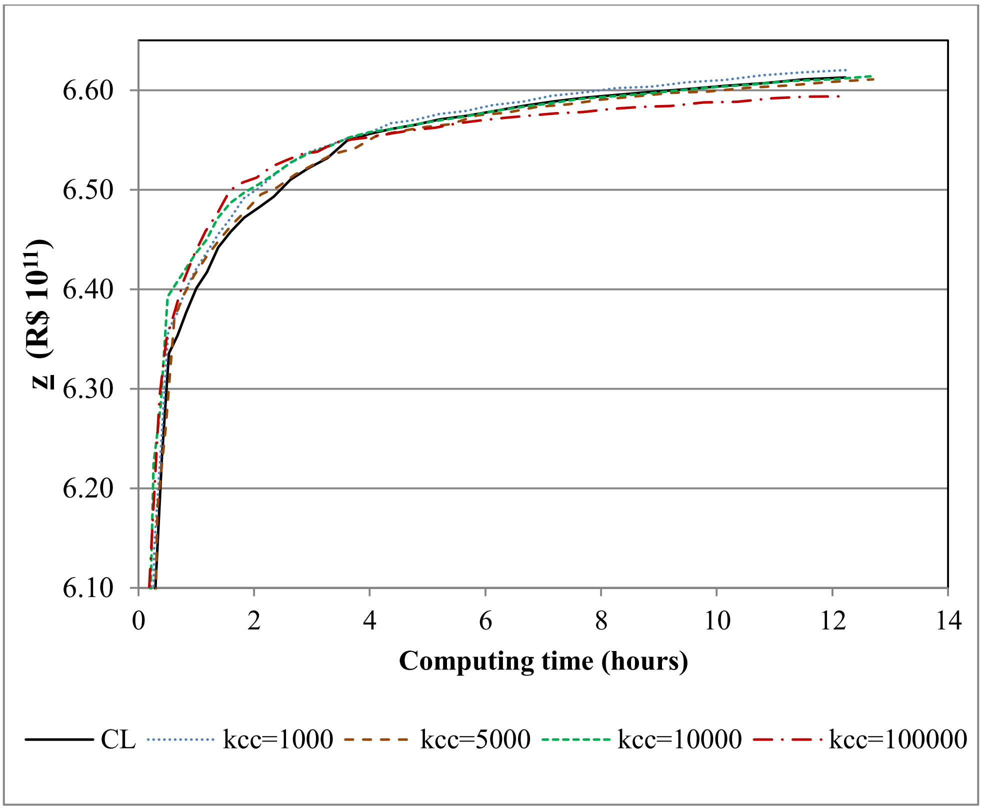
CC cases occur in the first part of the iterative process. In general, CC cases achieve a faster $\underline{z}$ increase. This behavior evinces that the CC cases construct the cost-to-go functions by exploring regions of the state variable different from the regions explored by the CL case (in the application under consideration, CC keeps more water in the reservoirs than the CL does in the first iterations). This work's primary purpose is to determine if the CC approach can improve the quality of the resulting cost-to-go functions, evaluated in Section A.2.3.

Figure 39 – Lower bound progress per iteration.



Source: (BELTRÁN et al., 2020).

Figure 40 – Lower bound progress per computing time.

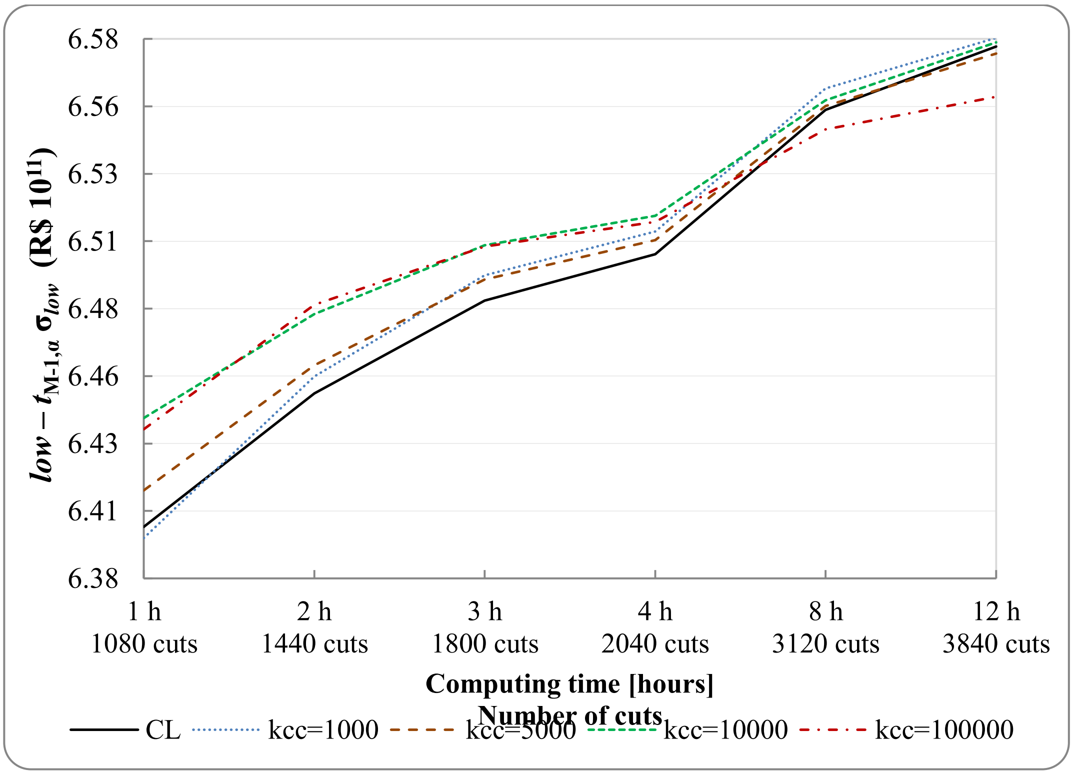


Source: (BELTRÁN et al., 2020).

A.2.3 Optimality gap

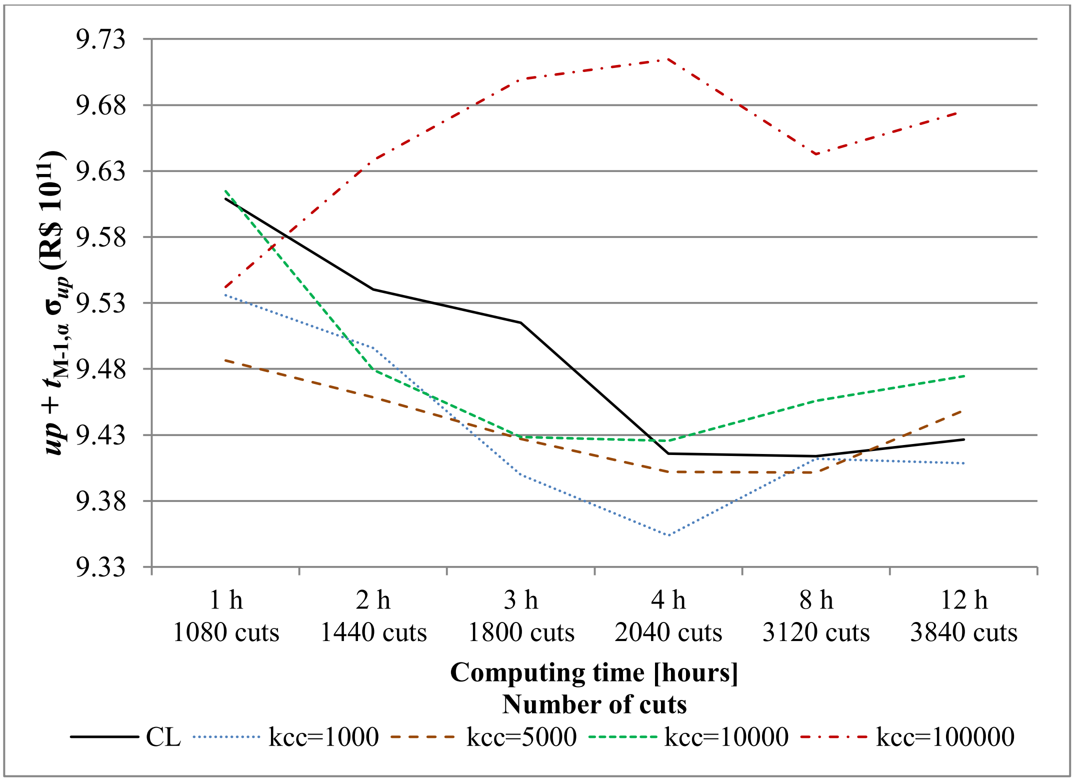
Optimality gaps related to the policies yielded by the CL and CC cases are computed after 1, 2, 3, 4, 8, and 12 hours of the optimization process. This process corresponds to 960, 1440, 1800, 2040, 3120, and 3840 cuts of the cost-to-go functions. Figure 41 presents the lower estimators of the optimality gap, whereas, Figure 42 shows the upper estimators, both described as a function of the computing time.

Figure 41 – Lower estimator of the optimality gap.



Source: (BELTRÁN et al., 2020).

Figure 42 – Upper estimator of the optimality gap.

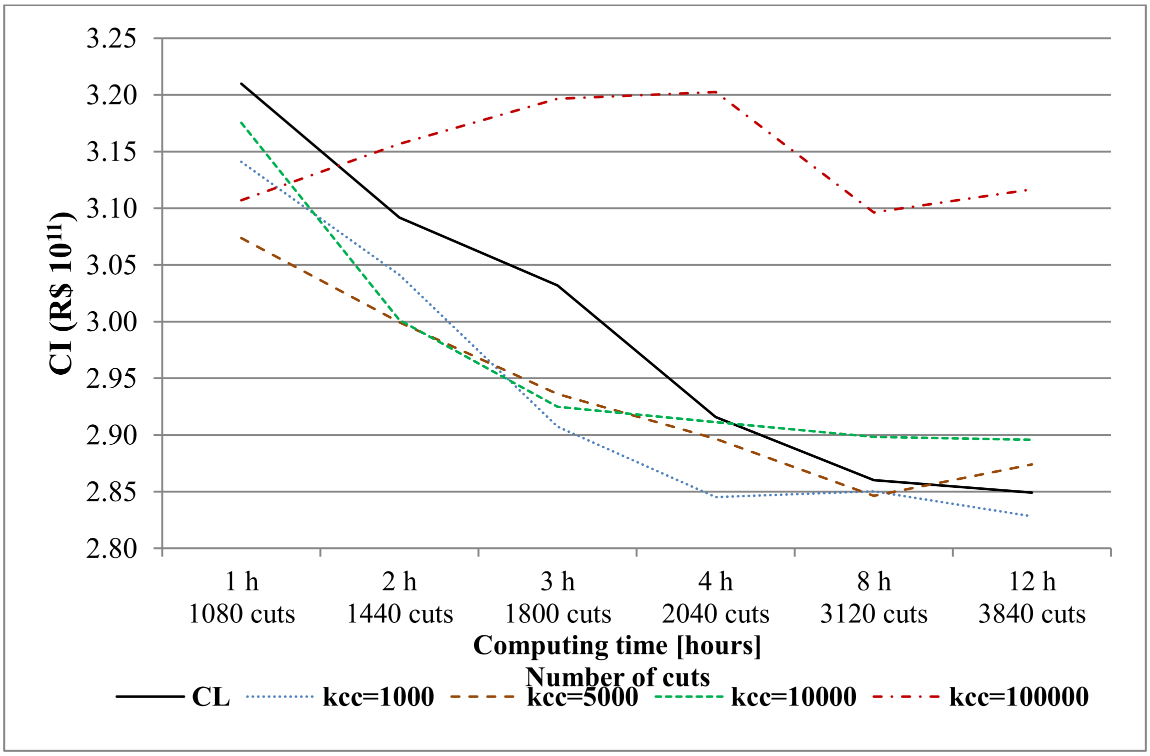


Source: (BELTRÁN et al., 2020).

Note that, in comparison with the CL case, the CC approach produces better lower estimators in the first processing hours (< 4 hours) if k_{CC} increases. On the other hand, the upper estimator is more sensitive to the rule employed for updating $\overline{\sigma}_t$. When $\overline{\sigma}_t$ is larger than expected from a certain iteration onwards, the cost-to-go functions start to be exclusively constructed in a specific region since considerably close iterates are obtained throughout the optimization process. In this context, other cost-to-go function regions that potentially improve the quality policy are no longer explored. For instance, k_{CC} = 100,000 presents the worst estimates for the upper bound, obtaining 3.0% higher upper values related to the CL case. Similarly, k_{CC} = 10,000 worsens the upper limit behavior from the fourth hour, indicating that the $\overline{\sigma}_t$ value must be lower at this iteration. This interpretation is confirmed with k_{CC} = 5,000, which presents high-quality behavior for up to 8 hours. Nevertheless, lower $\overline{\sigma}_t$ values obtained between hour 8 and hour 12 are necessary to generate useful iterates. On the other hand, an appropriate k_{CC} regularizes the iterative process because iterates come closer to the solution set. For instance, case k_{CC} = 1,000 achieves considerably successful results, i.e., larger lower estimators and smaller upper estimators are obtained during the whole optimization process.

Figure 43 presents the size of the optimality gap, in which the values below the CL reference curve represent an improvement of the policy quality. The percentage difference between the CI obtained from the CC cases regarding the CL case is reported in Table 19. Negative values indicate higher quality policies.

Figure 43 – Confidence interval of the optimality gap.



Source: (BELTRÁN et al., 2020).

Table 19 – CI difference regarding CL case (%).

k_{CC}	Optimization computing time [hours]					
	1	2	3	4	8	12
1,000	-2.74	-1.63	-4.11	-2.42	-0.35	-0.74
5,000	-4.23	-2.99	-3.16	-0.66	-0.49	0.86
10,000	-1.07	-2.92	-3.53	-0.16	1.34	1.63
100,000	-3.20	2.11	5.43	9.83	8.25	9.38

Source: (BELTRÁN et al., 2020).

It is highlighted that the Chebyshev iterates associated with case k_{CC} = 1,000 construct operational policies with higher quality during the whole SDDP run. Remark that, the Chebyshev effect is most significant in the first part of the optimization process (approximately ≤ 3 hours), attaining on average CI percentage reductions of 2.63, 3.46, and 2.51 for cases with k_{CC} equal 1,000, 5,000, and 10,000 respectively. In this sense, the Chebyshev approach appears to be a useful tool to compute a more reliable policy (w.r.t. CL) when processing time is tight. Furthermore, for long SDDP runs (≥12 hours), improved results can also be obtained from conservative k_{CC} values (for instance, in our study, k_{CC} =

1,000). Another alternative is to leverage the Chebyshev effect in the first part of the optimization process and finalizing with the classical SDDP approach. For that, set $k_{\mathrm{CC}} = 0$ from a given iteration. In order to illustrate this fact, Table 20 presents the results of a new case with $k_{\mathrm{CC}} = \{5{,}000;\ 10{,}000\}$ for the first 3 hours and, otherwise, $k_{\mathrm{CC}} = 0$. The results indicate that after 12 hours, the CC cases improve the quality of the resulting policy.

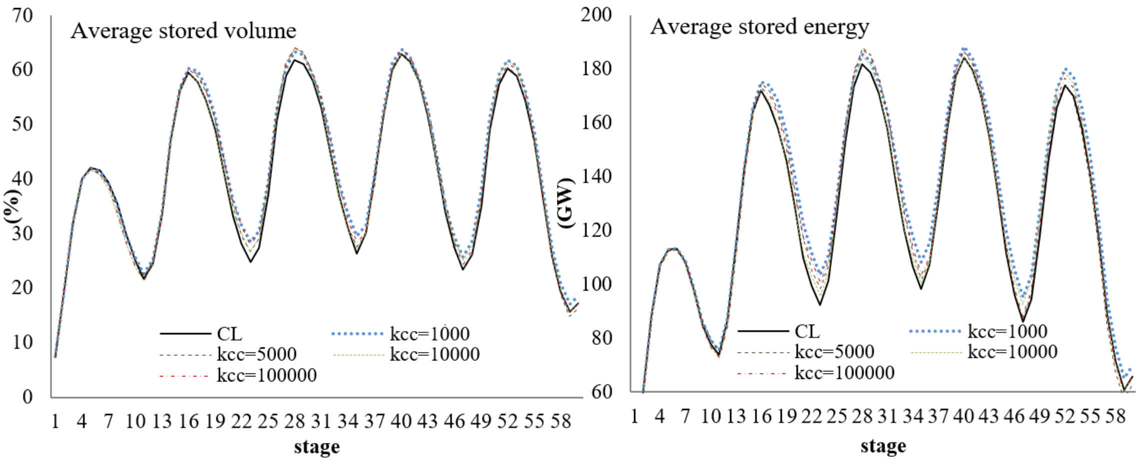
Table 20 – CI difference regarding CL case (%).

k_{CC}		Optimization computing time [hours]					
time ≤ 3 hours	time > 3 hours	1	2	3	4	8	12
5,000	0	-4.23	-2.99	-2.90	-0.89	-1.68	-1.89
10,000	0	-1.07	-2.92	-3.52	0.01	1.05	-1.23

Source: (BELTRÁN et al., 2020).

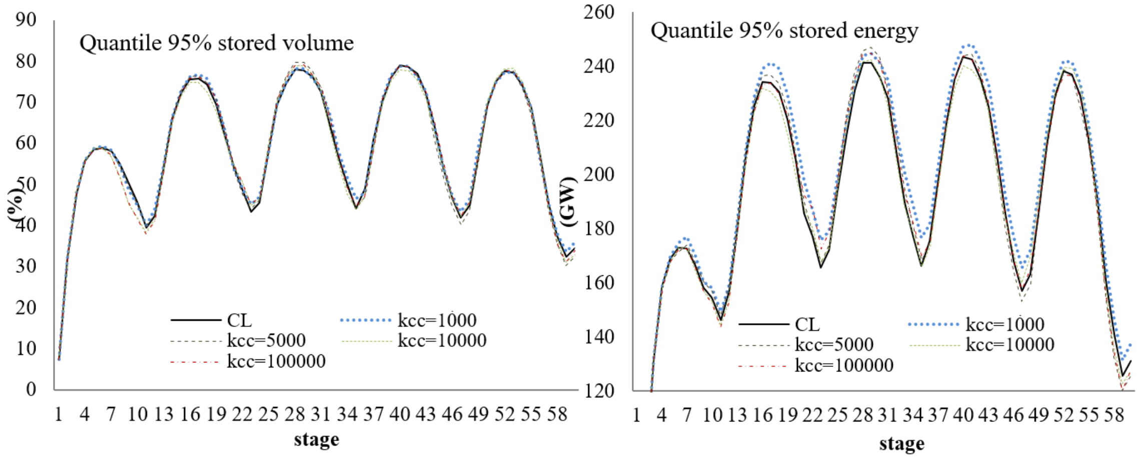
Improving the quality policy (especially in the first iterations) upgrades the future water cost valuation. In our case, an out-of-sample simulation reveals that the system operated by CC policies presents, for the planning horizon, a higher stored volume and stored energy tendency than the system operated by CL policies. Figure 44, Figure 45, and Figure 46 compare the average, the quantile 95%, and the quantile 5% of the stored volume and the stored energy. The stored volume values are presented in the proportion of the total system storage capacity. It is highlighted that, for droughts, the system can achieve higher storage and energy capacity when operated with the CC policies (see Figure 46). The quantile 95% results reveal a similar stored volume trajectory plus a better expectation of stored energy via CC policies. These results indicate that the water is allocated in different reservoirs of the system, in which it is more productive.

Figure 44 – Average stored volume and energy.



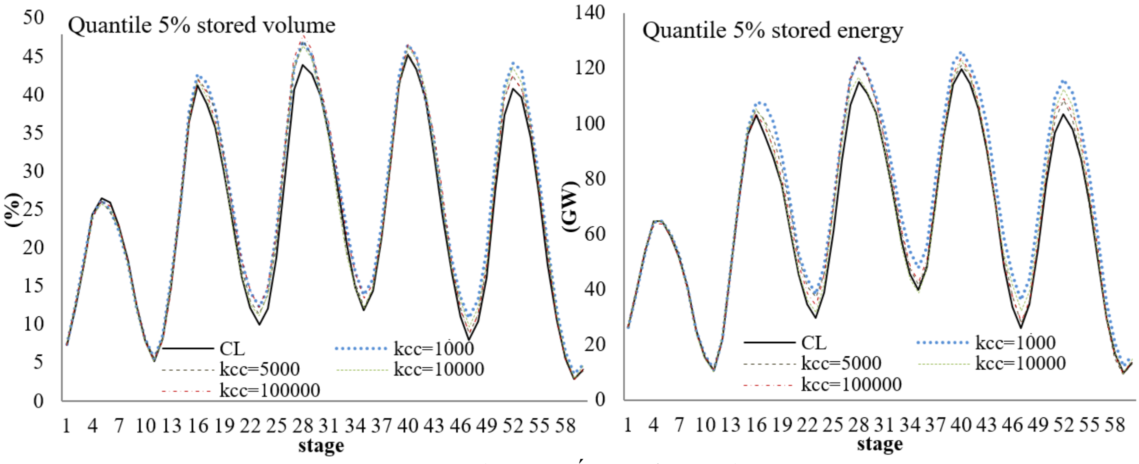
Source: (BELTRÁN et al., 2020).

Figure 45 – Quantile 95% stored volume and energy.



Source: (BELTRÁN et al., 2020).

Figure 46 – Quantile 5% stored volume and energy



Source: (BELTRÁN et al., 2020).

APPENDIX B

B.1 Stochastic Dual Dynamic Programming – SDDP

This section presents the SDDP algorithm of (PEREIRA; PINTO, 1991). In general terms, the SDDP is a cutting-plane based-method that carries out a scenario sampling for generating trial states in the forward step and Benders cuts in the backward step, for more details see (SHAPIRO, 2011) (PHILPOTT; GUAN, 2008) and (BELTRÁN et al., 2020). The main SDDP advantage is to maintain the number of subproblems solved per iteration proportional to the amount of sampled scenarios (SHAPIRO, 2011), instead of computing the total number of tree nodes as in Algorithm 1–NBD. Nevertheless, the SDDP method can only be applied for scenario trees with common samples since the sharing cuts between nodes of the same period must be guaranteed. Usually, the SDDP stopping criterion is based on a lower bound stabilization (SHAPIRO, 2011) or a limited computing time.

Algorithm 4 – SDDP

Step 0: Define $k = 1$ and $\mathcal{J}$, for $t = 1, \ldots, T$;

Step 1: Forward Step. Randomly select a sample W^k (resampling at each iteration) with $1 \leq |W^k| \leq NS$ scenarios, being NS the total number of scenarios. For $t = 1, \ldots, T$ and $\xi \in W^k$, get $(c_t, B_t, A_t, b_t) = \xi_t^s$ and solve (4.5) to obtain $x_t^k = x_t^k(\xi_{[t]})$. For $t=1$, obtain the lower bound according to (4.7);

Step 2: Stopping criterion. If the lower bound stabilizes during several iterations or the limit computing time is exceeded, stop. A lower bound estimator of the problem is $\underline{z}^k$, and the obtained cost-to-functions are the set of cuts $(\alpha_t^j, \beta_t^j)_{j \in \mathcal{J}}$ for $t = 1, \ldots; T$. Otherwise, go to Step 3;

Step 3: Backward Step. Solve (4.2) for all $\xi_T \in \Xi_t$ to calculate α_T^k and β_T^k. Set $\mathcal{J}$ $\mathcal{J}$. For $t = T-1, \ldots, 2$ and all $\xi_t^w \in \Xi_t$, solve (4.5) to calculate α_t^k and β_t^k. Set $\mathcal{J}$ $\mathcal{J}$.

Step 4: Loop: Set $k = k+1$ and return to Step 1.

In this work, the SDDP results are used as a benchmark for the two-stage solution methods explained in Section 4.3, when the scenario tree has common samples.