

Universidade Federal de Juiz de Fora
Faculdade de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Lílian Chaves Brandão dos Santos

**Programação Dinâmica Dual Inteira Estocástica: análise conceitual, avanço metodológico
e aplicação ao problema de planejamento energético.**

Juiz de Fora

2026

Lílian Chaves Brandão dos Santos

Programação Dinâmica Dual Inteira Estocástica: análise conceitual, avanço metodológico e aplicação ao problema de planejamento energético.

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas de Energia Elétrica, como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Bruno Henriques Dias, D.Sc.

Juiz de Fora

2026

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF com os dados fornecidos pelo(a) autor(a)

Chaves Brandão dos Santos, LÍlian.

Programação Dinâmica Dual Inteira Estocástica: análise conceitual, avanço metodológico e aplicação ao problema de planejamento energético. / LÍlian Chaves Brandão dos Santos. – 2026.

166 f. : il.

Orientador: Bruno Henriques Dias, D.Sc.

Doutorado – Universidade Federal de Juiz de Fora, Faculdade de Engenharia.
Programa de Pós-Graduação em Engenharia Elétrica, 2026.

Lílian Chaves Brandão dos Santos

Programação Dinâmica Dual Inteira Estocástica: análise conceitual, avanço metodológico e aplicação ao problema de planejamento energético

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutora em Engenharia Elétrica. Área de concentração: Sistemas de Energia Elétrica

Aprovada em 27 de agosto de 2026.

BANCA EXAMINADORA

Prof. Dr. Bruno Henriques Dias - Orientador

Universidade Federal de Juiz de Fora

Profa. Dra. Claudia Alejandra Sagastizábal

Universidade Estadual de Campinas

Prof. Dr. Erlon Cristian Finardi

Universidade Federal de Santa Catarina

Prof. Dr. André Luis Marques Marcato

Universidade Federal de Juiz de Fora

Prof. Dr. Leonardo Willer de Oliveira

Universidade Federal de Juiz de Fora

Profa. Dra. Danielle de Freitas da Silva

Centro de Pesquisas de Energia Elétrica

Juiz de Fora, 29/06/2026.



Documento assinado eletronicamente por **BRUNO HENRIQUES DIAS, Professor(a)**, em 27/08/2026, às 11:49, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Andre Luis Marques Marcato, Professor(a)**, em 27/08/2026, às 11:50, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Erlon Cristian Finardi, Usuário Externo**, em 31/08/2026, às 09:27, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado digitalmente
CLAUDIA ALEJANDRA SAGASTIZABAL
Data: 01/09/2026 08:38:56-0300
Verifique em <https://validar.iti.gov.br>



Documento assinado digitalmente
DANIELLE DE FREITAS DA SILVA
Data: 08/09/2026 08:43:13-0300
Verifique em <https://validar.iti.gov.br>



Documento assinado digitalmente
LEONARDO WILLER DE OLIVEIRA
Data: 08/09/2026 09:30:04-0300
Verifique em <https://validar.iti.gov.br>

RESUMO

O planejamento da operação hidrotérmica de médio prazo requer a tomada de decisões sob incerteza, considerando restrições operativas, elétricas, ambientais e regulatórias. Em aplicações reais, muitas dessas restrições possuem natureza condicional, exigindo a introdução de variáveis inteiras nos modelos de otimização. Entretanto, a presença dessas variáveis aumenta significativamente a complexidade dos problemas estocásticos multiestágio e impõe desafios à aplicação dos métodos clássicos de decomposição. Esta tese investiga os impactos da representação de restrições condicionais em problemas de planejamento hidrotérmico e propõe novas estratégias para sua resolução eficiente. Inicialmente, é realizada uma análise teórica da estrutura dual dos subproblemas utilizados no método de Programação Dinâmica Dual Inteira Estocástica, identificando propriedades relevantes para a construção de aproximações da função de recurso. Com base nesses resultados, propõe-se uma nova família de cortes de Lagrange, denominada *Tighter Lagrangian Cuts*, bem como uma estratégia híbrida de decomposição que combina características dos cortes de Benders e dos cortes Lagrangianos. As metodologias propostas são avaliadas em estudos de caso representativos do planejamento da operação do sistema elétrico brasileiro. Os resultados demonstram que os *Tighter Lagrangian Cuts* produzem aproximações mais precisas da função de recurso e aceleram a convergência do processo iterativo. Adicionalmente, a estratégia híbrida mostrou-se capaz de combinar escalabilidade computacional e maior capacidade de representação das não convexidades associadas às variáveis inteiras. Os resultados obtidos evidenciam o potencial das técnicas desenvolvidas para ampliar a aplicabilidade dos métodos de decomposição em problemas estocásticos multiestágio com restrições condicionais, contribuindo para o avanço do estado da arte na área.

Palavras-chave: Planejamento Energético, Otimização Estocástica, Programação Linear Inteira Mista, Programação Dinâmica Dual Inteira Estocástica, Cortes de Lagrange

ABSTRACT

Medium-term hydrothermal operation planning requires decision-making under uncertainty while accounting for operational, electrical, environmental, and regulatory constraints. In real-world applications, many of these constraints are conditional in nature, requiring the introduction of integer variables into optimization models. However, the presence of integer variables significantly increases the complexity of multistage stochastic optimization problems and poses major challenges to the application of classical decomposition methods. This thesis investigates the impacts of representing conditional constraints in hydrothermal planning problems and proposes new strategies for their efficient solution. First, a theoretical analysis of the dual structure of the subproblems arising in the Stochastic Dual Dynamic Integer Programming (SDDiP) framework is conducted, identifying properties that are relevant to the construction of cost-to-go function approximations. Based on these findings, a new family of Lagrangian cuts, termed Tighter Lagrangian Cuts, is proposed, together with a hybrid decomposition strategy that combines features of both Benders cuts and Lagrangian cuts. The proposed methodologies are evaluated through case studies representative of the Brazilian power system operation planning problem. The results show that Tighter Lagrangian Cuts provide more accurate approximations of the cost-to-go function and accelerate the convergence of the iterative solution process. Furthermore, the hybrid strategy proved capable of combining computational scalability with an enhanced ability to represent the nonconvexities associated with integer variables. Overall, the proposed techniques expand the applicability of decomposition methods to multistage stochastic problems with conditional constraints and contribute to advancing the state of the art in the field.

Keywords: Energy Planning, Stochastic Optimization, Mixed-Integer Linear Programming, Stochastic Dual Dynamic Integer Programming, Lagrangian Cuts

SUMÁRIO

1 – Considerações Iniciais	10
1.1 Introdução e Motivação	10
1.1.1 Desafios do Planejamento Energético	10
1.1.2 Desafios da Otimização Linear Inteira Mista	12
1.1.3 Desafios da Otimização Estocástica	12
1.2 Objetivos e contribuições	12
1.3 Publicações e apresentações do trabalho	13
1.4 Organização do texto	14
2 – Problema Multiestágio Estocástico Linear Inteiro Misto (MS-MILP) e Planeja- mento Hidrotérmico	15
2.1 Visão geral do planejamento energético	15
2.2 Componentes e modelagem do problema de planejamento energético	16
2.3 Caracterização de problemas de otimização estocástica multiestágios	17
2.4 Desafios da programação inteira no planejamento energético: restrições condi- cionais	18
2.5 Modelagem de restrições condicionadas	19
3 – Métodos para Resolução de MS-MILPs	21
3.1 Decomposição de problemas estocásticos multiestágios	21
3.2 Decomposição temporal em MS-MILP	22
3.3 Decomposição temporal em MS-MILP com variáveis de recurso binárias	25
3.4 Resumo dos cortes propostos para MS-MILPs	27
3.5 Aplicações da DDiP e SDDiP	28
3.6 Outras formas de decomposição para MS-MILPs	31
3.6.1 Algoritmos de otimização inteira global	31
3.6.2 Algoritmos de otimização inteira + Decomposição	31
3.6.3 Decomposição por cenários	32
3.6.4 Aprendizado de Máquina	32
4 – Os métodos SDDiP e DDiP	34
4.1 Métodos tradicionais DDP e SDDP e suas extensões para DDiP e SDDiP	34
4.2 Famílias de cortes para a DDiP e SDDiP	36
4.3 Cortes de Benders	38
4.4 Cortes Inteiros de Otimalidade (Integer Optimality Cuts)	39
4.5 Cortes de Lagrange	39

4.6	Cortes de Benders Reforçados (Strengthened Benders Cuts)	41
4.7	Aplicação da DDiP e da SDDiP no problema de planejamento energético	41
4.7.1	Representação do volume por meio de uma expansão binária	41
4.7.2	Impacto da “binarização” da variável de recurso	42
5	– Estudos de caso em pequena escala	44
5.1	Estudo de caso “tutorial”	46
5.1.1	Características gerais do caso “tutorial” base	46
5.1.2	Representação do volume armazenado no caso “tutorial”	47
5.1.3	Representação de restrições hidráulicas no caso “tutorial”	47
5.1.4	Solução global do problema “tutorial”	48
5.1.5	Análise da função de recurso do problema “tutorial”	52
5.1.6	Solução do problema “tutorial” utilizando cortes de Benders	54
5.1.7	Solução do problema “tutorial” utilizando cortes Inteiros de Otimalidade	55
5.1.8	Solução do problema “tutorial” utilizando cortes de Lagrange	56
5.1.9	Solução do problema “tutorial” utilizando cortes de Benders Reforçados	56
5.1.10	Resumo dos resultados do caso “tutorial-no-constr”	57
5.1.11	Resumo dos resultados do caso “tutorial-cond-constr”	57
5.2	Estudo de caso “reduced-system”	60
5.2.1	Características gerais do caso “reduced-system”	60
5.2.2	Representação do volume armazenado no caso “reduced-system”	61
5.2.3	Representação das restrições hidráulicas no caso “reduced-system” . . .	62
5.2.4	Solução global do problema “reduced-system”	63
5.2.5	Resultados “reduced-system-no-constr” resolvidos por decomposição .	64
5.2.6	Resultados “reduced-system-cond-constr” resolvidos por decomposição	66
6	– Análise das soluções duais	70
6.1	Infinitas soluções duais e solução dual extrema	70
6.2	Impacto da consideração da variável de cópia contínua	79
6.3	Infinitas soluções duais extremas	83
6.3.1	Análise no espaço dual	84
6.3.2	Análise no espaço primal	85
6.4	Busca dos valores duais para construção dos cortes de Lagrange	88
6.4.1	Método do Subgradiente	88
6.4.2	Método de Planos Cortantes	89
6.4.3	<i>Bundle methods</i>	89
7	– Cortes de Lagrange e uma nova variante	91
7.1	Cortes de Lagrange Tradicionais	91
7.2	<i>Square-minimization cuts (SMC)</i>	93

7.3	Cortes exatos não Lagrangianos	93
7.4	Nova variante de corte de Lagrange: <i>Tighter Lagrangian Cuts</i>	94
7.5	Comparação entre cortes lineares Lagrangianos e não-Lagrangianos	99
8	– Avaliação dos cortes de Lagrange	100
8.1	Estudo de caso	100
8.2	Análise comparativa e sensibilidade	100
8.3	Resultados	101
9	– Estudo de casos de grande porte	106
9.1	Descrição do caso	106
9.2	Solução global do problema de grande porte	108
9.3	Decomposição do problema de grande porte linear	109
9.4	Decomposição do problema de grande porte com restrições condicionadas . . .	110
9.4.1	Solução inexata via DDP	110
9.4.2	Solução exata via DDiP	111
9.5	Impacto da expansão binária no problema de grande porte	112
10	– Proposta de decomposição em duas etapas para MS-MILPs	114
10.1	Descrição da metodologia	114
10.1.1	Etapa 1: Determinação de soluções aproximadas	114
10.1.2	Etapa 2: Melhoria da solução aproximada	117
10.2	Resultados computacionais para o problema reduzido	119
10.3	Resultados computacionais para o problema de grande porte	121
10.3.1	Resultados de operação	123
10.3.2	Validação de desempenho em múltiplas instâncias	124
11	– Conclusões	127
	REFERÊNCIAS BIBLIOGRÁFICAS	131
	APÊNDICE A – Formulação do estudo de caso “tutorial”	138
	APÊNDICE B – Resultados do problema “tutorial”	141
B.1	Resultados do problema “tutorial-no-constr”	141
B.1.1	LP-global e MILP-global	141
B.1.2	Cortes de Benders	144
B.1.3	Cortes de Benders Reforçados	145
B.1.4	Cortes Inteiros de Otimalidade	146
B.1.5	Cortes de Lagrange	148

B.2	Resultados do problema com “tutorial-cond-constr”	149
B.2.1	MILP-global	149
B.2.2	Cortes de Benders	150
B.2.3	Cortes de Benders Reforçados	152
B.2.4	Cortes Inteiros de Otimalidade (5 variáveis binárias)	154
B.2.5	Cortes de Lagrange	156
APÊNDICE C – Resultados do problema “reduced-system”		157
C.1	Resultados do problema “reduced-system-no-constr”	157
C.1.1	Cortes de Benders	157
C.1.2	Cortes Inteiros de Otimalidade	158
C.1.3	Cortes de Lagrange	159
C.1.4	Cortes de Benders Reforçado	160
C.2	Resultados do problema “reduced-system-cond-constr”	160
C.2.1	Cortes de Benders	160
C.2.2	Cortes de Lagrange	162
C.2.3	Cortes de Benders Reforçado	163
APÊNDICE D – Exemplo unidimensional função de recurso		164
APÊNDICE E – Sensibilidade para diferentes regras de escolha das soluções iniciais para a decomposição em duas etapas		165

1 Considerações Iniciais

Este capítulo visa apresentar a motivação e os objetivos centrais deste estudo. Primeiramente, será feita uma introdução breve ao contexto do trabalho, acompanhada da exposição dos principais desafios associados. Em seguida, serão delineados os objetivos e as principais contribuições. Por fim, será apresentada a estrutura geral do texto.

1.1 Introdução e Motivação

O planejamento da operação energética visa auxiliar a gestão de recursos de geração de energia em um sistema de potência. Esta gestão deve considerar tanto o atendimento à demanda energética no presente e no futuro, quanto o atendimento de restrições operativas de caráter elétrico, hídrico, ambiental, político, entre outros [1]. O propósito é alocar os recursos de maneira a minimizar os custos, ao mesmo tempo em que se busca mitigar os riscos associados ao não atendimento da demanda futura, com vistas à segurança energética do sistema. Assim, devem-se considerar incertezas associadas a ocorrências futuras, tais como regime de chuvas ou variações na demanda energética, de modo a proporcionar uma resposta robusta. Desse modo, a otimização estocástica é amplamente utilizada para a modelagem e resolução do problema de planejamento energético [2, 3, 4, 5].

Restrições operativas e físicas relevantes de usinas hidrelétricas frequentemente apresentam caráter condicional, isto é, variam de acordo com o estado operativo da usina, de modo que sua representação adequada requer a introdução de variáveis inteiras. Tais restrições podem exercer impacto significativo sobre a operação global do sistema. Nesse contexto, torna-se fundamental representá-las de forma precisa nos modelos de planejamento, de modo a garantir a consistência temporal (*time consistency* [6]). Em outras palavras, as características do problema em estágios futuros, quando planejadas no presente, devem refletir da forma mais fiel possível as condições que serão efetivamente observadas quando esses estágios ocorrerem.

A aplicação conjunta de técnicas de otimização inteira e estocástica ao planejamento energético ainda apresenta desafios significativos. Nas seções seguintes, são discutidas separadamente as principais dificuldades associadas a cada uma dessas áreas, a fim de evidenciar a complexidade resultante de sua integração.

1.1.1 Desafios do Planejamento Energético

Sistemas de potência interligados de grande porte agregam diversas fontes de energia (ex: usinas hidrelétricas, térmicas, eólicas, fotovoltaicas), além de restrições e incertezas futuras, que tornam seu planejamento muito complexo. Esse problema exige modelagem matemática complexa para representar de forma exata ou aproximada cada aspecto do sistema real, além

de técnicas estatísticas para modelar as incertezas do planejamento. Assim, a utilização de ferramentas computacionais para a execução deste planejamento é essencial. Algoritmos de otimização são muito explorados nestas ferramentas de forma a encontrar a operação ótima considerando os aspectos e restrições presentes no sistema.

Assim, existem grandes desafios no planejamento energético relacionados tanto à modelagem quanto ao algoritmo de solução:

- modelagem: quanto mais precisa for a modelagem dos componentes e aspectos do sistema, mais realista será a resposta do modelo. No entanto, aumenta-se também a complexidade de sua resolução, o que pode resultar em tempos computacionais proibitivos ou até mesmo na ausência de um algoritmo capaz de encontrar uma solução satisfatória.
- algoritmo de solução: os algoritmos de solução procuram viabilizar a resolução dos modelos citados acima. Os dois principais pontos críticos em relação ao algoritmo de solução são:
 - robustez computacional, ou seja, o algoritmo deve ser capaz de resolver um razoável número de diferentes instâncias de tamanho real em um tempo aceitável. Algumas características relevantes para a robustez são: rapidez, eficiência e adequabilidade ao processamento paralelo;
 - qualidade da solução, na qual devem ser avaliadas a capacidade de encontrar a solução ótima e as imprecisões associadas.

No contexto do planejamento da operação de médio e longo prazos, os modelos atualmente utilizados incorporam as incertezas por meio de técnicas de otimização estocástica. No entanto, esses modelos geralmente são baseados em restrições lineares considerando apenas variáveis contínuas [2], [7], o que os classifica como problema de Programação Linear (LP - do termo *Linear Programming*) e, conseqüentemente, um problema convexo. Isso limita a representação de restrições operativas mais complexas, como aquelas condicionadas a outras variáveis de decisão, como, por exemplo, regras que determinam a defluência máxima a depender da faixa de volume na qual a usina está operando. Estas regras não podem ser adequadamente modeladas em um LP. Para representá-las, é necessário recorrer a aproximações/linearizações, como o uso de modelos lineares por partes.

Já no planejamento de curto prazo [8], aplicam-se técnicas de otimização linear inteira mista, que permitem a representação de decisões condicionadas. No entanto, as incertezas, embora presentes — especialmente em relação às fontes de energia intermitentes — são frequentemente desconsideradas nesse horizonte temporal.

1.1.2 Desafios da Otimização Linear Inteira Mista

A programação linear inteira mista (MILP - do termo *Mixed Integer Linear Programming*) trata de problemas de otimização nos quais algumas variáveis de decisão devem assumir valores inteiros e/ou binários [9]. A imposição de integralidade torna a região viável não convexa, de forma que a resolução do problema fica consideravelmente mais desafiadora, uma vez que métodos eficientes aplicados a otimização linear contínua não são aplicáveis. Em geral, lidar com variáveis de decisão inteiras recai em condições enumerativas e requer técnicas mais avançadas, como relaxação lagrangiana [10], algoritmos baseados em *branch-and-bound* [11] e heurísticas especializadas [12].

Esse tipo de problema é recorrente em várias áreas, incluindo logística, planejamento de expansão e problema de alocação. No contexto do planejamento da operação energética de curto prazo, uma aplicação amplamente reconhecida é a alocação de unidades (*unit commitment*), que busca modelar o acionamento e desligamento de unidades geradoras, como térmicas, hidráulicas e nucleares.

Embora a otimização linear inteira mista seja bastante explorada na literatura, ainda existem inúmeros desafios na resolução de problemas de grande porte, onde os algoritmos exatos não são capazes de encontrar a solução em tempo computacional aceitável e os algoritmos não exatos, por serem baseados em heurísticas, não garantem uma resposta adequada.

1.1.3 Desafios da Otimização Estocástica

A otimização estocástica [13] busca resolver problemas de otimização em que alguns parâmetros são incertos. Geralmente, está associada a problemas de decisão em dois ou mais estágios, onde a incerteza é revelada a cada estágio temporal. Assim, é necessário encontrar uma política que considere tais incertezas e estime o custo futuro associado a uma decisão.

Para obter uma política adequada é essencial avaliar diversos cenários possíveis, o que pode tornar o problema de grande magnitude. Portanto, algoritmos de otimização estocástica não apenas representam incertezas, mas também devem ser projetados para lidar com alta dimensionalidade. Técnicas de decomposição podem ser empregadas para reduzir o problema. No entanto, essas técnicas possuem premissas em relação ao tipo de problema considerado, o que pode limitar a sua aplicação.

1.2 Objetivos e contribuições

O principal objetivo deste trabalho é estudar o algoritmo de Programação Dinâmica Dual Inteira Estocástica (SDDiP) e sua aplicação ao problema de planejamento energético de médio e longo prazos, visando à representação de restrições condicionais. Além disso,

foca-se na metodologia DDiP (Programação Dinâmica Dual Inteira), que consiste na resolução de problemas estocásticos considerando toda árvore de cenários, ou seja, sem realização de amostragem dos cenários, o que permite observar efeitos de convergência de forma determinística. Tais metodologias combinam os desafios da otimização inteira e da otimização estocástica, possibilitando uma modelagem mais acurada do problema em estudo.

Este trabalho envolveu uma revisão bibliográfica sobre métodos de decomposição para a resolução de problemas de otimização estocástica multiestágio linear inteira mista, bem como sobre a aplicação dessas técnicas a problemas reais, com ênfase no algoritmo SDDiP. Esse estudo fundamentou as principais contribuições da pesquisa, que são apresentadas a seguir:

- Avaliação conceitual das características dos cortes que podem ser empregados no método SDDiP, com foco na exatidão, na complexidade e no desempenho computacional.
- Proposição de uma metodologia para a obtenção de novos cortes para o método SDDiP e DDiP.
- Incorporação de restrições condicionais ao problema de planejamento energético, acompanhada de uma análise em um caso tutorial para demonstrar os impactos dessas restrições na função de custo futuro.
- Aplicação do método DDiP ao problema de planejamento energético estocástico, com avaliação dos métodos de geração de cortes existentes na literatura, bem como da metodologia proposta.
- Proposição de uma metodologia heurística híbrida (DDP + DDiP) que viabiliza a aplicação do método DDiP a problemas de grande porte e sua comparação com os algoritmos empregados na prática.

1.3 Publicações e apresentações do trabalho

L. C. Brandão, A. L. Diniz and B. H. Dias, "*A Two-step Decomposition Approach for Stochastic Mid-term Power Generation Planning with Disjunctive Constraints for Reservoirs*", in **IEEE Access**, doi: 10.1109/ACCESS.2026.3704030.

L. C. Brandão, A. L. Diniz and B. H. Dias, "*A comparative study of cutting strategies within the SDDiP framework: building tighter Lagrangian cuts*", submetido para **Engineering Optimization** inicialmente em 17 de setembro de 2025 e a revisão 1 em 12 de abril de 2026.

L. C. Brandão, A. L. Diniz and B. H. Dias, "*Stochastic Dual Dynamic Integer Programming: finding better Lagrangian Cuts and application on energy planning problem*", apresentado no **XVIIth International Conference on Stochastic Programming (ICSP)**, Paris, França, 2025

L. C. Brandão, A. L. Diniz and B. H. Dias, "*Efficient Generation of Lagrangian Cuts through Boundary Dual Maximization in SDDiP*", apresentado no **XV Brazilian Workshop on Continuous Optimization (BRAZOPT)** , Blumenau, Brasil, 2026

1.4 Organização do texto

A sequência deste documento está organizada da seguinte forma. No Capítulo 2, é apresentada uma introdução ao problema de planejamento energético abordado neste trabalho, bem como uma discussão sobre a importância e os desafios associados à representação de variáveis inteiras nesse contexto. O Capítulo 3 apresenta uma revisão bibliográfica das principais metodologias empregadas na resolução de problemas de otimização estocástica multiestágio com variáveis inteiras, com ênfase nos algoritmos que utilizam variáveis de recurso binárias e em suas aplicações práticas.

No Capítulo 4, são detalhadas a formulação do problema e as metodologias SDDiP e DDiP, bem como os diferentes tipos de cortes utilizados por esses métodos e considerados neste trabalho. O Capítulo 5 apresenta estudos de caso de pequeno porte com o objetivo de avaliar o impacto da consideração de variáveis inteiras na função de recurso e analisar o desempenho de diferentes metodologias propostas na literatura.

O Capítulo 6 analisa, sob uma perspectiva conceitual, o processo de construção dos cortes de Lagrange, destacando as principais características do problema dual associado e seus efeitos sobre a qualidade dos cortes gerados. No Capítulo 7, são apresentados os cortes Lagrangianos e os procedimentos para sua obtenção, incluindo a proposição de um novo tipo de corte, denominado *Tighter Lagrangian Cut*. Em seguida, o Capítulo 8 avalia computacionalmente os diferentes tipos de cortes Lagrangianos, comparando seu desempenho em diversas configurações.

Por fim, o Capítulo 9 apresenta estudos de caso de grande porte e evidencia os desafios associados à sua resolução, tanto de forma monolítica quanto por meio de técnicas de decomposição. No Capítulo 10, é proposta uma heurística de decomposição em duas etapas, que combina a eficiência computacional de métodos aproximativos, como a DDP, com a maior exatidão de técnicas mais complexas, como a DDiP, buscando um equilíbrio entre qualidade da solução e esforço computacional. E o Capítulo 11 reúne as principais conclusões do trabalho e discute suas contribuições, além de indicar temas para pesquisas futuras.

2 Problema Multiestágio Estocástico Linear Inteiro Misto (MS-MILP) e Planejamento Hidrotérmico

Este capítulo apresenta os conceitos centrais necessários ao desenvolvimento deste trabalho. Inicialmente, discute-se o problema de planejamento energético, destacando seus principais aspectos no contexto de sistemas de potência de grande porte. Em seguida, são descritos os componentes e a modelagem adotada para o problema considerado. Posteriormente, introduzem-se os fundamentos da otimização estocástica multiestágio utilizados ao longo do texto. Por fim, são discutidos a motivação e os principais desafios associados à incorporação de programação inteira no planejamento energético de médio e longo prazos.

2.1 Visão geral do planejamento energético

O planejamento energético consiste em um processo complexo de coordenação de diferentes fontes de geração com o objetivo de garantir uma oferta estável, econômica e eficiente de energia elétrica. As usinas hidrelétricas desempenham papel fundamental nesse contexto por combinarem uma geração renovável com capacidade de armazenamento. A capacidade dessas usinas de ajustar a geração de acordo com a necessidade, armazenando água em reservatórios e liberando-a no futuro, proporciona flexibilidade e confiabilidade ao sistema. No entanto, a dependência do regime hidrológico, caracterizado por elevada incerteza no médio e longo prazos, representa um dos principais desafios para a gestão desse recurso.

As usinas térmicas, por sua vez, apresentam geração mais previsível, assumindo-se, como usual no planejamento, a inexistência de restrições relevantes no suprimento de combustível. Elas são fundamentais para o atendimento da carga em períodos de escassez hídrica ou alta demanda. Contudo, possuem custos operacionais elevados, impactos ambientais significativos e restrições operativas que dificultam sua modelagem, especialmente em horizontes de curtíssimo prazo. As usinas eólicas e solares, cada vez mais presentes nos sistemas elétricos, constituem uma fonte limpa e renovável, porém sujeita a elevada variabilidade, sobretudo no curto prazo, exigindo sua complementação por outras fontes ou mecanismos de armazenamento.

Outros recursos energéticos, como a energia geotérmica, também podem fazer parte do portfólio energético. Cada diferente fonte traz suas próprias vantagens e desafios, e seu papel na segurança e eficiência de um sistema de potência interligado.

O planejamento energético exige uma abordagem integrada que considera as características de cada tipo de usina, a capacidade de armazenamento, a rede de transmissão e os aspectos econômicos e ambientais; usualmente é dividido em diferentes horizontes temporais:

- Planejamento de longo prazo: considera horizontes de 5 a 10 anos, com discretização mensal. Seu objetivo principal é a definição de políticas operativas que garantam a

segurança energética futura, mitigando riscos associados às incertezas hidrológicas. Tal política visa à determinação do valor intertemporal do recurso hídrico.

- Planejamento de médio prazo: abrange horizontes de até um ano, com discretização semanal ou mensal. Nesse nível, busca-se maior detalhamento da representação das usinas, da rede de transmissão e da discretização temporal, permitindo a obtenção de políticas operativas mais próximas da realidade e fornecendo subsídios para análises operativas e sinalização de preços mais adequados para serem utilizados no horizonte de curto prazo.
- Planejamento de curto prazo: considera horizonte de até uma semana com discretização horária ou semi-horária. O objetivo é determinar um despacho viável e econômico compatível com a operação em tempo real, servindo como apoio à tomada de decisão e proporcionando um preço de mercado alinhado com a operação econômica do sistema.

Em sistemas de grande porte com elevada participação hidrelétrica, o planejamento de médio e longo prazos torna-se particularmente complexo devido a três fatores principais: *(i)* a incerteza associada às vazões afluentes; *(ii)* o acoplamento temporal decorrente da gestão intertemporal dos reservatórios; e *(iii)* o acoplamento espacial resultante da operação em cascata das usinas, na qual decisões locais impactam o desempenho global do sistema.

Nesse contexto, ferramentas de otimização estocástica multiestágio são amplamente empregadas, pois permitem representar explicitamente a incerteza hidrológica e os efeitos intertemporais da operação dos reservatórios. Exemplos de aplicações em sistemas predominantemente hídricos incluem o Brasil [2][14][8], o Chile [15], a Noruega [16][3] e o Canadá [17][5].

2.2 Componentes e modelagem do problema de planejamento energético

Neste trabalho, considera-se uma modelagem de longo e médio prazos do planejamento energético que contempla os seguintes aspectos:

- horizonte temporal de meses a anos, com discretização semanal e mensal;
- representação individualizada de usinas hidrelétricas e térmicas;
- modelagem de usinas hidrelétricas com ou sem capacidade de regularização, organizadas em cascata em diferentes bacias hidrográficas;
- consideração explícita da incerteza associada às vazões afluentes;
- consideração da produtividade hidráulica variável com a vazão turbinada e da altura de queda;

- representação da demanda por meio de submercados, cada um contendo um subconjunto das usinas do sistema;
- interligações entre submercados, com limites de intercâmbio associados às capacidades de transmissão;
- divisão da carga em patamares (leve, médio e pesado) para capturar variações intra-período;
- inclusão de restrições físicas e operativas hidráulicas e elétricas;
- acoplamento de uma função de custo futuro linear por partes no final do horizonte, representando a valoração da água remanescente.

A modelagem resultante é composta por variáveis contínuas, restrições lineares e lineares por partes, e uma função objetivo linear. A introdução de variáveis inteiras, necessária para representar determinadas restrições operativas, é discutida nas seções seguintes. A formulação detalhada dos componentes descritos baseia-se em [14].

2.3 Caracterização de problemas de otimização estocástica multiestágios

Problemas de otimização estocástica multiestágios podem ser representados por meio de uma árvore de cenários, na qual cada nível corresponde a um estágio temporal t e cada nó neste nível representa uma realização da incerteza no estágio. Pode ser descrito pela equação de Bellman, conforme as Equações (2.1)-(2.3).

$$Q_t(x_{t-1}) = \min_{y_t, x_t} \mathbb{E}_{\xi_t} [c(x_t, y_t, \xi_t) + Q_{t+1}(x_t)] \quad (2.1)$$

$$s.t. \quad y_t, x_t \in \mathcal{Y}(x_{t-1}, \xi_t) \quad (2.2)$$

$$y_t \in \mathbb{R}^n \times \mathbb{Z}^m, x_t \in \mathbb{R}^p \times \mathbb{Z}^q \quad (2.3)$$

Quanto à representação da incerteza, destacam-se duas abordagens principais: (i) árvores completas (CT), nas quais a variável estocástica (ξ_t) possui suporte finito; e (ii) árvores amostrais (ST), utilizadas quando o suporte é infinito, sendo a solução obtida por meio de amostragem sucessiva. A escolha da estrutura da árvore depende do horizonte de planejamento e do nível de detalhamento desejado.

Neste trabalho, consideramos que a função c é linear e o conjunto $\mathcal{Y}$ definido por restrições lineares e restrições de integralidade, de modo que o *subproblema* associado a cada nó corresponde a um problema de otimização linear. Este pode ser classificado em três tipos: (i) com variáveis de decisão contínuas (LP), ou seja, $m = 0, n > 0, q = 0$ e $p > 0$; (ii) com variáveis de decisão inteiras (IP): $m > 0, n = 0, q > 0$ e $p = 0$; (iii) com variáveis de decisão contínuas e inteiras (MILP), neste caso, tem-se $m > 0, n > 0, q > 0$ e $p > 0$.

Em cada subproblema, as variáveis de decisão são particionadas em dois grupos: as *variáveis de recurso* (x_t) e as *variáveis locais* (y_t). As primeiras correspondem a decisões tomadas no período t e que influenciam diretamente o período $t + 1$, enquanto as variáveis locais não produzem esse efeito intertemporal direto. Assim, o subproblema de um estágio/cenário pode ser representado como uma função das variáveis de recurso do estágio anterior ($Q_t(x_{t-1})$), tal função é conhecida como *função de recurso*.

Um Problema Linear Multiestágio Estocástico (MS-LP) se caracteriza quando todos os subproblemas são LP. Nesse caso, a função de recurso associada a cada subproblema é convexa, poliedral e contínua, o que viabiliza a aplicação de métodos computacionalmente eficientes baseados na decomposição do problema e em aproximações lineares sucessivas da função de recurso.

Um Problema Linear Inteiro Misto Multiestágio Estocástico (MS-MILP) caracteriza-se pela presença de pelo menos um subproblema MILP ao longo da árvore de cenários. Nestes casos, a função de recurso de um subproblema é semicontínua inferiormente e poliedral por partes [18]. A resolução de MS-MILPs é, portanto, mais desafiadora, uma vez que envolve simultaneamente o tratamento de funções de recurso não convexas e a otimização inteira nos subproblemas.

2.4 Desafios da programação inteira no planejamento energético: restrições condicionais

A operação de usinas hidrelétricas está sujeita a diversas regras de natureza regulatória, operativa, ambiental e socioeconômica, frequentemente expressas por restrições condicionais dependentes do estado do sistema, como níveis de armazenamento ou vazões. Tais regras impactam diretamente o custo de operação e a valoração da água [19], tornando sua representação no planejamento energético essencial para aproximar o modelo da operação real.

Em [20] são apresentadas as principais regras operativas hidráulicas aplicadas na operação de usinas hidrelétricas, assim como os desafios da representação de regras não convexas no planejamento energético. Exemplos incluem a resolução 193/2024 da Agência Nacional de Águas (ANA) [21] no Brasil, que exige que a vazão máxima média mensal da usina seja $400\text{m}^3/\text{s}$ caso o reservatório esteja operando abaixo de 30% do volume útil; caso esteja entre 30% e 70% do volume útil a vazão defluente máxima permitida é $500\text{m}^3/\text{s}$, e não há restrição de vazão defluente caso o reservatório esteja acima de 70% de seu volume útil. Outras resoluções que possuem exigências condicionadas semelhantes são apresentadas em [22], [23] e [24]. Outro exemplo são as regras condicionadas aplicadas no Chile [25], onde as quantidades mínima e máxima de água destinadas para irrigação e geração elétrica são condicionadas a faixas de volume armazenado do reservatório.

Existem também aspectos físicos das usinas hidráulicas que possuem características condicionadas, como por exemplo a operação do vertedouro: não é fisicamente possível realizar

vertimento caso o volume operativo esteja abaixo do nível do vertedouro. Porém, caso o volume do reservatório esteja acima deste limite, o vertimento pode ser realizado sem restrições. Abordagens contínuas aproximativas podem ser utilizadas; no entanto, elas não evitam que o modelo obtenha um resultado que pode não ser realizado na prática [26].

A representação exata dessas restrições requer a introdução de variáveis binárias, o que inviabiliza sua incorporação direta em modelos tradicionais de MS-LP. A transição para modelos MS-MILP implica no aumento da complexidade de resolução dos subproblemas, perda da convexidade da função de recurso e limitação da aplicabilidade das técnicas clássicas de decomposição, motivando o desenvolvimento de novas abordagens metodológicas.

2.5 Modelagem de restrições condicionadas

Consideramos uma restrição condicionada que pode ser descrita da seguinte forma:

$$y_1 \leq \begin{cases} L_T & \text{se } y_2 < L \\ L_F & \text{caso contrário} \end{cases} \quad (2.4)$$

onde y_1, y_2 são variáveis de decisão contínuas ou inteiras. A variável y_1 representa a variável sujeita à restrição, enquanto y_2 atua como variável de controle. As constantes associadas à restrição são: $L \in \mathbb{R}_+$ que define o limite da condição sobre y_2 ; $L_T \in \mathbb{R}_+$ o limite da restrição quando a condição é satisfeita; e $L_F \in \mathbb{R}_+$, que corresponde ao limite quando a condição não é satisfeita.

Existem diferentes formas de modelar esse tipo de restrição em problemas de otimização. Assume-se que a variável de controle y_2 possui limites de caixa, isto é,

$$0 \leq \underline{Y}_2 \leq y_2 \leq \overline{Y}_2,$$

o que é uma hipótese razoável, dado que, em geral, essa variável representa uma grandeza física, como volume, turbinamento, entre outras.

Nesse contexto, a restrição condicionada pode ser modelada por meio da introdução de duas variáveis binárias adicionais, $b_T, b_F \in \{0, 1\}$, que representam o estado da condição da seguinte forma:

- Se a condição é satisfeita, então $b_T = 1$ e $b_F = 0$
- Se a condição não é satisfeita, então $b_T = 0$ e $b_F = 1$

Essa lógica é imposta por meio da inclusão das seguintes restrições:

$$b_T + b_F = 1 \quad (2.5)$$

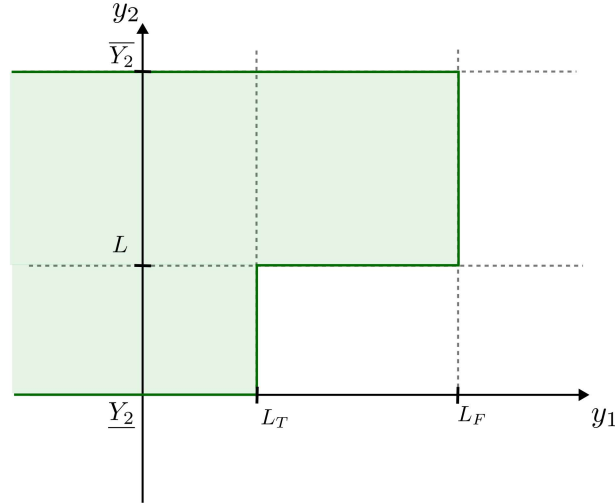


Figura 1 – Região viável para as variáveis y_1 e y_2 considerando a restrição (2.4)

$$b_T \geq 1 - y_2/L \quad (2.6)$$

$$-M \cdot b_F < 1 - y_2/L \quad (2.7)$$

A Equação (2.5) garante que exatamente uma das variáveis binárias assumam valor unitário. A Equação (2.6) força $b_T = 1$ quando $y_2 < L$, enquanto a Equação (2.7) assegura que $b_F = 1$ quando $y_2 \geq L$, sendo $M \in \mathbb{R}$ uma constante suficientemente grande. Como a variável y_2 possui limite superior, um valor suficiente para M é dado por

$$M > \frac{\overline{Y_2}}{L} - 1.$$

Por fim, a restrição condicionada pode ser expressa em função das variáveis binárias da seguinte forma:

$$y_1 \leq L_T \cdot b_T + L_F \cdot b_F \quad (2.8)$$

Com a introdução das variáveis b_T e b_F e das restrições (2.5)–(2.8), a restrição condicionada original (2.4) passa a ser representada no modelo de otimização. A Figura 1 ilustra a região viável no espaço das variáveis (y_1, y_2) , evidenciando seu caráter não convexo. Essa formulação permite a representação de restrições condicionadas de natureza física ou operativa, como aquelas aplicadas às usinas hidrelétricas, conforme discutido anteriormente.

3 Métodos para Resolução de MS-MILPs

Este capítulo apresenta uma revisão bibliográfica sobre métodos para a resolução de problemas de otimização estocástica multiestágio, com ênfase em técnicas de decomposição temporal. Inicialmente, são introduzidos de forma sucinta dois métodos amplamente utilizados na resolução de MS-LPs. Em seguida, são discutidas as principais abordagens da literatura aplicadas a MS-MILPs gerais e a MS-MILPs com variáveis de recurso binárias, em particular o algoritmo SDDiP. Posteriormente, são apresentadas as aplicações e os resultados reportados do método SDDiP. Por fim, são mencionadas outras formas de decomposição para a resolução de MS-MILPs.

3.1 Decomposição de problemas estocásticos multiestágios

Problemas estocásticos multiestágio apresentam dimensões elevadas, uma vez que o número de variáveis cresce exponencialmente com o número de estágios. Essa característica torna indispensável o uso de técnicas de decomposição para viabilizar a sua resolução. Neste contexto, o método *L-shaped* [27] introduziu a decomposição em dois estágios. Posteriormente, a Programação Dinâmica Dual (DDP) [28] foi introduzida para problemas de múltiplos estágios. Desde então, diversas contribuições teóricas e aplicações práticas têm sido desenvolvidas [29, 30, 31, 32]. A ideia central desses métodos consiste em decompor o problema original em subproblemas, resolvidos iterativamente até a obtenção da solução do problema global. Embora a DDP permita resolver problemas que seriam intratáveis sem decomposição temporal, ela está limitada à árvores de cenários de tamanho intermediário.

Com o objetivo de superar essa limitação, os autores de [33] propuseram a Programação Dinâmica Dual Estocástica (SDDP), uma variante de natureza estatística. Este método é amplamente utilizado na prática [2, 4, 34], e diversas extensões e aprimoramentos têm sido propostos ao longo dos anos [35, 36, 37]. A SDDP é particularmente adequada para situações em que a árvore de cenários é muito grande, no caso de modelos de árvore completa (CT), ou mesmo infinita, no caso de modelos de árvore amostral (ST). Sua principal característica é o uso de amostragem de cenários, dispensando a enumeração completa da árvore e conferindo ao método um caráter de convergência estatística.

Apesar de amplamente empregadas, tanto a DDP quanto a SDDP requerem que os subproblemas resolvidos sejam convexos, de modo que a função de recurso associada seja convexa. Sob essa hipótese, esses métodos constroem sucessivas aproximações lineares que fornecem limitantes inferiores para cada função de recurso, até que um critério de parada seja satisfeito [38, 39]. Ressalta-se que a aplicação dessas técnicas utiliza como premissa a convexidade dos subproblemas estando restrita a problemas de programação linear estocástica multiestágios (MS-LPs) [40].

3.2 Decomposição temporal em MS-MILP

Diversos trabalhos propõem o uso de técnicas de convexificação para viabilizar a aplicação dos métodos tradicionais DDP e SDDP em problemas de programação inteira mista multiestágios. Uma abordagem recorrente consiste na convexificação da função de recurso, seja por meio da relaxação linear das variáveis inteiras [41], seja pela aplicação de relaxação Lagrangiana a um subconjunto das restrições [42]. Nessas abordagens, não há garantia teórica de convergência, embora, para determinadas aplicações, as soluções obtidas apresentem *gap* dentro de tolerâncias aceitáveis [43, 44, 45]. Outra estratégia consiste na convexificação de relações não lineares ou não convexas, como aquelas associadas à função de produção hidrelétrica, conforme discutido em [41, 46, 47]. Embora a DDP e a SDDP convirjam para o problema convexificado, as aproximações introduzidas podem comprometer a qualidade da solução final, ou seja, a solução obtida pode não estar aderente à solução ótima do problema real. Além disso, restrições disjuntivas, como aquelas relacionadas à alocação de unidades e às restrições hidráulicas condicionais apresentadas na Seção 2.4, são particularmente difíceis de representar adequadamente por meio de convexificações.

Diante dessas limitações, foram propostos métodos especificamente voltados à resolução de problemas MS-MILP, que tratam explicitamente a natureza inteira mista do problema. Em geral, esses métodos exploram propriedades estruturais específicas e assumem determinadas premissas, conforme resumido a seguir:

- Quando as variáveis de recurso são binárias, a função de recurso pode ser representada por um poliedro convexo. Métodos que exploram essa propriedade são apresentados em [48, 49, 50]. Esse caso será discutido em maior detalhe na Seção 3.3.
- No caso em que o problema do segundo estágio é inteiro e as variáveis de recurso são contínuas, [51] propõe a construção de cortes baseados em dualidade para problemas inteiros, resultando em funções do tipo *Chvátal*. Entretanto, esse trabalho não apresenta resultados numéricos.
- Quando a função de recurso é monótona, o método proposto em [52] adapta a SDDP substituindo os cortes lineares por funções em degrau, *Step cut*, construídas a partir da solução ótima dos subproblemas MILP. Os autores demonstram convergência do método sob as hipóteses usuais da SDDP.
- Para problemas com recurso contínuo e completo (*complete continuous recourse*), mesmo na presença de variáveis inteiras, a função de recurso é Lipschitz contínua com constante L [53]. Nesse contexto, são propostos cortes não lineares baseados em normas reversas e na constante de Lipschitz, *Reverse Norm cuts*, que fornecem aproximações inferiores da função de recurso. Alternativamente, [54] utiliza cortes baseados no Lagrangiano aumentado (*Augmented Lagrangian cut*), com funções de regularização, mostrando que

existe um coeficiente de penalidade finito capaz de eliminar o *gap* de dualidade. Ambos os tipos de cortes podem ser aplicados nos casos CT e ST, com convergência assintótica garantida, permitindo que os subproblemas sejam MILPs sem restrições adicionais sobre as variáveis de decisão ou de recurso. A incorporação desses cortes no subproblema antecessor requer a introdução de variáveis binárias auxiliares, aumentando a complexidade computacional, embora mantendo a estrutura MILP.

- Quando a função de recurso não é Lipschitz contínua, [55] propõe a modificação dos subproblemas de forma que a função de recurso seja substituída pelo seu envelope Lipschitz. Essa regularização é obtida pela introdução de uma variável adicional que replica a variável de recurso, juntamente com a relaxação da restrição de cópia via penalização na função objetivo. Para um fator de penalidade suficientemente grande, as regiões viáveis das funções original e regularizada coincidem. Esse procedimento permite tratar problemas MS-MINLP. Após a regularização, os autores constroem cortes de conjugação generalizada (*Generalized Conjugacy cut*), obtidos a partir do Lagrangiano dual aumentado com a mesma função de regularização.
- Sob hipóteses específicas sobre o envelope convexo da função de recurso, [56] propõe os chamados cortes escalonados (*Scaled cut*) no contexto da DDP. Esses cortes são derivados da estimação de parâmetros (α, β, τ) em expressões não lineares da forma $\alpha + \beta x + \tau \phi(x)$, onde $\phi(x)$ é uma aproximação inferior da função de recurso real $Q(x)$. A função $\phi(x)$ é escolhida como a aproximação construída na iteração anterior. Após a determinação desses parâmetros para cada cenário, constrói-se um corte linear correspondente à agregação:

$$Q(x) \geq \frac{\mathbb{E}(\alpha) - \mathbb{E}(\beta)x}{1 + \mathbb{E}(\tau)}$$

Embora os cortes finais sejam lineares, a estimação dos parâmetros exige a resolução de problemas auxiliares de grande porte, implicando elevado esforço computacional. Ainda assim, esses problemas podem ser resolvidos em paralelo. Os autores demonstram que os cortes escalonados são capazes de reproduzir o envelope convexo da função de recurso de MILPs e, em testes numéricos para problemas MS-MILP de dois estágios, apresentam *gap* de convergência menores do que métodos baseados em [50], ainda que com maior custo computacional.

Entre os trabalhos mais recentes que não estão publicados, podem ser citados:

- Em [57], é proposto um *framework* para a geração de cortes de Lagrange baseados em funções de regularização, permitindo a obtenção de cortes com diferentes propriedades. Os resultados são apresentados apenas para o caso em que as variáveis de recurso são binárias e indicam que, para instâncias de pequeno porte, os cortes regularizados melhoram a convergência da SDDiP. Os autores destacam, contudo, a necessidade de aprimorar a

eficiência computacional para viabilizar a aplicação do método em problemas de maior escala.

- O trabalho em [58] aprofunda a análise da estrutura da função dual e propõe cortes não lineares baseados na reformulação das restrições de não antecipatividade por meio de funções ReLU (do inglês, Rectified Linear Unit). Esses cortes têm potencial para recuperar o envelope convexo da função de recurso. Os resultados apresentados consideram apenas casos com variáveis de recurso binárias. Neste contexto, o *ReLU cut* assume forma linear e apresenta desempenho superior em relação aos cortes de Benders, Benders reforçados e cortes Inteiros de Otimalidade.
- Quando a dimensão da variável de recurso é moderada, [59] propõe a elevação da dimensão do espaço das variáveis de recurso com o objetivo de reduzir o grau de não convexidade da função de recurso nesse novo espaço, facilitando sua aproximação por cortes lineares. São construídos cortes elevados (*Lifted cut*), cuja validade é controlada por variáveis binárias associadas a regiões específicas do espaço. Adicionalmente, os autores introduzem cortes Lagrangianos Pareto-ótimos (*Pareto-optimal Lagrangian cut*), obtidos pela maximização do valor do corte em pontos do interior da região viável. Os resultados indicam que a elevação do espaço permite alcançar convergência com tolerâncias satisfatórias em menor tempo quando comparada à utilização de expansões binárias para representar variáveis de recurso não binárias com a SDDiP.

A principal vantagem das abordagens mencionadas é a existência de garantias formais de convergência do método SDDP sob as hipóteses consideradas. No entanto, tais abordagens frequentemente geram cortes que induzem funções não convexas [51, 52, 53, 55, 58]. Apesar das propriedades teóricas de convergência, a presença desses cortes aumenta substancialmente a dificuldade computacional do problema, uma vez que impõe restrições não convexas aos subproblemas. Consequentemente, a sua aplicabilidade fica, em geral, limitada a instâncias de menor dimensão. Além disso, alguns dos cortes propostos, ainda que assumam forma linear [51, 55, 56, 58, 59], apresentam elevada complexidade de cálculo, o que contribui para um aumento significativo do custo computacional global.

Em síntese, a literatura recente evidencia um esforço significativo no desenvolvimento de métodos de decomposição temporal capazes de lidar explicitamente com a natureza inteira mista dos problemas estocásticos multiestágios. As abordagens propostas diferem quanto às hipóteses estruturais assumidas, ao tipo de regularização empregada e à forma de construção dos cortes, refletindo um compromisso entre generalidade teórica, qualidade da aproximação da função de recurso e esforço computacional. Embora avanços importantes tenham sido alcançados, especialmente na recuperação do envelope convexo da função de recurso e na redução do *gap* de convergência, a aplicação desses métodos a problemas de grande escala ainda impõe desafios relevantes. Esses desafios motivam a investigação de estratégias que conciliem rigor teórico,

eficiência computacional e aderência às características físicas e operativas dos problemas reais de planejamento, tema que será explorado nas seções subsequentes.

3.3 Decomposição temporal em MS-MILP com variáveis de recurso binárias

Uma estratégia relevante para a aplicação de métodos de decomposição onde os subproblemas são MILPs consiste em restringir as variáveis de recurso ao domínio binário. Nesse caso, a função de recurso passa a ser uma função definida sobre variáveis binárias, a qual pode ser representada exatamente, em todo o seu domínio, por um poliedro convexo. Essa propriedade permite a construção de uma aproximação inferior da função de recurso com cortes lineares, viabilizando a aplicação de métodos de decomposição como a DDP e a SDDP, desde que os cortes gerados satisfaçam critérios de validade e exatidão.

Uma das primeiras abordagens para a resolução de MS-MILPs com variáveis de recurso binárias foi proposta em [48]. Os autores introduzem o método denominado *integer L-shaped* para problemas de dois estágios, cuja formulação pode ser estendida ao caso multiestágio. O método é desenvolvido no contexto de árvores do tipo CT e considera, inicialmente, todas as variáveis de decisão do primeiro estágio como binárias. No entanto, sua aplicação também é válida quando apenas as variáveis que representam recurso no estágio subsequente são binárias, permitindo que as demais variáveis sejam inteiras ou contínuas. O subproblema de segundo estágio pode, portanto, conter variáveis contínuas e/ou inteiras, caracterizando um MILP.

A metodologia baseia-se na construção dos chamados cortes Inteiros de Otimalidade (*Integer Optimality cut*). A obtenção desses cortes requer o conhecimento de um limite inferior L para o valor ótimo do subproblema de segundo estágio para qualquer decisão do primeiro estágio. Tal limite é utilizado no cálculo de inclinações que garantem a validade dos cortes; quanto mais preciso for o valor de L , mais próxima a função poliedral construída estará do envelope convexo exato da função de recurso. Os autores discutem diferentes estratégias para a determinação desse limite. Detalhes adicionais sobre a construção dos cortes inteiros de otimalidade são apresentados na Seção 4.4.

Um aprimoramento dessa abordagem é apresentado em [49], onde os autores exploram os pontos já visitados ao longo do processo iterativo de decomposição para refinar o valor do limite inferior. Nesse caso, passa-se a considerar um limite L_i associado a cada ponto i do domínio das variáveis de recurso. Contudo, os resultados numéricos indicam que o uso de cortes menos relaxados não resulta em ganhos significativos de desempenho quando comparado aos cortes originais. Os melhores resultados foram observados quando os cortes inteiros são combinados com cortes de Benders, que embora forneçam aproximações relaxadas da função de recurso, carregam informações sobre direções de descida.

Mais recentemente, [50] introduz o método denominado Programação Dinâmica Dual Inteira Estocástica (SDDiP), voltado à resolução de MS-MILPs e aplicável tanto a árvores CT

quanto ST. A proposta consiste em adaptar a metodologia da SDDP sob duas premissas centrais:

- todas as variáveis de recurso são binárias, garantindo que a função de recurso possa ser representada por uma função poliedral convexa;
- o problema resolvido possui recurso completo (*complete recourse*).

Inspirada nos métodos DDP e SDDP, na SDDiP também é feita uma decomposição em subproblemas. No entanto, a formulação tradicional do subproblema é alterada pela adição de uma variável de cópia (para cada variável de recurso) e uma restrição que iguala a variável ao valor decidido anteriormente (restrição de cópia). A partir dessa reformulação, os autores propõem a construção de cortes de Lagrange (*Lagrangian cut*), obtidos a partir dos valores duais associados às restrições de cópia. É demonstrado que, para tal estrutura de problema, há dualidade forte, o que torna estes cortes válidos e exatos (*tight*).

Os autores também destacam que, quando as variáveis de recurso não são binárias, mas contínuas ou inteiras gerais, pode-se recorrer a uma expansão binária, o que aumenta a dimensão do domínio da função de recurso, mas torna o problema compatível com a metodologia da SDDiP. Entretanto, essa estratégia pode dificultar a resolução do subproblema MILP tanto do ponto de vista matemático quanto computacional [60], visto que o método de *branch-and-bound* tende a explorar um número significativamente maior de nós quando as variáveis inteiras são representadas por expansões binárias, em comparação com formulações que utilizam variáveis inteiras gerais.

Além disso, os autores apresentam os cortes de Benders Reforçados (*Strengthened Benders cut*), que, embora sejam válidos, não são exatos. Ainda assim, esses cortes podem apresentar bom desempenho prático. O detalhamento da construção desses cortes é apresentado nas Seções 4.5 e 4.6, respectivamente.

Por fim, estudos de caso realizados com o método proposto indicam uma convergência satisfatória da SDDiP quando se utilizam os cortes de Lagrange apresentados. Os cortes de Benders Reforçados também demonstram bons resultados quando empregados em conjunto com outros tipos de cortes exatos, como, por exemplo, os cortes Inteiros de Otimalidade.

Em [61], os autores propõem um refinamento dos cortes de Lagrange, denominados cortes de Minimização dos Quadrados (*Square-Minimization cut* – SMC). Esses cortes podem ser interpretados como variações dos cortes de Lagrange tradicionais, cujos coeficientes são determinados de modo a minimizar a soma dos seus valores ao quadrado. Os resultados apresentados indicam que os SMCs são capazes de melhorar tanto a taxa de convergência quanto o tempo computacional da SDDiP quando comparados aos cortes de Lagrange convencionais.

Na literatura mais recente, trabalhos não publicados ou em fase de pré-publicação, conforme mencionado anteriormente, como [58], [57] e [59] abordam problemas gerais de

MS-MILPs, com ênfase particular em formulações com variáveis de recurso binárias (SDDiP), nas quais concentram a maior parte de seus resultados.

Nesse contexto, em [58], os autores propõem uma metodologia para a obtenção de cortes por meio da rotação de um corte inteiro de otimalidade, em que a rotação é definida a partir da melhor aproximação possível do epígrafo do problema linearmente relaxado. Argumenta-se que essa abordagem apresenta vantagens computacionais relevantes, por basear-se exclusivamente na resolução de problemas lineares, resultando em maior eficiência tanto em tempo computacional quanto na qualidade dos cortes, quando comparada aos cortes de Lagrange. De forma similar, os cortes Pareto-ótimos apresentados em [59] também exploram rotações de um corte de Lagrange arbitrário, tal rotação é obtida pela resolução de um problema de otimização convexa. Por sua vez, [57] propõe um *framework* que incorpora uma função de regularização na determinação dos multiplicadores duais dos subproblemas MILP.

Em conjunto, esses três trabalhos têm como objetivo o fortalecimento dos cortes de Lagrange no contexto da SDDiP, visando acelerar a convergência do método, propósito alinhado ao desta tese. Além disso, todos foram divulgados durante o desenvolvimento desta pesquisa, evidenciando a atualidade e a relevância do tema. O fato de esses trabalhos ainda se encontrarem em fase de *preprint* e ainda não terem sido publicados reflete o caráter emergente da área e a elevada complexidade dos desafios teóricos e computacionais envolvidos.

3.4 Resumo dos cortes propostos para MS-MILPs

A Tabela 1 sintetiza, em ordem cronológica, os trabalhos discutidos nas Seções 3.2 e 3.3. A coluna “Exato para MS-MILPs” indica se o corte permite uma reconstrução precisa (*tight*) da função de recurso em problemas MS-MILPs gerais, sob hipóteses relativamente fracas. Já a coluna “Exato para SDDiP” indica se o corte fornece uma reconstrução exata para MS-MILPs cujas variáveis de recurso são binárias. A coluna “Linear” informa se o corte pode ser representado por meio de uma restrição linear com variáveis contínuas. Por fim, a coluna “Complexidade de cálculo” reflete o grau de dificuldade associado à implementação e ao custo computacional necessário para a obtenção de cada tipo de corte.

De modo geral, cortes lineares e de menor complexidade computacional tendem a ser mais aplicáveis na prática, especialmente em problemas de grande porte. Contudo, esses cortes usualmente aproximam a função de recurso de forma mais gradual, exigindo um maior número de iterações para a convergência. Em contrapartida, cortes não lineares, embora mais complexos e computacionalmente onerosos, costumam ser restritos a problemas de menor dimensão, mas apresentam maior capacidade de aproximar funções de recurso “não convexas” de maneira mais eficiente.

Corte Proposto	Exato para MS-MILPs	Exato para SDDiP	Linear	Complexidade de cálculo
<i>Benders' cut</i> [62, 41]	Não	Não	Sim	Muito baixa
<i>Integer Optimality cut</i> [48]	Não	Sim	Sim	Muito baixa
<i>Augmented Lagrangian cut</i> [54]	Sim	Sim	Não	Muito Alta
<i>Step cut (Midas)</i> [52]	Sim	Sim	Não	Baixa
<i>Lagrangian cut</i> [41, 50]	Não	Sim	Sim	Média
<i>Strengthened Benders cut</i> [50]	Não	Não	Sim	Baixa
<i>Generalized Conjugacy cut</i> [55]	Sim	Sim	Não	Muito Alta
<i>Reverse norm cut</i> [53]	Sim	Sim	Não	Muito Alta
<i>Scaled cut</i> [56]	Não	Sim	Sim	Muito Alta
<i>Square-Minimization cut</i> [61]	Não	Sim	Sim	Alta
<i>ReLU Lagrangian cut</i> [58]	Sim	Sim	Não	Muito Alta
<i>Lifted Lagrangian cut</i> [59]	Não	Sim	Não	Muito Alta
<i>Pareto-Optimal Lagrangian cut</i> [59]	Não	Sim	Sim	Alta

Tabela 1 – Resumo dos tipos de cortes utilizados para decomposição temporal de MS-MILPs

3.5 Aplicações da DDiP e SDDiP

Em [63], a SDDiP é aplicada a um problema multiestágio de alocação de unidades térmicas sob incerteza. As variáveis de recurso associadas ao status das máquinas são naturalmente binárias, enquanto variáveis contínuas relacionadas ao despacho — utilizadas em restrições de rampa — são representadas por meio de expansão binária. Para viabilizar a aplicação em instâncias de maior porte, os autores adotam três estratégias principais: (i) uso do método de níveis (*level method*) para a resolução do problema dual Lagrangiano, necessário à construção dos cortes de Lagrange; (ii) processamento paralelo no passo *backward*; e (iii) uma abordagem híbrida SDDiP–SDDP, em que, a partir de determinado estágio, as restrições de integralidade são relaxadas. Os resultados indicam que o método de níveis supera o subgradiente na construção dos cortes de Lagrange. Embora esses cortes apresentem convergência superior, seu custo computacional é elevado; em instâncias maiores, os cortes de Benders alcançam *gap* satisfatório com tempos de solução mais razoáveis.

Em [43], os autores utilizam a DDiP para resolver um problema determinístico de planejamento da expansão eletroenergética de grande porte, envolvendo variáveis de recurso inteiras e contínuas, ainda que sem garantia formal de convergência. São avaliados cortes de Benders, de Lagrange e de Benders Reforçados, todos alcançando convergência com *gap* inferior a 1%. Os cortes de Benders apresentam o menor tempo computacional, seguidos pelos cortes de Benders Reforçados e, por fim, pelos cortes de Lagrange. Para instâncias maiores, apenas resultados com cortes de Benders são reportados.

Em [64], a SDDiP é aplicada ao problema estocástico de dimensionamento de lotes sem limites de capacidade. Para melhorar a eficiência computacional, os autores empregam: (i) agregação de nós da árvore de cenários, resolvendo múltiplos períodos em um único MILP; (ii) iterações iniciais com SDDP relaxado ou SDDiP com cortes de Benders; e (iii) uma variante de cortes de Benders Reforçados, cujas inclinações são obtidas a partir de um problema dual com desigualdades válidas adicionais. Os resultados evidenciam limitações de escalabilidade da SDDiP, mitigadas pelas estratégias propostas.

Em [65], os autores abordam um problema MS-MILP de planejamento da operação energética com importantes restrições não convexas, incluindo volumes mínimos para vertimento, relações não lineares entre geração e turbinamento e restrições de acionamento de usinas. A SDDiP é aplicada em duas variantes: com variáveis de recurso binárias (via expansão binária) e com variáveis de recurso binárias e contínuas. Embora esta última não possua garantia de convergência, apresenta custo computacional significativamente menor. São avaliados cortes de Lagrange, cortes Inteiros de Otimalidade, cortes de Benders e cortes de Benders Reforçados, sendo que, para variáveis de recurso contínuas, apenas os dois últimos são aplicáveis. Os autores relatam dificuldades práticas com os cortes de Lagrange e indicam a combinação de cortes de Benders Reforçados com cortes Inteiros de Otimalidade como uma solução de compromisso eficiente.

Em [66], a SDDiP é empregada no planejamento da expansão de sistemas elétricos com integração de *microgrids* baseadas em renováveis, hidrogênio e baterias, considerando incertezas em múltiplas escalas temporais. Na aplicação da SDDiP os autores realizam uma binarização das variáveis inteiras e utilizam os cortes de Benders Reforçados, devido à sua eficiência computacional, não são mostrados resultados para os cortes de Lagrange. Comparações com a formulação como MILP único e com o método de *Progressive Hedging* mostram que, à medida que o número de cenários cresce, apenas a SDDiP mantém convergência satisfatória.

Uma aplicação similar com decisões de planejamento envolvendo a inserção de hidrogênio como fonte de geração foi apresentada em [67]. Neste trabalho, os autores utilizam a SDDiP com cortes de Benders Reforçados em conjunto com cortes Inteiros de Otimalidade para garantir a convergência. Além disso, é proposta a introdução de restrições auxiliares para evitar decisões degeneradas. Os resultados mostrados incluem 3 estágios e até 125 cenários por estágio. Para esta mesma aplicação, [68] propõe uma formulação com objetivo de obter a melhor relação

entre tipos e formas de armazenamento (curto prazo e longo prazo) para atender variações das gerações intermitentes assim como as sazonalidades de mais longo prazo. Os autores aplicam a SDDiP para a resolução do problema, porém utilizam apenas os cortes de Benders Reforçados, os resultados proporcionam convergência satisfatória em tempos computacionais muito melhores quando comparados com a resolução em MILP único, porém apenas casos pequenos são testados.

Em [69], a SDDiP é aplicada ao planejamento multiestágio para rede elétrica com geração distribuída sob incerteza, em um problema cujo recurso é naturalmente binário. Utilizando exclusivamente cortes de Benders Reforçados, os autores obtêm resultados satisfatórios tanto para instâncias pequenas quanto grandes. Já em [70], a SDDiP é aplicada a um problema de planejamento da operação de médio prazo com restrições horárias. Os autores comparam cortes de Lagrange e cortes de Benders tradicionais, mostrando que, embora os cortes de Benders sejam computacionalmente mais eficientes, não produzem convergência satisfatória, ao passo que os cortes de Lagrange, apesar do maior custo computacional, fornecem soluções de melhor qualidade.

Em [71], a SDDiP é utilizada em um problema estocástico de planejamento da produção de curto prazo e grande porte, caracterizado por decisões discretas. Diante das dificuldades computacionais, os autores recorrem a estratégias semelhantes às de [64] e utilizam cortes de Benders Reforçados com variáveis de recurso contínuas, obtendo convergência satisfatória.

Em [72], os autores modelam um problema estocástico multiestágio de manutenção de unidades hidrelétricas, considerando incertezas em preços e afluências. A SDDiP é aplicada com cortes de Lagrange, alcançando convergência e bons resultados, embora apenas para instâncias de pequeno porte.

Por fim, em [61], a DDiP é aplicada a um problema de otimização de interrupções elétricas voltado à mitigação de riscos de incêndios florestais. Trata-se de uma formulação de dois estágios, com recurso binário que indica a desconexão de componentes da rede. Os autores utilizam cortes de Lagrange e os *Square-Minimization Cuts* (SMC), mostrando melhorias tanto na taxa de convergência quanto no tempo computacional. Uma extensão multiestágio é apresentada em [73], na qual são comparados cortes de Benders, Benders Reforçados, Lagrange e SMC, com variáveis de cópia contínuas e inteiras. Os autores concluem que o uso de variáveis de cópia inteiras é vantajoso e que os cortes de Benders Reforçados e SMC são os mais adequados para a aplicação considerada.

Entre essas aplicações, diversos autores concluem que cortes não exatos, como os cortes de Benders e Benders Reforçados, apresentam uma melhor relação entre qualidade da solução e esforço computacional [65, 43, 69, 71, 66, 67]. A utilização dos Cortes Lagrangianos Tradicionais é frequentemente acompanhada por estratégias destinadas a atenuar o seu elevado custo computacional, como a sua combinação com cortes de Benders nas iterações iniciais ou nas fases finais do algoritmo [63, 64], ou ainda pela redução do horizonte de planejamento ou do número de estágios [64, 70]. Assim, embora os cortes lagrangianos sejam exatos, a sua aplicação

permanece limitada por questões de eficiência computacional. Em síntese, a metodologia SDDiP apresenta um elevado potencial de aplicação, mas ainda existem desafios importantes a serem superados para viabilizar a sua utilização em problemas de grande porte, especialmente no que se refere à derivação eficiente de cortes de Lagrange.

3.6 Outras formas de decomposição para MS-MILPs

Na sequência serão mostradas abordagens diferentes às descritas anteriormente para resolução de um MS-MILP:

3.6.1 Algoritmos de otimização inteira global

Quando se trata de problemas com variáveis inteiras, métodos que se baseiam em reduções consecutivas do espaço de soluções são muito utilizados. Um método muito importante é o *branch-and-bound*, que pode ser muito eficiente. Alguns autores adaptaram o *branch-and-bound* para resolver um MS-MILP, como no trabalho [74] que propõe uma alteração na escrita do problema de forma a se obter uma função de recurso constante por partes. Assim, dentro de limites definidos, garante-se uma quantidade finita de partes descontínuas, assegurando a convergência finita do método. Essa metodologia é aplicada para problemas de dois estágios que utilizam CT com função de recurso inteira (segundo estágio inteiro puro). O trabalho [75] mostra que uma redução da dimensão do espaço pode ser aplicada para melhorar a eficiência computacional e uso de memória em [74], o que permite aplicação desta metodologia em problemas de dimensões maiores. No entanto estes métodos não são trivialmente escaláveis para problemas inteiros mistos multiestágios e/ou que necessitam de representação de variável aleatória com suporte contínuo.

3.6.2 Algoritmos de otimização inteira + Decomposição

Formas espertas de conciliar técnicas de decomposição juntamente com o *branch-and-bound* também foram propostas. A ideia é decompor o problema MS-MILP em subproblemas (MILPs reduzidos). Estes subproblemas são resolvidos iterativamente por um método de programação disjuntiva [76], como o *branch-and-bound*. No entanto, informações adicionais do *branch-and-bound*, obtidas antes de sua convergência, são aproveitadas no processo iterativo principal:

- na solução de um subproblema de segundo estágio, os nós folha da árvore de busca do *branch-and-bound* caracterizam um envelope convexo do espaço de solução e podem ser utilizados na construção do corte para o estágio anterior.

- na solução de um subproblema de primeiro estágio, soluções viáveis porém não ótimas, obtidas durante o processo de *branch* podem ser utilizadas para antecipar a iteração, sendo passadas para o estágio seguinte.

Vários autores propõem metodologias que fazem uso deste recurso [77, 78, 79]. Já outros autores usam esta técnica construindo cortes parametrizados de *Gomory* tanto na decomposição (composição da função de recurso) quanto na resolução dos MILPs do segundo estágio (processo de ramificação) [80, 81]. Apesar destes métodos terem a vantagem da não resolução dos MILPs até a otimalidade, aproveitando informações obtidas durante o processo de disjunção, são mais adequados quando as variáveis de recurso são binárias e/ou as decisões inteiras têm características lógicas. Além disso, tais trabalhos só foram aplicados para problemas de dois estágios e a sua generalização para problemas multiestágios não é trivial.

3.6.3 Decomposição por cenários

Além da decomposição de Benders, a decomposição por cenários também é amplamente utilizada. Inicialmente aplicada para resolução de MS-LP [82], em [83] é utilizada para resolução para MS-MILPs onde são mostrados resultados para dois estágios, embora os autores relatem que o método é aplicável para múltiplos estágios. Para cada cenário resolve-se o problema dual onde é feita a relaxação Lagrangiana das restrições de não-antecipatividade (restrições que obrigam as decisões de primeiro estágio a serem iguais). Pelo teorema da dualidade fraca o resultado do problema dual é um limite inferior para o valor ótimo.

Ao se resolver o lagrangiano dual de todos os cenários encontram-se vários valores para as variáveis primais do primeiro estágio x_i^* (já que as restrições de não antecipatividade estão relaxadas). Uma média $\bar{x}_i^* = \sum p_s x_{i,s}^*$ destes valores é feita, ponderada pela probabilidade p_s de cada cenário, e restrições de limites (*bounds*) são impostas criando-se duas ramificações (*branches*) tal que: $x_i \leq \lfloor \bar{x}_i^* \rfloor$ e $x_i \geq \lfloor \bar{x}_i^* \rfloor + 1$. O processo continua resolvendo os nós criados pelo processo de ramificações até que seja encontrada uma solução $\bar{x}_i^*$ viável para o problema primal e, portanto, ótima.

Em [84] a aplicação desta metodologia é feita em um problema MS-MILP onde as decisões inteiras estão relacionadas ao problema de alocação de unidades (*unit commitment*), na busca de uma operação ótima para um sistema de potência. O método foi aplicado para um problema de grande porte, mas de apenas dois estágios, e os autores utilizam várias estratégias para reduzir o tempo computacional principalmente no cálculo do problema lagrangiano dual.

3.6.4 Aprendizado de Máquina

Outra abordagem para lidar com o problema MS-MILP é por meio de meta-heurísticas. Em [85], os autores propõem resolver um problema de dois estágios combinando técnicas de

decomposição com aprendizado supervisionado para prever soluções do segundo estágio. Eles destacam os significativos ganhos em termos de tempo computacional em comparação com os métodos exatos, especialmente quando o segundo estágio é computacionalmente custoso e soluções de problemas semelhantes resolvidos anteriormente estão disponíveis.

4 Os métodos SDDiP e DDiP

Conforme discutido no capítulo anterior, a metodologia SDDiP é um desenvolvimento relativamente recente e tem demonstrado potencial em diversas aplicações. Entretanto, sua utilização em problemas de grande porte ainda enfrenta limitações relacionadas principalmente ao custo computacional, o que evidencia a existência de oportunidades para o aprimoramento da técnica e justifica sua investigação neste trabalho.

Assim, este capítulo tem como objetivo apresentar os métodos DDiP e SDDiP. Inicialmente, são revisados os fundamentos dos algoritmos DDP e SDDP e suas extensões para problemas com variáveis inteiras. Na sequência, são descritas as principais abordagens propostas na literatura para a geração de cortes, enfatizando suas formulações e propriedades. Por fim, é abordada a representação de variáveis de recurso contínuas por meio de expansões binárias e sua aplicação no contexto do planejamento energético.

4.1 Métodos tradicionais DDP e SDDP e suas extensões para DDiP e SDDiP

Os métodos DDiP e SDDiP são baseados em suas versões contínuas, DDP [27], [28] e SDDP [33], que foram desenvolvidas para resolver problemas de otimização estocástica que, em geral, envolvem múltiplos estágios de decisão, o que frequentemente resulta em problemas de grande porte. Assim, estes métodos utilizam a formulação dinâmica, conforme descrito na Seção 2.3, para decompor o problema original em subproblemas menores, geralmente associados a um estágio de decisão e a um cenário estocástico. Cada subproblema passa a ser descrito como uma função de variáveis decididas em estágios anteriores, mas que impactam o presente estágio, que são chamadas de *variáveis de recurso*. Esta função é chamada de *função de recurso* ou *função de custo futuro*, pois, para um dado valor do recurso, indica o custo daquele estágio em diante.

Os algoritmos da DDP ou SDDP se baseiam na resolução iterativa destes subproblemas, construindo-se sucessivas aproximações lineares da função de recurso, também chamadas de *cortes*. À medida que novas aproximações são adicionadas, a representação da função de recurso torna-se mais precisa e o método atinge a *convergência*. Cada iteração se resume a dois passos:

- *Passo Forward*: ilustrado na Figura 2, lado esquerdo, consiste na resolução dos subproblemas do estágio inicial até o estágio final, de forma encadeada, ou seja, as decisões tomadas no estágio t são levadas como variáveis de recurso para o estágio $t + 1$, e assim por diante. Assumindo a propriedade de recurso completo (*complete recourse* [86]), esta solução proporciona uma solução viável para o problema original. O custo desta solução é chamado de *limite superior* ou *zsup* da iteração, pois limita superiormente o custo ótimo.
- *Passo Backward*: ilustrado na Figura 2, lado direito, consiste na resolução dos subproblemas do estágio final até o inicial, onde para cada estágio são construídos um ou mais

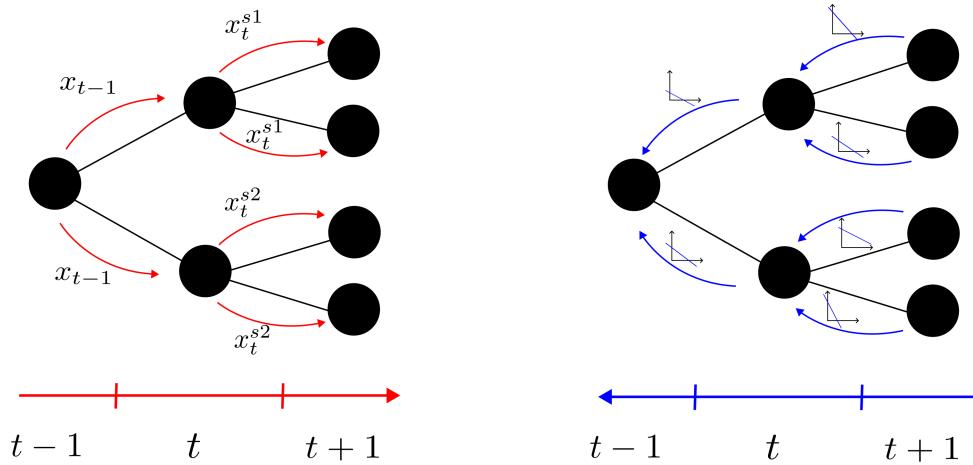


Figura 2 – Passo *forward* e *backward* da DDP

cortes responsáveis por aproximar inferiormente a função de recurso com base no ponto consultado no passo *Forward*. Os cortes gerados são então adicionados ao conjunto de cortes construídos nas iterações anteriores.

Ao final da iteração, tem-se uma aproximação da função de recurso melhor, ou igual, do que a aproximação da iteração anterior, e com isso se obtém o *limite inferior* ou *zinf* para a iteração seguinte. Dados os limites superior e inferior, é estabelecida uma métrica para determinar se o algoritmo convergiu ou não. Uma vez que a solução ótima do problema necessariamente é menor que o limite superior e maior que o limite inferior, define-se o “*gap* de convergência”:

$$gap = \frac{zsup - zinf}{zinf} \quad (4.1)$$

Frequentemente, o valor do *gap* é expresso em termos percentuais $gap(\%) = 100 \cdot gap$. O valor do $gap(\%)$ representa a diferença percentual máxima entre a solução obtida em uma iteração e a solução ótima do problema. Para garantir comportamento monótono decrescente do valor do *gap* é necessário considerar como *zsup* o menor custo das soluções obtidas em todas as iterações anteriores ($minzsup_i$). Dessa forma, ao final de cada iteração, a convergência do algoritmo é avaliada verificando se o valor do *gap* é inferior à tolerância estabelecida, a qual define a distância admissível entre a solução encontrada e a solução ótima.

Os métodos DDP e SDDP têm como premissa a convexidade dos subproblemas (ex.: PL), característica que garante que a função de recurso do subproblema seja convexa. Outra premissa necessária para a convergência é em relação aos cortes construídos, que devem ser:

- válidos: um corte é válido se, e somente se, para todos os pontos do domínio das variáveis de recurso, o valor do corte é inferior ou igual ao valor da função de recurso no ponto.
- exato (*tight*): um corte é exato se, para o ponto em que foi construído, o valor do corte é igual ao valor da função de recurso.

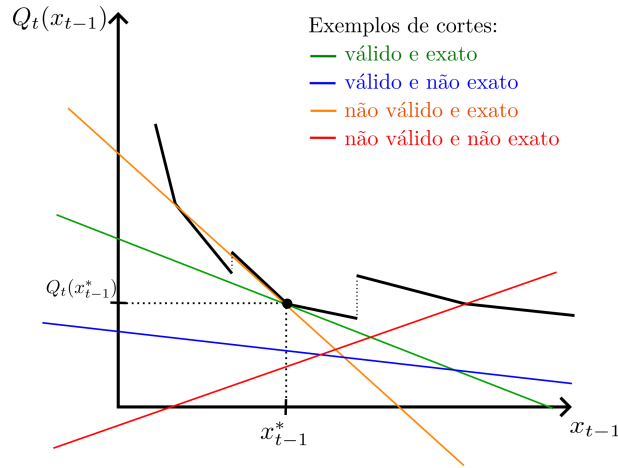


Figura 3 – Exemplo de cortes quanto à validade e à exatidão

A Figura 3 ilustra uma função de recurso unidimensional $Q_t(x_{t-1})$, representada pelo traço preto, a qual é não convexa e não contínua. A figura também apresenta exemplos de cortes construídos no ponto x_{t-1}^* , classificados de acordo com sua validade e exatidão. O corte verde corresponde a um corte válido e exato, enquanto o corte laranja representa um corte exato, porém não válido. O corte azul, por sua vez, é válido, mas não exato, e o corte vermelho não é nem válido nem exato.

As premissas de convexidade da função e dualidade forte, assumidas pelos métodos DDP e SDDP, garantem que cortes lineares obtidos sejam válidos e exatos ao construir aproximações afins de primeira ordem baseadas na solução dual associada ao ponto consultado.

No entanto, quando o subproblema envolve variáveis inteiras (ex.: MILP), esta função de recurso deixa de ser convexa, conforme ilustrado pela função $Q_t(x_{t-1})$ representada na Figura 3. Assim, em algumas regiões do domínio não é possível obter um corte linear que seja válido e exato. Neste caso, esses cortes lineares deixam de garantir a convergência dos métodos DDP e SDDP e, conforme discutido na seção anterior, uma alternativa seria obter cortes não lineares ou aplicar técnicas como a DDiP/SDDiP.

No caso da DDiP e SDDiP, a premissa de que as variáveis de recurso são binárias garante que a função de recurso possa ser representada por um poliedro convexo, mesmo que o subproblema não seja convexo. Dessa forma, é possível obter cortes lineares válidos e exatos e que garantam a convergência do método. No entanto, para subproblemas não convexos, como a obtenção das soluções duais não é direta, algumas metodologias para construção destes cortes serão detalhadas ao longo deste capítulo.

4.2 Famílias de cortes para a DDiP e SDDiP

Considerando a função de recurso do período t e cenário s dada pelas Equações (4.2)-(4.3).

$$Q_{t,s}(x_{t-1,s^{(a)}}) = \min_{x_{t,s}, y_{t,s}} c_{t,s}^T \cdot [x_{t,s}, y_{t,s}] + \sum_{s^{(p)} \in \mathcal{S}_{t,s}} p_{s^{(p)}|s} Q_{t+1,s^{(p)}}(x_{t,s}) \quad (4.2)$$

$$s.t. (x_{t,s}, y_{t,s}) \in \mathcal{V}(x_{t-1,s^{(a)}}) \quad (4.3)$$

Onde:

$x_{t-1,s^{(a)}} \in \{0, 1\}^N$: representa o vetor das N variáveis de recurso, ou seja, decisão tomada no período $t - 1$ no cenário $s^{(a)}$, correspondente ao subproblema antecessor na árvore de cenários.

$x_{t,s} \in \{0, 1\}^N$: representa o vetor das N variáveis de decisão no subproblema do período t , cenário s que representam variáveis de recurso para o período $t + 1$.

$y_{t,s} \in \mathbb{R}^M \times \mathbb{Z}^P$: são as variáveis de decisão no subproblema do período t , cenário s que não representam variáveis de recurso. Correspondem a um vetor de M variáveis contínuas e P variáveis inteiras.

$c \in \mathbb{R}^{N+M+P}$: é o vetor de custo associado às variáveis de decisão.

$\mathcal{V}(x_{t-1,s^{(a)}}) \subset \{0, 1\}^N \times \mathbb{R}^M \times \mathbb{Z}^P$: representa o conjunto de soluções viáveis para as variáveis de decisão dado um valor para as variáveis de recurso $x_{t-1,s^{(a)}}$.

$\mathcal{S}_{t,s}$: representa o conjunto de cenários no estágio $t + 1$ que podem se realizar dada a realização do cenário s no período t .

$p_{s^{(p)}|s}$: representa a probabilidade condicionada da realização do cenário $s^{(p)}$ em $t + 1$ dada a realização do cenário s em t .

O processo de aproximação iterativa da SDDiP e DDiP pode ser descrito através da substituição das funções de recurso do estágio posterior ($t + 1$), cuja representação exata é desconhecida, por um conjunto de restrições lineares (cortes), inseridas a cada iteração k do processo. Assim, a aproximação da função de recurso na iteração K fica então representada da seguinte forma:

$$Q_{t,s}^K(x_{t-1,s^{(a)}}) = \min_{x_{t,s}, y_{t,s}, \alpha_{t,s}} c_{t,s}^T \cdot [x_{t,s}, y_{t,s}] + \sum_{s^{(p)} \in \mathcal{S}_{t,s}} p_{s^{(p)}|s} \alpha_{t,s} \quad (4.4)$$

$$s.t. (x_{t,s}, y_{t,s}) \in \mathcal{V}(x_{t-1,s^{(a)}}) \quad (4.5)$$

$$\alpha_{t,s} \geq rhs_{k,t,s} - \lambda_{k,t,s}^T \cdot x_{t,s}, \forall k = 1, K \quad (4.6)$$

Onde:

$\alpha_{t,s} \in \mathbb{R}$: é uma variável de decisão que representa o custo futuro do período t e cenário s .

$rhs_{k,t,s} \in \mathbb{R}$: representa o termo independente do k -ésimo corte que compõe o modelo linear por partes da função de recurso.

$\lambda_{k,t,s} \in \mathbb{R}^N$: representa o vetor de coeficientes do k -ésimo corte que compõe o modelo linear por partes da função de recurso.

A determinação de um corte, dado pelos coeficientes rhs e λ , é feita através da resolução do subproblema no passo *backward* e pode ser feita de várias formas, conforme descrito na sequência.

4.3 Cortes de Benders

Os cortes de Benders [62] são largamente utilizados em inúmeras aplicações da DDP e SDDP devido à sua simplicidade e eficiência. Cada corte é construído a partir do valor da função objetivo e das soluções duais associadas ao subproblema.

No caso da DDiP e SDDiP, onde cada subproblema pode ser um MILP, a obtenção de soluções duais não é direta. Assim, os cortes são construídos com a solução objetivo e soluções duais do MILP resolvido com as restrições de integralidade relaxadas. Os cortes construídos desta forma são relaxados em relação à função de custo futuro real. Assim, esses cortes são válidos, porém não exatos (vide Figura 3, corte azul) e, conseqüentemente, a convergência da DDiP ou SDDiP com cortes de Benders não é garantida.

O corte construído por um subproblema em uma iteração tem a seguinte forma:

$$rhs = z_{LR}^* + \mu_{LR}^{*T} x_{t-1,s}^*(a)$$

$$\lambda = \mu_{LR}^*$$

Onde:

$z_{LR}^* \in \mathbb{R}$: é a solução da função objetivo (custo total) do subproblema linearmente relaxado.

$x_{t-1,s}^*(a)$: representa o vetor de valores para as variáveis de recurso da iteração (decididos no passo *forward*).

$\mu_{LR}^* \in \mathbb{R}^N$: é o vetor de soluções duais (ou composição das soluções duais) do problema relaxado, relativo às restrições onde as variáveis de recurso estão presentes.

O principal aspecto em se utilizar os cortes de Benders para a DDiP e SDDiP é que esses cortes tendem a ser mais eficazes quanto mais próximo o problema inteiro misto for do problema linearmente relaxado. Ou seja, como a função de recurso do problema linearmente relaxado é uma aproximação inferior e convexa da função de recurso do MILP, quanto mais próximas estas duas funções forem, no ponto de otimalidade, mais perto do ótimo ficará a solução obtida utilizando Benders. Outra observação importante é que este corte pode ser construído mesmo que a variável de recurso não seja binária.

4.4 Cortes Inteiros de Otimalidade (Integer Optimality Cuts)

As características do corte inteiro de otimalidade [48] são as seguintes: tangencia a função no ponto onde foi construído ($x_{t-1,s(a)}^*$), consultado no passo *forward*, sendo exato nesse ponto, e possui uma inclinação válida. Como as inclinações não contêm informações de primeira ordem, os cortes não refletem adequadamente a geometria local da função de recurso e podem apresentar sinais incorretos sobre as direções de decrescimento. Consequentemente, observa-se um processo de convergência lento, por vezes quase enumerativo, entre as combinações de soluções inteiras.

O corte construído tem a seguinte forma:

$$rhs = z_{MILP} \cdot (1 - 1^\top x_{t-1,s(a)}^*)$$

$$\lambda = z_{MILP} \cdot (2x_{t-1,s(a)}^* - 1)$$

Onde:

$z_{MILP} \in \mathbb{R}$: é a solução da função objetivo (custo total) do subproblema, ou seja, $z_{MILP} = Q_{t,s}^K(x_{t-1,s(a)}^*)$

A expressão apresentada para o corte foi descrita de forma simplificada assumindo que a função $Q_{t,s}$ possui um limite inferior de 0, a forma geral pode ser encontrada em [48].

4.5 Cortes de Lagrange

Os cortes de Lagrange [50], assim como os cortes Inteiros de Otimalidade, são exatos e válidos, o que garante a convergência do método. No entanto, este método se baseia em soluções duais para o cálculo das inclinações, o que melhora a convergência da SDDiP em relação aos cortes Inteiros de Otimalidade.

Para garantir que os cortes de Lagrange sejam válidos e exatos, é necessária a criação de uma variável de cópia $z \in [0, 1]^N$, bem como uma restrição de cópia que atrela seu valor ao valor do recurso. Além disso, a variável de cópia é utilizada no restante da formulação para representar a variável de recurso. Assim, para execução da DDiP ou SDDiP utilizando os cortes de Lagrange, o subproblema $Q_{t,s}^K(x_{t-1,s(a)})$ é reformulado como $Q_{t,s}^K(x_{t-1,s(a)}, z)$, e sua formulação é dada por:

$$\mathcal{Q}_{t,s}^K(x_{t-1,s^{(a)}}) = \min_{x_{t,s}, y_{t,s}, z_{t,s}, \alpha_{t,s}} c_{t,s}^T \cdot [x_{t,s}, y_{t,s}] + \sum_{s^{(p)} \in \mathcal{S}_{t,s}} p_{s^{(p)}|s} \alpha_{t,s} \quad (4.7)$$

$$s.t. (x_{t,s}, y_{t,s}) \in \mathcal{V}(z_{t,s}) \quad (4.8)$$

$$\alpha_{t,s} > rhs_{k,t,s} - \lambda_{k,t,s}^T \cdot x_{t,s}, \forall k = 1, K \quad (4.9)$$

$$z_{t,s} = x_{t-1,s^{(a)}} \quad (4.10)$$

Este problema alterado possui a mesma solução do problema original e permite a relaxação da restrição (4.10), o que é necessário para garantir as características dos cortes de Lagrange. Assim, define-se o problema de Lagrange como o problema $\mathcal{Q}_{t,s}^K(x_{t-1,s^{(a)}})$ com a restrição (4.10) relaxada e penalizada na função objetivo pelos fatores $\mu_{t,s} \in \mathbb{R}^N$, ou seja:

$$L_{t,s}^K(\mu_{t,s}, x_{t-1,s^{(a)}}) = \min_{x_{t,s}, y_{t,s}, z_{t,s}, \alpha_{t,s}} c_{t,s}^T \cdot [x_{t,s}, y_{t,s}] + \sum_{s^{(p)} \in \mathcal{S}_{t,s}} p_{s^{(p)}|s} \alpha_{t,s} + \mu_{t,s}^T \cdot (z_{t,s} - x_{t-1,s^{(a)}}) \quad (4.11)$$

$$s.t. (x_{t,s}, y_{t,s}) \in \mathcal{V}(z_{t,s}) \quad (4.12)$$

$$\alpha_{t,s} > rhs_{k,t,s} - \lambda_{k,t,s}^T \cdot x_{t,s}, \forall k = 1, K \quad (4.13)$$

Para dado ponto $x_{t-1,s^{(a)}}^*$, decidido no passo *forward*, os valores duais ótimos $\mu_{t,s}^*$ são dados pela solução primal da maximização do problema de Lagrange, ou seja:

$$z_D = \max_{\mu_{t,s}} L_{t,s}(\mu_{t,s}, x_{t-1,s^{(a)}}^*) \quad (4.14)$$

Através desta formulação é provado em [50] que há propriedade de dualidade forte, ou seja:

$$L_{t,s}^K(\mu_{t,s}^*, x_{t-1,s^{(a)}}^*) = \mathcal{Q}_{t,s}^K(x_{t-1,s^{(a)}}^*)$$

Assim, constrói-se o corte para a DDiP ou SDDiP através das expressões (4.15) e (4.16):

$$rhs = \mathcal{Q}_{t,s}^K(x_{t-1,s^{(a)}}^*) - \mu_{t,s}^{*T} \cdot x_{t-1,s^{(a)}}^* \quad (4.15)$$

$$\lambda = \mu_{t,s}^* \quad (4.16)$$

Dessa forma, para calcular os cortes de Lagrange é necessário determinar as soluções duais $\mu_{t,s}^*$, associadas às restrições (4.10). Como este problema é um MILP, a obtenção destes duais não é trivial, sendo necessário resolver o problema (4.14). A determinação dos duais será explorada no Capítulo 6.

4.6 Cortes de Benders Reforçados (Strengthened Benders Cuts)

Uma versão aprimorada dos cortes de Benders é proposta em [50]. Esses cortes, denominados cortes de Benders reforçados, possuem a mesma inclinação dos cortes de Benders tradicionais, porém são deslocados verticalmente para cima, de modo a serem menos relaxados que os originais. Ainda assim, esses cortes são válidos, mas não exatos. A forma geral desses cortes é dada por:

$$rhs = L_{t,s}(\mu_{LR}^*) + \mu_{LR}^{*T} x_{t-1,s}^{*(a)}$$

$$\lambda = \mu_{LR}^*$$

Essa solução é necessariamente maior ou igual ao custo total do problema relaxado, produzindo um corte paralelo e mais próximo da função real.

4.7 Aplicação da DDiP e da SDDiP no problema de planejamento energético

Ao decompor o problema de planejamento energético, obtém-se como variável de recurso o volume armazenado nos reservatórios, uma vez que o volume inicial de um dado estágio de tempo deve ser igual ao volume final do estágio de tempo anterior. Os volumes dos reservatórios são naturalmente grandezas contínuas. Dessa forma, para garantir a convergência da DDiP ou SDDiP é preciso transformar os volumes em um conjunto de variáveis binárias, o que pode ser feito através de uma expansão binária.

4.7.1 Representação do volume por meio de uma expansão binária

O volume final de um reservatório, $vf \in \mathbb{R}$, pode ser aproximado como uma expansão binária de $N \in \mathbb{N}$ variáveis binárias $b_n \in \{0, 1\}, n = 0, N-1$, tal que:

$$vf = \frac{\overline{vf}}{2^N - 1} \sum_{n=0}^{N-1} 2^n \cdot b_n \quad (4.17)$$

Onde $\overline{vf} \in \mathbb{R}$ é o valor máximo da variável vf (volume máximo do reservatório da usina em questão). Define-se $\varepsilon = \frac{\overline{vf}}{2^N - 1}$ sendo a precisão (passo de discretização) ao representar o volume, que pode variar entre 0 e $\overline{vf}$, com N variáveis binárias.

Por exemplo, considerando $\overline{vf} = 90$, com $N = 1$, haverá apenas 2^1 valores de volume final e a expressão binária será:

$$vf = 90 \cdot b_0$$

Ou seja, quando $b_0 = 0 \Rightarrow vf = 0$ e quando $b_0 = 1 \Rightarrow vf = 90$, e o passo de discretização é $\varepsilon = 90$. No caso de $N = 2$, existem 2^2 valores de volume final tal que:

$$vf = \frac{90}{3} \cdot b_0 + \frac{90}{3} \cdot 2 \cdot b_1$$

Nesta situação, vf pode assumir os seguintes valores:

- $b_0 = 0$ e $b_1 = 0 \Rightarrow vf = 0$
- $b_0 = 1$ e $b_1 = 0 \Rightarrow vf = 30$
- $b_0 = 0$ e $b_1 = 1 \Rightarrow vf = 60$
- $b_0 = 1$ e $b_1 = 1 \Rightarrow vf = 90$

Ou seja, a precisão $\varepsilon = \frac{\overline{vf}}{2^N - 1} = 30$ para este caso. Dessa forma, quanto maior N , melhor a representação do volume como uma variável contínua. No entanto, ao aumentar o valor de N o número de variáveis binárias cresce, assim como a complexidade computacional do problema.

Para representar um volume de uma usina que possa variar de 0 a $\overline{vf}$ com uma dada precisão ε é necessário fazer o cálculo inverso:

$$\varepsilon = \frac{\overline{vf}}{2^N - 1}$$

$$2^N - 1 = \frac{\overline{vf}}{\varepsilon}$$

$$2^N = 1 + \frac{\overline{vf}}{\varepsilon}$$

Como N é uma variável inteira, procura-se o menor valor de N que garanta pelo menos a precisão desejada:

$$N = \lceil \log_2 \left(1 + \frac{\overline{vf}}{\varepsilon} \right) \rceil \quad (4.18)$$

Ou seja, suponhamos que uma usina hidrelétrica tenha um volume útil total de $\overline{vf} = 1000 \text{ hm}^3$ e deseja-se representá-lo com uma precisão de $\varepsilon = 1 \text{ hm}^3$, são necessárias $N = 10$ variáveis binárias. Mas, se para este mesmo volume for desejada uma precisão de $\varepsilon = 0.1 \text{ hm}^3$, seriam necessárias $N = 14$ variáveis binárias.

4.7.2 Impacto da “binarização” da variável de recurso

A introdução das variáveis binárias e restrições de expansão binárias (4.17) permite que as variáveis contínuas de volume sejam transformadas em variáveis de decisão enquanto as

variáveis binárias se tornam as variáveis de recurso. Ou seja, o vetor de variáveis de recurso, $x_{t,s} \in \mathbb{R}^H$, inicialmente composto pelas variáveis de volumes finais das H usinas do estudo:

$$x_{t,s} = [vf_{t,s}^1, vf_{t,s}^2, \dots, vf_{t,s}^H],$$

passa a ser um vetor das $N = \sum_{h=1}^H N_h$ variáveis binárias, onde N_h corresponde ao número de variáveis binárias usado na expansão binária da usina h :

$$x_{t,s} = [b_{1t,s}^1, \dots, b_{N_{1t,s}}^1, b_{1t,s}^2, \dots, b_{N_{2t,s}}^2, \dots, b_{1t,s}^H, \dots, b_{N_{Ht,s}}^H],$$

Isso pode corresponder a um aumento expressivo da dimensão da variável de recurso $x_{t,s}$, que passa de dimensão H para dimensão $N = \sum_{h=1}^H N_h$. A título de exemplo, para representar os reservatórios com precisão de 10%, ou seja, um ponto a cada 10% de seu volume máximo, pela Equação (4.18), são necessárias 4 variáveis binárias por usina, o que representa um aumento de 4 vezes na dimensão deste vetor. Para representar com uma precisão de 1% são necessárias 7 variáveis binárias.

Em problemas de grande porte, onde esta dimensão é elevada, por exemplo, no sistema brasileiro são representados tipicamente 67 reservatórios, tem-se:

Variável contínua	Expansão binária (precisão 10%)	Expansão binária (precisão 1%)
67	268	469

Tabela 2 – Dimensão do espaço da função de recurso

O aumento da dimensão da variável de recurso faz com que a função de recurso se torne mais complexa de ser aproximada, necessitando de uma quantidade maior de cortes lineares e, conseqüentemente, iterações da SDDiP/DDiP. Além disso, a inserção destas variáveis e da restrição de expansão binária torna o subproblema bastante mais complexo de ser resolvido.

Dessa forma, embora a utilização de variáveis de recurso binárias permita a aplicação de métodos exatos e que fazem uso de cortes lineares, como a SDDiP e DDiP, também traz um ônus quando, para o problema em questão, as variáveis de recurso não são naturalmente binárias. Nos capítulos seguintes, esse impacto e a sensibilidade desses métodos em relação ao número de variáveis binárias utilizadas serão explorados por meio de experimentos computacionais.

5 Estudos de caso em pequena escala

O objetivo desta seção é avaliar de forma detalhada, por meio de estudos de caso reduzidos, os seguintes aspectos:

- a resposta e o desempenho do método DDiP e dos algoritmos descritos no Capítulo 4;
- o efeito das restrições condicionadas apresentadas na Seção 2.4 sobre a função de recurso;
- os impactos da representação da variável de recurso contínua por meio de uma expansão binária.

Optou-se por avaliar a DDiP em vez da SDDiP, pois assim é possível observar de forma isolada o impacto das sensibilidades avaliadas na convergência do método, uma vez que há convergência determinística quando se resolve a árvore de cenários completa.

Neste contexto, ao longo da seção é adotada uma nomenclatura padronizada para a definição de métodos e representações, conforme descrito a seguir. No que se refere à representação da variável de recurso, esta pode ser realizada de duas formas:

- **“continuous-storage”**: indica que as variáveis de recurso (volume inicial) e de decisão (volume final) são representadas por variáveis contínuas, correspondendo à sua representação natural. Nesse caso, quando o problema é resolvido por decomposição, a função de recurso depende de uma ou mais variáveis contínuas.
- **“binary-storage”**: indica que as variáveis de recurso (volume inicial) e de decisão (volume final) são representadas por variáveis binárias. Para isso, são utilizadas expressões de expansão binária que convertem os volumes em variáveis binárias, conforme descrito na Seção 4.7.1, sendo necessário especificar o número N de variáveis binárias consideradas. Nesse caso, quando o problema é resolvido por decomposição, a função de recurso passa a depender exclusivamente de variáveis binárias.

Os estudos de caso avaliados nesta seção consideram três variantes distintas quanto à inclusão de restrições operativas:

- **“no-constr”**: corresponde ao caso base, contendo apenas as variáveis e restrições necessárias à representação do problema de planejamento energético, sem a inclusão de restrições operativas hidráulicas condicionais;
- **“cond-constr”**: corresponde ao caso base acrescido das restrições operativas condicionais pertinentes a cada estudo de caso, cuja modelagem utilizada está descrita na Seção 2.5;

- **“with-constr”**: corresponde ao caso base acrescido das mesmas restrições operativas, porém consideradas de forma incondicional, isto é, aplicáveis independentemente do estado do sistema.

Com relação aos métodos empregados para a resolução dos estudos de caso, são considerados:

- Global, referidos como **“MILP-global”** ou **“LP-global”**, que correspondem à solução direta do problema por meio de um único MILP ou LP, no qual todos os nós da árvore de cenários são agregados em um único problema de otimização, resolvido de forma centralizada. Quando a representação adotada é “binary-storage” e/ou “cond-constr”, o problema assume natureza inteira mista (MILP); caso contrário, trata-se de um problema linear (LP);
- **DDP**, aplicado quando o problema é resolvido por decomposição e a representação da variável de recurso é do tipo “continuous-storage”. Neste caso, podem ser utilizados cortes de Benders clássicos ou Reforçados;
- **DDiP**, aplicado quando o problema é resolvido por decomposição e a representação da variável de recurso é do tipo “binary-storage”. Neste caso, podem ser empregados cortes de Benders clássicos, Benders Reforçados, cortes Inteiros de Otimalidade e cortes de Lagrange.

Os estudos de caso em pequena escala apresentam as características gerais descritas no Capítulo 2. Foram elaborados dois estudos de caso:

- Estudo de caso **“tutorial”** (Seção 5.1): caso determinístico com apenas uma usina hidráulica, desenvolvido com o objetivo de permitir uma análise mais aprofundada e detalhada dos resultados;
- Estudo de caso **“reduced-system”** (Seção 5.2): representa uma versão reduzida do sistema elétrico brasileiro, de natureza estocástica, com uma árvore de cenários igualmente reduzida.

Todos os experimentos foram conduzidos em computador equipado com processador Intel i7 de 11a geração com 8 núcleos e 32 GB de memória RAM. A implementação foi realizada em C++, utilizando o solver CPLEX da IBM, com multiprocessamento habilitado por meio de duas *threads*. O método adotado para a solução dos problemas inteiros mistos foi o *branch-and-bound*, enquanto os problemas lineares foram resolvidos por meio do método *simplex*.

5.1 Estudo de caso “tutorial”

O estudo de caso “tutorial” representa um problema de planejamento energético com dimensões reduzidas ao mínimo. Seu principal objetivo é servir como um ambiente controlado para validação metodológica e análise detalhada do comportamento dos métodos considerados. Especificamente, este estudo visa:

- verificar a implementação computacional;
- analisar de forma detalhada o comportamento de cada um dos métodos de solução, considerando:
 - a capacidade de alcançar o ótimo global do problema;
 - a taxa de convergência em relação aos demais métodos;
 - o impacto da representação de uma variável de recurso contínua por meio de expansão binária;
 - o efeito da inserção de restrições condicionais envolvendo variáveis inteiras na DDiP (solução exata) e na DDP (solução não exata).

Adicionalmente, avalia-se o impacto da introdução de restrições hidráulicas condicionais sobre as características da função de recurso associada a um subproblema de planejamento energético.

5.1.1 Características gerais do caso “tutorial” base

O caso “tutorial” segue a formulação geral de planejamento energético apresentada no Capítulo 2, porém com dimensões significativamente reduzidas. As principais características do problema estão resumidas na Tabela 3.

Os custos variáveis unitários (CVU) e os limites de geração das usinas termelétricas são apresentados na Tabela 4. A demanda energética por patamar de carga e período é descrita na Tabela 5. Cada subproblema, correspondente a um período, considera as restrições listadas na Tabela 6.

Número de hidrelétricas com reservatórios	1
Número de termelétricas	5
Número de submercados de energia	1
Número de patamares de carga	3
Número de períodos	2
Número de cenários por período	1

Tabela 3 – Dimensões do problema “tutorial”

A formulação completa do problema tutorial é apresentada no Apêndice A.

Código da usina	CVU (R\$/MWh)	Potência Instalada (MW)
1	31.17	8
4	310.41	2
9	978.10	2
12	1700.00	2
991	6524.05	999

Tabela 4 – Características das usinas térmicas no problema “tutorial”

Período / Patamar	1	2	3
1	25	19	15
2	45	39	35

Tabela 5 – Demanda energética em MW no problema “tutorial”

Restrição de balanço hidráulico	1
Restrições de atendimento à demanda	3
Restrições de função de produção hidráulica	18

Tabela 6 – Restrições do subproblema de cada período do caso “tutorial”

5.1.2 Representação do volume armazenado no caso “tutorial”

A usina hidrelétrica considerada no estudo “tutorial” é a usina real de Camargos, pertencente ao sistema elétrico brasileiro. Essa usina possui potência instalada de 46 MW e reservatório de regularização com volume útil máximo de 672 hm³.

A variável de decisão associada ao volume armazenado foi representada de duas formas:

- **“continuous-storage”**: representação por meio de uma variável contínua;
- **“binary-storage”**: representação por meio de expansão binária, considerando $N = 23$ e $N = 5$ variáveis binárias, conforme ilustrado na Figura 4.

5.1.3 Representação de restrições hidráulicas no caso “tutorial”

São consideradas três variantes do caso “tutorial” no que se refere à modelagem de restrições hidráulicas. Seja $q_{p,t}$ a vazão defluente da usina hidrelétrica no período t e no patamar p . Então:

- **“no-constr”**: corresponde ao caso base, no qual nenhuma restrição hidráulica adicional é incorporada ao modelo:

$$q_{p,t} \leq \infty, \forall t \in \{1, 2\}, p \in \{1, 2, 3\}; \quad (5.1)$$

VF_U_001 =			
0.00008010	*	BIN_VOLF_U_001_I_00	
0.00016021	*	BIN_VOLF_U_001_I_01	
0.00032043	*	BIN_VOLF_U_001_I_02	
0.00064086	*	BIN_VOLF_U_001_I_03	
0.00128173	*	BIN_VOLF_U_001_I_04	
0.00256347	*	BIN_VOLF_U_001_I_05	
0.00512695	*	BIN_VOLF_U_001_I_06	
0.01025390	*	BIN_VOLF_U_001_I_07	
0.02050781	*	BIN_VOLF_U_001_I_08	
0.04101562	*	BIN_VOLF_U_001_I_09	
0.08203125	*	BIN_VOLF_U_001_I_10	
0.16406251	*	BIN_VOLF_U_001_I_11	
0.32812503	*	BIN_VOLF_U_001_I_12	
0.65625007	*	BIN_VOLF_U_001_I_13	
1.31250015	*	BIN_VOLF_U_001_I_14	
2.62500031	*	BIN_VOLF_U_001_I_15	
5.25000062	*	BIN_VOLF_U_001_I_16	
10.5000012	*	BIN_VOLF_U_001_I_17	
21.0000025	*	BIN_VOLF_U_001_I_18	
42.0000050	*	BIN_VOLF_U_001_I_19	
84.0000100	*	BIN_VOLF_U_001_I_20	
168.000020	*	BIN_VOLF_U_001_I_21	
336.000040	*	BIN_VOLF_U_001_I_22	

VF_U_001 =			
21.6774	*	BIN_VOLF_U_001_I_00	
43.3548	*	BIN_VOLF_U_001_I_01	
86.7096	*	BIN_VOLF_U_001_I_02	
173.4193	*	BIN_VOLF_U_001_I_03	
346.8387	*	BIN_VOLF_U_001_I_04	

Figura 4 – Representação do volume da usina de Camargos utilizando 23 variáveis binárias (esquerda) e 5 variáveis binárias (direita)

- **“with-constr”**: nesta variante, é adicionada ao problema original uma restrição hidráulica incondicional, dada por:

$$q_{p,t} \leq 200, \forall t \in \{1, 2\}, p \in \{1, 2, 3\}. \quad (5.2)$$

Essa restrição impõe um limite máximo de defluência válido para todos os períodos e patamares, independentemente do estado do sistema.

- **“cond-constr”**: nesta variante, é introduzida uma restrição hidráulica condicional, inspirada em restrições operativas reais do planejamento energético. A restrição limita a vazão defluente máxima em função do volume inicial do reservatório:

$$q_{p,t} \leq 200, \text{ se } vi_t < 55\%, \forall t \in \{1, 2\}, p \in \{1, 2, 3\}. \quad (5.3)$$

Nessa equação, vi_t denota o volume inicial da usina no período t . Assim, a restrição de defluência máxima é aplicada apenas quando o volume inicial do reservatório é inferior a 55% do volume útil máximo, evitando elevados níveis de defluência em situações de baixo armazenamento. Essa restrição foi incorporada ao modelo por meio da introdução de duas variáveis binárias adicionais, conforme descrito na Seção 2.5.

5.1.4 Solução global do problema “tutorial”

Devido às reduzidas dimensões do problema “tutorial”, é possível resolvê-lo diretamente como um único problema de otimização, na forma de um LP ou MILP, agregando-se todas as

variáveis e restrições em um modelo global. Essa solução, denominada LP-global ou MILP-global, é exata e serve como referência para validação dos resultados obtidos posteriormente por meio dos métodos de decomposição.

O custo total da solução global para cada variante do problema é apresentado na Tabela 7, onde se pode observar que:

Custo total da solução (R\$)			
	no-constr	with-constr	cond-constr
continuous-storage	1950765.1284*	2248289.3900*	1952445.9155
binary-storage (23)	1950765.1392	2248289.4348	1952445.9200
binary-storage (5)	1952035.0724	2542048.7022	2488007.9306

Tabela 7 – Custo total do caso “tutorial” obtido pela solução LP/MILP-global. Células marcadas com * indicam que o problema é um LP, por não conter variáveis inteiras.

- Os custos obtidos com “continuous-storage” e “binary-storage” (23) são bastante próximos. Isso indica que a expansão binária com 23 variáveis fornece uma aproximação suficientemente precisa da variável contínua, produzindo apenas um aumento marginal no custo em função da discretização.
- No caso “binary-storage” (5), o custo ótimo obtido é significativamente superior aos demais casos, refletindo a limitação imposta pela baixa resolução na representação do volume armazenado.
- Comparando-se as diferentes variantes de restrições, observa-se que o problema com restrição incondicional apresenta o maior custo, como esperado. O caso com restrição condicional resulta em um custo intermediário entre os casos sem restrição e com restrição incondicional.

No caso sem restrições hidráulicas adicionais (“no-constr”), a operação ótima caracteriza-se pelo enchimento do reservatório no primeiro período, seguido de um aumento do turbinamento no segundo período, no qual a demanda é mais elevada. Essa estratégia é ilustrada na Tabela 8. Observa-se ainda que o volume armazenado é integralmente consumido ao final do horizonte de planejamento, uma vez que não há função de custo futuro associada ao armazenamento no final do horizonte.

Ao introduzir a restrição hidráulica — seja de forma condicional ou incondicional — essa solução deixa de ser viável. No segundo período, a usina atinge um turbinamento de $206.81 \text{ m}^3/\text{s}$, superior ao limite de $200 \text{ m}^3/\text{s}$, e o volume inicial do período é de 54.34%, inferior ao limiar de 55% estabelecido pela restrição. Esse fato explica o aumento de custo observado na Tabela 7.

Período	Patamar	Volume Inicial (%)	Volume Final (%)	Turbinamento (m^3/s)	Vertimento (m^3/s)
1	1	25.54	54.34	69.12	0.0
	2			50.30	0.0
	3			26.60	0.0
2	1	54.34	0.00	206.81	0.0
	2			189.36	0.0
	3			155.97	0.0

Tabela 8 – Operação ótima do reservatório no problema “tutorial-no-constr” com “binary-storage” (23)

No caso “tutorial-with-constr”, a defluência é limitada a $200 m^3/s$ independentemente do volume. Como consequência, no primeiro patamar do segundo período torna-se necessário acionar termelétricas de maior custo. Para mitigar esse efeito, a solução ótima passa a aumentar o volume armazenado, inicial e final, do segundo período, de modo a elevar a produtividade hidráulica e, conseqüentemente, a geração ao se utilizar a defluência máxima permitida neste patamar. A Tabela 9 apresenta a operação correspondente.

Período	Patamar	Volume Inicial (%)	Volume Final (%)	Turbinamento (m^3/s)	Vertimento (m^3/s)
1	1	25.54	56.07	69.01	0.0
	2			41.16	0.0
	3			22.87	0.0
2	1	56.07	4.09	200.00	0.0
	2			183.71	0.0
	3			150.32	0.0

Tabela 9 – Operação ótima do reservatório no problema “tutorial-with-constr” com “binary-storage” (23)

Quando a restrição é condicional (“tutorial-cond-constr”), o modelo passa a explorar estratégias de operação mais flexíveis. Em particular, a otimização pode optar por manter o volume inicial do segundo período acima de 55%, permitindo turbinamentos superiores a $200 m^3/s$, ou aceitar a limitação da defluência. Dessa forma, o problema admite duas estratégias operativas viáveis. A Tabela 10 mostra a operação ótima obtida nesse caso.

Observa-se que a introdução da restrição condicionada resulta em uma maior retenção no armazenamento no primeiro estágio, elevando o volume inicial no segundo estágio para 55%. Esse ajuste permite que a defluência da usina ultrapasse $200 m^3/s$. Em outras palavras, o custo do primeiro estágio aumenta ligeiramente em comparação ao problema original (sem restrição), mas esse aumento de custo contribui para reduzir o custo do segundo estágio, em relação ao custo observado em um problema com a restrição incondicional.

No que se refere à representação com baixa resolução do volume armazenado (5 variáveis binárias), constata-se que os volumes ótimos obtidos com a formulação contínua não podem ser

Período	Patamar	Volume Inicial (%)	Volume Final (%)	Turbinamento (m^3/s)	Vertimento (m^3/s)
1	1	25.54	55.00	69.08	0.0
	2			50.30	0.0
	3			23.12	0.0
2	1	55.00	0.00	206.81	0.0
	2			188.72	0.0
	3			155.56	0.0

Tabela 10 – Operação ótima do reservatório no problema “tutorial-cond-constr” com “binary-storage” (23)

exatamente reproduzidos. A Tabela 11 apresenta os valores discretizados disponíveis próximos ao ótimo contínuo, enquanto a Tabela 12 compara os volumes finais obtidos com representação contínua e com expansão binária de 5 variáveis.

Representação binária	Volume correspondente (%)
$b = (1, 0, 0, 0, 0)$	51.61
$b = (1, 0, 0, 0, 1)$	54.84
$b = (1, 0, 0, 1, 0)$	58.06

Tabela 11 – Valores discretizados do volume armazenado com 5 variáveis binárias

Caso “tutorial”	variável contínua	expansão com 5 binárias
no-constr	54.34	54.84
cond-constr	55.00	58.06
with-constr	56.07	54.84

Tabela 12 – Volumes úteis finais (%) ao final do período 1

Conclui-se que, nos casos “no-constr” e “with-constr”, os volumes ótimos contínuos (54.34% e 56.07%) são aproximados por 54.84%, valor discretizado mais próximo dos ótimos contínuos. Já no caso “cond-constr”, o modelo escolhe o valor 58.06%, pois este é o menor valor discretizado que satisfaz a condição necessária para desativar a restrição no segundo período. Esses resultados evidenciam a importância de uma modelagem cuidadosa de restrições condicionais e mostram que expansões binárias de baixa resolução podem introduzir distorções relevantes na representação de variáveis contínuas, levando a decisões sub-ótimas em relação ao modelo original.

Os resultados detalhados da operação para o problema resolvido de forma global podem ser encontrados nos Apêndices B.1.1 e B.2.1.

5.1.5 Análise da função de recurso do problema “tutorial”

Com o objetivo de analisar de forma mais detalhada os efeitos das restrições condicionais sobre a função de recurso, foi realizada uma avaliação sistemática dessa função no segundo estágio do problema “tutorial”. Para isso, o subproblema do segundo estágio foi resolvido 672 vezes, variando-se o volume útil inicial desde o valor mínimo (0 hm^3) até o valor máximo admissível (672 hm^3). Para cada valor de volume, foi computado o custo total da operação do segundo estágio, resultando na função de recurso correspondente.

Nessa análise, foram consideradas três variantes da restrição hidráulica, conforme descrito na Seção 5.1.3:

- **no-constr**: sem restrição à defluência, conforme Equação (5.1);
- **with-constr**: com restrição hidráulica incondicional, conforme Equação (5.2);
- **cond-constr**: com restrição hidráulica condicional, sendo analisadas duas formas de condicionamento:
 - restrição condicionada ao volume inicial (**v_init**), utilizada na solução do problema “tutorial”, conforme a Equação (5.3);
 - restrição condicionada ao volume médio no período (**v_mean**), na qual o critério de ativação da restrição depende da média entre o volume inicial e final do período:

$$q_{t,p} \leq 200, \text{ se } \frac{v_{i_t} + v_{f_t}}{2} < 55\%, \forall t \in \{1, 2\}, p \in \{1, 2, 3\}. \quad (5.4)$$

A Figura 5 apresenta as quatro funções de recurso obtidas em função do volume inicial. Observa-se que, devido à escala adotada, as curvas apresentam comportamento visualmente semelhante ao longo de grande parte do domínio. Por esse motivo, a Figura 6 apresenta ampliações em duas regiões específicas: à esquerda, em torno de 55% do volume útil máximo, e à direita, em torno de 82%.

Nessas regiões, torna-se evidente que a função de custo do problema sem restrição hidráulica (curva verde) é inferior à do caso com restrição incondicional (curva amarela), conforme esperado. A função de custo associada à restrição condicionada ao volume inicial, **v_init** (curva azul), coincide, para volumes inferiores ao limiar de ativação, com a função do caso com restrição incondicional. No entanto, a partir do momento em que o volume inicial atinge o limiar da restrição (55%), essa função passa a coincidir com a função do problema sem restrição. Esse comportamento evidencia que a função de recurso associada à restrição hidráulica condicional ao volume inicial apresenta uma estrutura não convexa, mais precisamente, convexa por partes, com uma descontinuidade no ponto de transição entre os regimes de ativação e desativação da restrição.

No caso em que a restrição é condicionada ao volume médio do período, v_mean (curva vermelha), essa transição ocorre de forma mais gradual e em valores mais elevados de volume inicial, situando-se aproximadamente entre 78% e 86%. Esse comportamento decorre do fato de que o relaxamento da restrição passa a depender também do volume final do subproblema, que é uma variável de decisão e pode ser ajustada pela otimização, suavizando o efeito da condição sobre a defluência máxima.

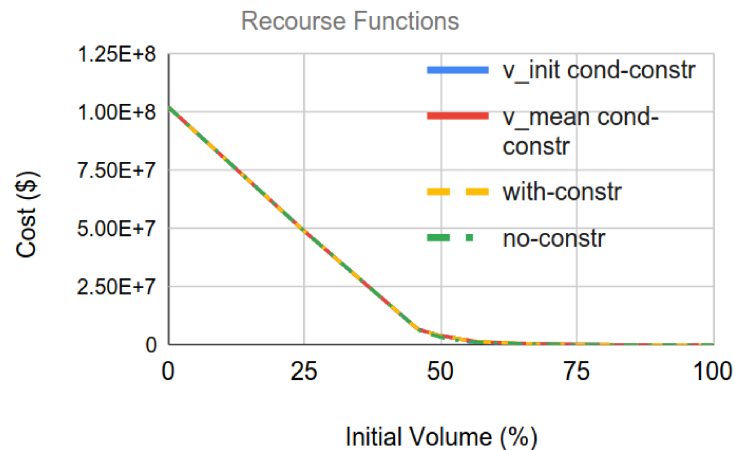


Figura 5 – Função de recurso do segundo estágio do caso “tutorial”, avaliada em 672 valores do volume inicial entre os limites mínimo e máximo

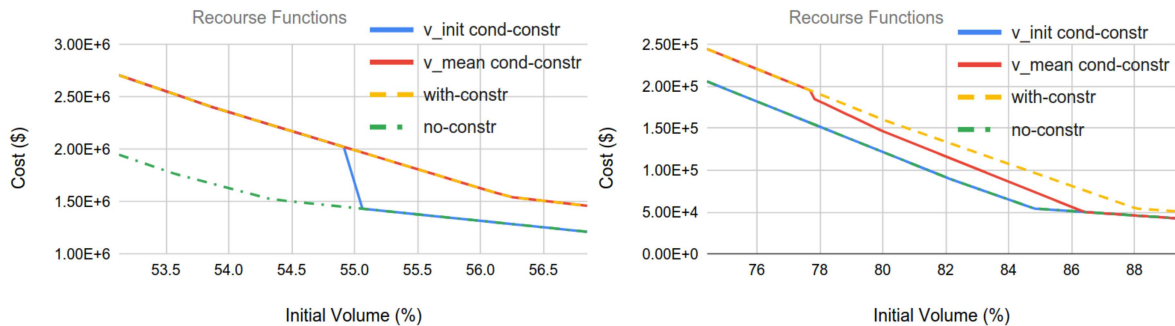


Figura 6 – Ampliação nas regiões de descontinuidade da função de recurso do segundo estágio do caso “tutorial”

Embora as diferenças entre as funções de recurso nos quatro casos analisados sejam relativamente pequenas e concentradas em regiões específicas do domínio, destaca-se que esse tipo de restrição não convexa tende a ocorrer simultaneamente em diversos reservatórios e em diferentes regiões do sistema. Nesse contexto, é plausível que o sistema opere com frequência nessas regiões, o que pode levar a uma avaliação inadequada do valor da água e dos custos futuros de operação nessas condições.

Em síntese, a incorporação de restrições condicionais nos subproblemas de uma árvore de cenários em programação estocástica altera significativamente a estrutura da função de recurso, transformando-a de uma função convexa (poliedral) em uma função semicontínua inferior e

convexa por partes. Além da maior dificuldade na representação das não convexidades e descon- tinuidades da função de recurso, essas alterações aumentam a complexidade computacional do problema, frequentemente resultando em formulações do tipo MILP mais difíceis de resolver. Os resultados observados evidenciam, portanto, o compromisso inerente entre maior fidelidade na modelagem das restrições operativas e a viabilidade computacional dos métodos de solução empregados.

5.1.6 Solução do problema “tutorial” utilizando cortes de Benders

A Tabela 13 apresenta um resumo da operação obtida pelo método de Benders para o caso “cond-constr” (valores em vermelho), comparativamente à solução ótima do problema “tutorial” sem restrições condicionais (valores em azul).

Período	Patamar	Volume Inicial (%)	Volume Final (%)	Defluência ou Turbinamento (m^3/s)	Custo no estágio (R\$)
1	1	25.54	54.34 (54.34)	69.12 (69.12)	429095.63 (429095.63)
	2			41.16 (41.16)	
	3			32.00 (32.00)	
2	1	54.34 (54.34)	5.88 (0.00)	200.00 (206.81)	2231487.47 (1521604.66)
	2			188.51 (189.36)	
	3			155.12 (155.97)	

Tabela 13 – Resultados da operação do reservatório no problema “tutorial-cond-constr” com o método de Benders (em vermelho), comparativamente à solução ótima do problema “tutorial” sem a restrição condicional (em azul)

A análise desses resultados permite extrair conclusões importantes a respeito do uso de cortes de Benders em problemas com restrições condicionais:

- Embora a restrição condicional esteja, em princípio, presente tanto no primeiro quanto no segundo estágio (uma vez que os volumes iniciais são inferiores a 55%), a solução do primeiro estágio para o caso “cond-constr” coincide exatamente com a solução do problema “no-constr”. Isso ocorre porque:
 - as vazões no primeiro estágio são baixas, de modo que a restrição, embora presente, não fica ativa;
 - no passo *backward*, os cortes são construídos a partir do problema relaxado, no qual as variáveis inteiras podem assumir valores fracionários. Nessas condições, a restrição condicional não influencia a formação dos cortes, fazendo com que todos os cortes gerados do segundo para o primeiro estágio sejam idênticos aos do problema “no-constr”.

- Como consequência, o volume inicial do segundo estágio permanece em 54.34%, que é ótimo para o problema irrestrito. Contudo, no passo *forward*, a restrição condicional é considerada, impedindo a usina de defluir acima de $200\text{m}^3/\text{s}$, o que resulta em uma operação significativamente mais onerosa no segundo estágio.

Ao observar o processo de convergência, para o caso “no-constr”, no qual o segundo estágio é convexo, uma vez que não há expansão binária no último estágio, a solução é obtida com *gap* de convergência nulo na quinta iteração, mostrando que o método atinge o valor ótimo. De fato, custo de operação encontrado com a DDP-Benders, tanto na representação “continuous-storage” quanto “binary-storage”, coincidem exatamente com aqueles obtidos pelas soluções LP-global e MILP-global, respectivamente. Esses resultados são apresentados nas Tabelas 14 e 15.

No entanto, para o caso “cond-constr”, tanto na representação “continuous-storage” quanto na “binary-storage”, observa-se que o método não converge para a solução ótima. O processo iterativo se estabiliza a partir da quinta iteração, com valores constantes de z_{inf} e z_{sup} , resultando em um *gap* de aproximadamente 36% em ambos os casos. Os custos totais, valores de *gap* e número de iterações são apresentados nas Tabelas 16 e 17.

Nesta variante, custo da operação (R\$2660647.55010) é superior ao ótimo global do problema com restrição condicional (MILP-global: R\$1952445.9200670), e também superior ao custo do problema com restrição incondicional (R\$2248289.4348740). Ou seja, quando a restrição existe incondicionalmente, a solução ótima é elevar o volume inicial no segundo estágio até 56.07%, de forma a reduzir os custos no segundo estágio. Entretanto, quando aplicado Benders, o passo *backward* ignora a restrição condicionada na formação do corte e o primeiro estágio não antecipa corretamente o impacto dessa restrição sobre os custos futuros.

O processo de convergência completo das execuções do “tutorial” utilizando os cortes de Benders pode ser encontrado nos Apêndices B.1.2 e B.2.2.

5.1.7 Solução do problema “tutorial” utilizando cortes Inteiros de Otimalidade

Os cortes Inteiros de Otimalidade, descritos na Seção 4.4, apresentam convergência relativamente lenta, conforme discutido em [49]. Esse comportamento torna-se ainda mais crítico à medida que se aumenta a dimensão da função de recurso. Ao resolver o caso “tutorial” utilizando a representação do volume com 23 variáveis binárias, observou-se que, mesmo após 500 iterações da DDiP, o processo de convergência ainda apresentava um *gap* elevado, indicando que a solução estava distante do ótimo. Diante disso, para viabilizar a análise da convergência dessa metodologia, optou-se por utilizar o caso em que a variável de volume é representada por apenas 5 variáveis binárias.

Nessas condições, para o caso “no-constr”, foi possível obter a solução ótima via DDiP

em 19 iterações. Observa-se que os custos das soluções obtidas com 5 variáveis binárias, tanto pela abordagem MILP-global quanto pela DDiP, coincidem exatamente, evidenciando a convergência adequada do método. Esses resultados encontram-se sintetizados na Tabela 15.

De forma similar, para o caso “cond-constr”, a solução ótima também foi alcançada após 19 iterações da DDiP. O custo total obtido é idêntico ao custo resultante da resolução do problema via MILP-global com 5 variáveis binárias, conforme apresentado na Tabela 17.

Com relação à operação ótima, os resultados foram muito semelhantes àqueles obtidos pela solução global, com pequenas diferenças atribuídas à existência de soluções alternativas de mesmo valor ótimo. Os detalhes da operação e o histórico completo do processo de convergência das execuções do caso “tutorial” utilizando cortes Inteiros de Otimalidade estão disponíveis nos Apêndices B.1.4 e B.2.4.

5.1.8 Solução do problema “tutorial” utilizando cortes de Lagrange

O método baseado em cortes de Lagrange, descrito na Seção 4.5, apresentou convergência para a solução ótima em ambos os casos analisados. Especificamente, a convergência foi atingida após 5 iterações no caso “no-constr” e após 26 iterações no caso “cond-constr”. Em ambos os cenários, o valor da função objetivo coincide exatamente com aquele obtido pela resolução do problema via MILP-global, conforme apresentado nas Tabelas 15 e 17.

Em todas as execuções, foi adotada a representação mais detalhada da variável de volume, utilizando 23 variáveis binárias. No que se refere à operação ótima, os resultados obtidos com os cortes de Lagrange são idênticos àqueles encontrados pela abordagem MILP-global. O detalhamento completo do processo de convergência e dos resultados operativos pode ser consultado nos Apêndices B.1.5 e B.2.5.

5.1.9 Solução do problema “tutorial” utilizando cortes de Benders Reforçados

A metodologia de cortes de Benders Reforçados, descrita na Seção 4.6, pode ser aplicada ao problema “tutorial” tanto com a representação “continuous-storage” (DDP) quanto com a representação “binary-storage” (DDiP) da variável de recurso.

No caso “no-constr”, para ambas as representações, o subproblema do segundo estágio não contém variáveis inteiras, caracterizando-se como um problema convexo. Nessas condições, os cortes gerados pelo método de Benders Reforçados são idênticos, salvo diferenças numéricas, aos cortes de Benders tradicionais. Isso ocorre porque as soluções duais do problema relaxado linearmente coincidem com as soluções do problema original, uma vez que não há distinção entre as duas formulações. Consequentemente, os valores do custo total e os resultados operativos obtidos são exatamente os mesmos daqueles gerados pelo método de Benders tradicional.

Para o caso “cond-constr”, por sua vez, diferentemente do observado com os cortes de Benders tradicionais, os cortes de Benders Reforçados foram capazes de alcançar a solução ótima em ambas as representações. Ressalta-se, entretanto, que a convergência desse método não é garantida em termos gerais, e o bom desempenho observado pode estar associado às dimensões muito reduzidas do problema “tutorial”.

Os custos totais obtidos, assim como detalhes do processo iterativo dos dois casos, nas duas representações, estão nas Tabelas 14, 15, 16 e 17. Com relação aos resultados da operação, foram similares aos métodos globais, apenas com diferenças numéricas em poucos valores. Mais detalhes sobre o processo de convergência podem ser encontrados nos Apêndices B.1.3 e B.2.3.

5.1.10 Resumo dos resultados do caso “tutorial-no-constr”

As Tabelas 14 e 15 resumem os resultados encontrados para o problema “tutorial-no-constr” nas representações “continuous-storage” e “binary-storage”, respectivamente, em cada uma das metodologias exploradas.

	Custo da solução (R\$)	<i>Gap</i> (%)	Número de iterações
LP-global	1950765.12849	-	-
Benders	1950765.12849	0.000	5
Benders Reforçado	1950765.12849	0.000	5

Tabela 14 – Resumo dos resultados do problema “tutorial-no-constr” com representação “continuous-storage” para as metodologias exploradas.

A partir destes resultados, conclui-se que:

- Todos os métodos apresentaram convergência para o valor ótimo, com *gap* de convergência nulo, à exceção do método de Cortes Inteiros de Otimalidade. Nesse caso, não foi possível observar a convergência no problema original, sendo o processo iterativo interrompido após 500 iterações devido ao elevado tempo computacional.
- Esse método, no entanto, apresentou convergência adequada quando o número de variáveis binárias no problema foi reduzido de 23 para 5.
- Os resultados operativos obtidos foram semelhantes entre todos os métodos analisados.

5.1.11 Resumo dos resultados do caso “tutorial-cond-constr”

As Tabelas 16 e 17 resumem os resultados encontrados para o problema “tutorial-cond-constr” para cada uma das metodologias exploradas.

	Custo da solução (R\$)	Gap (%)	Número de iterações
MILP-global (23bin)	1950765.1392	-	-
Benders (23bin)	1950765.1392	0.000	5
Benders Reforçado (23bin)	1950765.1392	0.000	5
Inteiros de Otimalidade (23bin)	37129974.4547	122345638.542	500*
Cortes de Lagrange (23bin)	1950765.1392	0.000	5
MILP-global (5bin)	1952035.0724	-	-
Inteiros de Otimalidade (5bin)	1952035.0724	0.000	19

Tabela 15 – Resumo dos resultados do problema “tutorial-no-constr” com a representação “binary-storage” para as metodologias avaliadas. “*” indica que o processo iterativo da DDiP foi interrompido devido ao tempo ou número de iterações.

	Custo da solução (R\$)	Gap (%)	Número de iterações
LP-global	-	-	-
Benders	2660647.2855	36.389	5
Benders Reforçado	1952445.9155	0.000	4

Tabela 16 – Resumo dos resultados do problema “tutorial-cond-constr” com representação “continuous-storage” para as metodologias exploradas.

A partir destes resultados, conclui-se que:

- o método DDP ou DDiP com cortes de Benders é incapaz de convergir para a solução ótima nesse caso, apresentando estagnação do processo iterativo após poucas iterações.
- Por sua vez, os cortes de Benders Reforçados, aplicados tanto na DDP quanto na DDiP, mostraram-se capazes de obter a solução ótima. No entanto, é importante ressaltar que, embora tenham sido eficazes na prática para este estudo de caso, esses cortes não fornecem garantia teórica de convergência.
- O algoritmo DDiP com cortes de Lagrange também se mostrou eficaz na obtenção da solução ótima. Ainda assim, o número de iterações necessário para a convergência foi relativamente maior quando comparado aos demais métodos analisados.
- A metodologia baseada na construção de cortes Inteiros de Otimalidade apresentou bons

	Custo da solução (R\$)	Gap (%)	Número de iterações
MILP-global (23bin)	1952445.9200	-	-
Benders (23bin)	2660646.2860	36.389	5
Benders Reforçado (23bin)	1952445.9200	0.000	5
Inteiros de Otimalidade (23bin)	*	*	*
Cortes de Lagrange (23bin)	1952445.9200	0.000	26
MILP-global (5bin)	2488007.9306	-	-
Inteiros de Otimalidade (5bin)	2488007.9306	0.000	19

Tabela 17 – Resumo dos resultados do problema “tutorial-cond-constr” com a representação “binary-storage” para as metodologias avaliadas. “*” indica que o processo iterativo da DDiP foi interrompido devido ao tempo ou número de iterações

resultados, porém mostrou-se limitada a problemas de menor complexidade, caracterizados por um número reduzido de variáveis binárias.

5.2 Estudo de caso “reduced-system”

O estudo de caso “reduced-system” representa um problema de planejamento energético com todos os principais aspectos, porém com dimensões reduzidas. Essa redução visa tornar viável a aplicação e a comparação sistemática das metodologias de solução analisadas. Os principais objetivos da análise deste caso são:

- obter resultados para todos, ou para a maior parte, dos métodos avaliados, o que motivou a adoção de um problema de menor porte;
- analisar a robustez da resposta de cada metodologia, considerando:
 - o tempo computacional;
 - a eficiência na obtenção do ótimo global do problema;
 - a taxa de convergência, em comparação com os demais métodos.

5.2.1 Características gerais do caso “reduced-system”

As principais características do problema “reduced-system” são apresentadas na Tabela 18. Os custos incrementais e limites de geração das usinas térmicas são definidos conforme a Tabela 19. A demanda energética é considerada em dois submercados interligados, sendo especificada para cada patamar de carga e período na Tabela 20. Cada subproblema, correspondente a um nó da árvore de cenários, contempla o conjunto de restrições descrito na Tabela 21.

Total de usinas hidráulicas	14
Número de usinas hidráulicas com reservatórios	4
Número de usinas térmicas	7
Número de submercados de energia	2
Número de reservatórios equivalentes de energia	3
Número de patamares de carga	3
Número de períodos	3
Número de cenários por período	2
Total de nós na árvore de cenários	7

Tabela 18 – Dimensões do problema “reduced-system”

As usinas consideradas neste estudo de caso são usinas reais do sistema elétrico brasileiro e foram representadas de forma semelhante à utilizada pelo Operador Nacional do Sistema (ONS) no planejamento da operação. Essas usinas estão organizadas em três cascatas distintas, sendo cada cascata associada a um Reservatório Equivalente de Energia (REE).

A Figura 7 apresenta um diagrama esquemático da organização das usinas consideradas. Os REEs são utilizados no acoplamento com a Função de Custo Futuro (FCF), responsável

Código da usina	CVU (R\$/MWh)	Potência Instalada (MW)
1	29.13	657
2	653.43	131
3	550.00	584
4	310.41	220
50	470.34	200
991	999999.00	9999999
992	999999.00	9999999

Tabela 19 – Características das usinas térmicas no problema “reduced-system”

Período	1			2			3		
Subm. \ Pat.	1	2	3	1	2	3	1	2	3
1	2861	2791	2175	2861	3191	2575	3261	3191	2575
2	4121	4492	3337	4521	4992	3837	4521	4892	3737

Tabela 20 – Demanda energética em MW no problema “reduced-system” para cada período, patamar e submercado

Restrição de balanço hidráulico	14
Restrições de atendimento à demanda	6
Restrições de função de produção hidráulica	237
Restrições da FCF multivariada por REE (presente nos subproblemas do último estágio)	600

Tabela 21 – Descrição das restrições do subproblema do caso “reduced-system”

pela valoração do armazenamento ao final do horizonte de planejamento. Essa valoração é representada por uma função linear por partes da energia armazenada em cada REE, obtida a partir da execução de um modelo de horizonte temporal mais longo.

5.2.2 Representação do volume armazenado no caso “reduced-system”

As variáveis de decisão associadas aos volumes armazenados nos reservatórios no caso “reduced-system” podem ser representadas de duas formas: “continuous-storage” e “binary-storage”, de maneira análoga à adotada no estudo de caso “tutorial”. Quando a representação binária é utilizada, introduz-se uma expressão de expansão binária para a determinação do volume armazenado em cada usina.

Nesse caso, a quantidade de variáveis binárias associadas a cada reservatório é definida de modo a garantir uma precisão mínima de discretização do volume de 0.0001 hm^3 . A Tabela 22 apresenta o volume útil disponível para regularização de cada um dos quatro reservatórios considerados no caso “reduced-system”, bem como o número de variáveis binárias utilizado na respectiva expansão binária. Neste caso, a expansão binária descrita na tabela será referida como “binary-storage” (28).

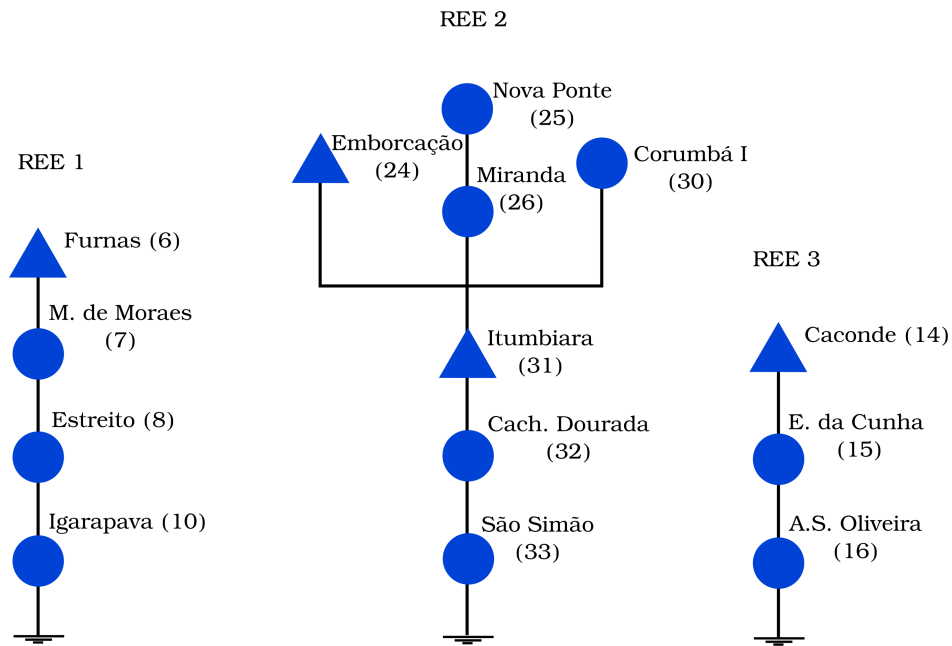


Figura 7 – Organização das usinas representadas no caso “reduced-system”. Os triângulos representam usinas com capacidade de regularização, enquanto os círculos representam usinas a fio d’água.

Usina	Volume útil (hm^3)	Número de variáveis binárias
Furnas	17217	28
Emborcação	13056	27
Itumbiara	12454	27
Caconde	504	23

Tabela 22 – Volume útil disponível para regularização nas usinas com reservatório do caso “reduced-system” e o número de variáveis binárias utilizadas na representação do volume

Com essa configuração, são introduzidas, no total, 105 variáveis de recurso associadas aos volumes armazenados.

5.2.3 Representação das restrições hidráulicas no caso “reduced-system”

Na variante “cond-constr”, foram introduzidas três restrições hidráulicas condicionais associadas ao volume inicial dos reservatórios. Essas restrições são aplicadas a todos os períodos do horizonte de planejamento e estão definidas conforme descrito a seguir:

- Usina de Furnas:

$$q_{p,t} \leq 800 m^3/s, \text{ se } v_{i_t} < 65\%, \forall t \in \{1, 2, 3\}, p \in \{1, 2, 3\}; \quad (5.5)$$

- Usina de Caconde:

$$q_{p,t} \leq 80 m^3/s, \text{ se } v_{i_t} < 54\%, \forall t \in \{1, 2, 3\}, p \in \{1, 2, 3\}; \quad (5.6)$$

- Usina de Itumbiara:

$$q_{p,t} \leq 900m^3/s, \text{ se } vi_t < 62\%, \forall t \in \{1, 2, 3\}, p \in \{1, 2, 3\}; \quad (5.7)$$

Nessas expressões, $q_{p,t}$ representa a vazão defluente da respectiva usina no patamar de carga p e período t , enquanto vi_t denota o volume inicial do reservatório no mesmo período. Os valores dos limites máximos de defluência e dos limiares percentuais de volume foram calibrados de modo a garantir que as restrições se tornassem ativas ao longo do horizonte de planejamento, possibilitando, assim, a análise dos impactos decorrentes da introdução dessas restrições condicionais.

5.2.4 Solução global do problema “reduced-system”

As dimensões do problema “reduced-system” ainda permitem a obtenção da solução exata através da resolução de um único problema de otimização, na forma LP-global ou MILP-global. As soluções globais obtidas para as duas variantes do problema, correspondentes à ausência e à presença de restrições hidráulicas condicionais (“no-constr” e “cond-constr”), são apresentadas, respectivamente, nas Tabelas 23 e 24.

“reduced-system-no-constr”		
	Custo da solução (R\$)	Tempo (s)
continuous-storage	3367632219.2374	0.40
binary-storage (28)	3367632222.0325	50.68

Tabela 23 – Custo total e tempo de processamento da solução com LP-global/MILP-global do caso “reduced-system-no-constr”

“reduced-system-cond-constr”		
	Custo da solução (R\$)	Tempo (s)
continuous-storage	3368929947.0213	1.03
binary-storage (28)	3368929950.7895	6381.78

Tabela 24 – Custo total e tempo de processamento da solução com MILP-global do caso “reduced-system-cond-constr”

A partir destes resultados, observa-se que:

- Os custos ótimos obtidos com as representações “continuous-storage” e “binary-storage (28)” são muito próximos, refletindo o elevado nível de fidelidade da expansão binária detalhada.
- A introdução das restrições hidráulicas condicionais aumenta o custo total do problema em relação ao caso sem restrições, embora essa diferença não seja expressiva em termos absolutos.

- Em termos de tempo computacional, observa-se um aumento significativo da complexidade quando as restrições condicionais são consideradas, especialmente na representação “binary-storage (28)”, na qual o tempo de solução cresce de forma substancial.

Esse aumento desproporcional do tempo computacional está diretamente associado à maior complexidade inerente à resolução de problemas MILP. Na execução “cond-constr-continuous-storage”, o modelo inclui apenas as variáveis binárias associadas às restrições condicionais, totalizando 126 variáveis binárias no problema global. Já na execução “cond-constr-binary-storage”, além dessas mesmas 126 variáveis binárias, estão presentes as variáveis da expansão binária associadas ao volume dos reservatórios, que totalizam $28 + 27 + 27 + 23 = 105$ variáveis por nó da árvore de cenários, exceto nos nós do último período, resultando em 315 variáveis adicionais. Assim, o MILP global passa a conter, no total, 441 variáveis binárias.

Além do aumento expressivo no número de variáveis inteiras, a estrutura das restrições de expansão binária concentra diversas variáveis em um único conjunto de restrições, o que tende a dificultar o processo de enumeração do algoritmo *branch-and-bound*, conforme discutido em [60].

5.2.5 Resultados “reduced-system-no-constr” resolvidos por decomposição

A Tabela 25 apresenta os resultados obtidos para o problema “reduced-system-no-constr” utilizando a representação “continuous-storage”, na qual a metodologia de solução corresponde ao algoritmo tradicional da DDP. Nessa representação, apenas os cortes de Benders e de Benders Reforçados podem ser empregados. Para fins de comparação, o resultado da solução global também é apresentado na tabela. Observa-se que o problema é convexo e que ambas as metodologias são capazes de encontrar a solução ótima. A utilização de cortes de Benders Reforçados demanda um esforço computacional ligeiramente maior em relação aos cortes de Benders tradicionais ou à solução LP-global; entretanto, o tempo absoluto de resolução é bastante reduzido, tornando essa diferença pouco relevante. Para uma avaliação mais representativa do desempenho computacional dessas metodologias, seria necessário aplicá-las a problemas de maior porte.

	Custo da solução (R\$)	Gap (%)	Número de iterações	Tempo (s)
LP-global	3367632219.2374	-	-	0.40
Benders (DDP)	3367632219.2373	0.000	5	0.50
Benders Reforçado (DDP)	3367632219.2373	0.000	6	1.08

Tabela 25 – Resultados do problema “reduced-system-no-constr” com representação “continuous-storage” para as metodologias apresentadas.

A Tabela 26 apresenta os resultados obtidos para o problema “reduced-system-no-constr” utilizando a representação “binary-storage”, na qual a resolução é realizada por meio da DDiP.

	Custo da solução (R\$)	Gap (%)	Núm. de iterações	Tempo (s)
MILP-global	3367632222.0325	-	-	50.68
Benders	3367632222.3635	0.000	7	22.08
Benders Reforçado	3367632222.2313	0.000	6	53.65
Inteiros de Otimalidade	3549384123.3647*	480550036.16*	578*	7200*
Cortes de Lagrange	3367632222.6040	0.000	7	30.39

Tabela 26 – Resultados do problema “reduced-system-no-constr” com a representação “binary-storage” para as metodologias apresentadas. “*” indica que o processo iterativo da DDiP foi interrompido antes de atingir a convergência

A partir destes resultados, conclui-se que:

- Os cortes Inteiros de Otimalidade não foram capazes de atingir a solução ótima, mesmo após longos tempos de execução (cerca de duas horas) e um grande número de iterações, o *gap* de convergência permaneceu elevado, indicando que essa abordagem não é adequada para o problema em questão, restringindo-se a modelos de menor complexidade, com número reduzido de variáveis de recurso.
- Observou-se convergência adequada ao se utilizar cortes de Benders, Benders Reforçados e de Lagrange. Para os cortes de Benders e de Lagrange, o tempo computacional foi inferior ao tempo requerido pela solução MILP-global, evidenciando as vantagens do uso da decomposição. Esse benefício tende a se tornar ainda mais relevante à medida que o número de cenários do problema aumenta.
- Embora a convergência dos cortes de Benders e de Benders Reforçados não seja teoricamente garantida no contexto da DDiP, dois fatores podem ter contribuído para o bom desempenho observado:
 - Os subproblemas do terceiro período, em todos os quatro cenários, são convexos, pois não possuem estágios futuros e as variáveis de volume final não são representadas por variáveis binárias. Dessa forma, os cortes gerados do terceiro para o segundo estágio são válidos e exatos.
 - O subproblema do segundo estágio é não convexo devido à presença de variáveis binárias associadas ao volume final, o que resulta em cortes de Benders potencialmente frouxos. No entanto, como a discretização do volume é bastante próxima da representação contínua, a relaxação linear utilizada na construção dos cortes fornece uma

aproximação bastante precisa, fazendo com que os cortes obtidos estejam próximos dos cortes exatos e permitindo a convergência da DDiP.

O histórico completo do processo de convergência das metodologias para o caso “reduced-system-no-constr” pode ser consultado no Apêndice C.1.

5.2.6 Resultados “reduced-system-cond-constr” resolvidos por decomposição

A Tabela 27 apresenta os resultados obtidos para o problema “reduced-system-cond-constr” utilizando a representação “continuous-storage”, cuja resolução é realizada por meio do algoritmo tradicional da DDP. Nessa representação, apenas cortes de Benders e de Benders Reforçados podem ser aplicados. Entretanto, para o caso em questão, a presença de restrições condicionais introduz não convexidades nos subproblemas, de modo que tais cortes deixam de garantir convergência.

Observa-se que não foi possível atingir *gap* nulo em nenhuma das duas abordagens. Além disso, os resultados obtidos com Benders e Benders Reforçados foram praticamente idênticos. A partir da quinta iteração, os valores de *zinf* e *zsup* permanecem estagnados, indicando que o procedimento iterativo atingiu o limite da aproximação fornecida pelos cortes gerados.

	Custo da solução (R\$)	Gap (%)	Número de iterações	Tempo (s)
MILP-global	3368929947.0213	-	-	1.03
Benders	3371415773.3374	0.112	5	1.85
Benders Reforçado	3371415773.3374	0.112	5	3.85

Tabela 27 – Resultados do problema “reduced-system-cond-constr” com representação “continuous-storage” para as metodologias aplicadas.

A Tabela 28 apresenta os resultados para o mesmo problema utilizando a representação “binary-storage”, em que a solução é obtida por meio da DDiP.

O processo de convergência completo das metodologias para o caso “reduced-system-cond-constr” pode ser encontrado no Apêndice C.2. A partir destes resultados, conclui-se que:

- Assim como na representação “continuous-storage”, não foi possível atingir a solução ótima com cortes de Benders e de Benders Reforçados na representação “binary-storage”. Além disso, não se observou melhoria com o uso da representação binária em relação à representação contínua neste caso.
- Com cortes de Lagrange não foi possível observar convergência dentro do limite de tempo considerado. Apesar de a convergência dos cortes de Lagrange ser garantida teoricamente,

	Custo da solução (R\$)	Gap (%)	Núm. de iterações	Tempo (s)
MILP-global	3368929950.7895	-	-	6381.78
Benders	3371415782.6081	0.112	6	15.81
Benders Reforçado	3371415794.7158	0.112	6	48.57
Inteiros de Otimalidade	*	*	*	*
Cortes de Lagrange	3371963611.7895*	0.170*	97*	27800*

Tabela 28 – Resultados do problema “reduced-system-cond-constr” com a representação “binary-storage” para as metodologias avaliadas. “*” indica que o processo iterativo da DDiP foi interrompido antes de atingir a convergência

constatou-se uma taxa de convergência muito lenta, e a execução foi interrompida após aproximadamente 7 horas, totalizando 97 iterações. Isso motivou a análise de sensibilidade da DDiP com cortes de Lagrange variando a quantidade de variáveis binárias realizada na próxima seção.

- Os cortes Inteiros de Otimalidade, assim como os cortes de Lagrange, não foram capazes de alcançar convergência em um tempo/número de iterações razoável.

5.2.6.1 Análise de sensibilidade da DDiP com expansão binária

Com o objetivo de avaliar o desempenho da DDiP sob diferentes níveis de precisão na expansão binária, foi considerado o caso “reduced-system” com a variante “cond-constr”. O problema foi resolvido utilizando a DDiP com 7, 6 e 5 variáveis binárias para representar o volume armazenado em cada um dos quatro reservatórios. Com isso, foi possível representar, respectivamente, 128, 64 e 32 níveis discretos de volume por reservatório.

A Figura 8 ilustra o processo de convergência para cada uma dessas configurações, apresentando a evolução do limite inferior (*LB*) e do limite superior (*UB*) ao longo das iterações, juntamente com o custo ótimo obtido pela resolução do problema completo por meio do MILP-global (representado pela linha preta). O limite superior corresponde ao melhor custo encontrado até a iteração corrente, enquanto a convergência é atingida quando o limite inferior se aproxima suficientemente do limite superior.

A Tabela 29 resume o número de iterações, o tempo computacional total e o custo obtido ao final do processo iterativo para cada nível de discretização. A partir desses resultados, podem ser feitas as seguintes observações:

- À medida que o número de variáveis binárias utilizadas na expansão do volume aumenta, o custo ótimo obtido pela DDiP diminui e se aproxima do custo ótimo associado à

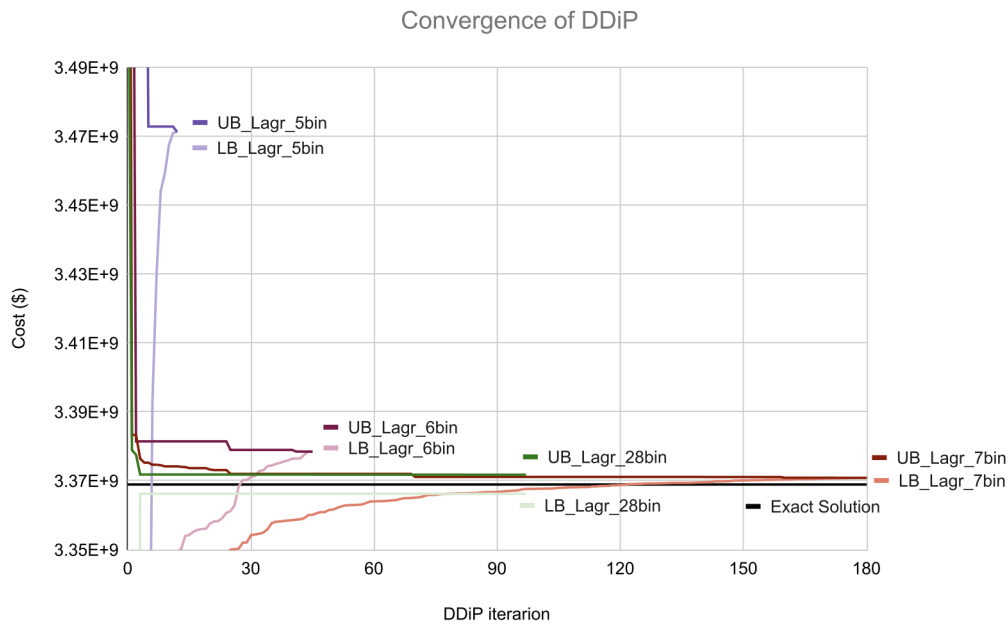


Figura 8 – Limite superior, z_{sup} (UB) e inferior, z_{inf} (LB) a cada iteração da DDiP.

Método	Iterações	Tempo (s)	Custo ótimo (\$)
MILP-global	-	-	3368929946.91
Lagr_28bin	97*	27800*	3371963611.79*
Lagr_7bin	180	2993	3370855102.14
Lagr_6bin	45	556	3378488276.07
Lagr_5bin	12	97	3471309152.11

Tabela 29 – Resultados de convergência da DDiP com cortes de Lagrange com diferentes precisões na expansão binária, * processo de convergência interrompido devido ao excesso de tempo

representação contínua dos reservatórios.

- Por outro lado, tanto o número de iterações quanto o tempo computacional total aumentam significativamente, apresentando um crescimento acentuado conforme se eleva a quantidade de variáveis binárias.
- Embora a DDiP seja capaz de convergir para soluções viáveis sob as discretizações adotadas, a obtenção do valor ótimo exato exigiria um número elevado de variáveis binárias, o que se mostrou computacionalmente ineficiente.

Em síntese, a obtenção de uma solução exata por meio da DDiP com cortes de Lagrange requer uma discretização muito fina da variável de recurso, implicando um número elevado de variáveis binárias e, consequentemente, um custo computacional proibitivo, mesmo para problemas de porte relativamente reduzido. Nesse contexto, soluções inexatas passam a desempenhar um papel relevante, como aquelas obtidas com cortes de Benders e de Benders Reforçados.

Por fim, à luz dos resultados e análises apresentados neste capítulo, observam-se algumas considerações gerais. No uso de cortes não exatos (como Benders e Benders Reforçados) em problemas não convexos, verificou-se que existe potencial para a obtenção de soluções de boa qualidade; contudo, esses cortes podem conduzir a decisões com custo total superior ao de aproximações mais conservadoras, como a imposição incondicional de restrições, isto é, uma modelagem mais restritiva do fenômeno não convexo.

No que se refere aos cortes exatos, os cortes de Lagrange mostraram-se mais robustos do que os cortes Inteiros de Otimalidade, embora a construção e a aplicação desses cortes impliquem um custo computacional significativamente mais elevado. Ainda assim, os resultados indicam que os cortes de Lagrange também apresentam limitações práticas relacionadas à taxa de convergência, o que pode inviabilizar sua aplicação em determinados contextos. Dessa forma, identifica-se um espaço relevante para aprimoramentos na SDDiP/DDiP com cortes de Lagrange; uma discussão mais aprofundada sobre aspectos de convergência e possíveis estratégias de melhoria são apresentada nos Capítulos 6 e 7.

6 Análise das soluções duais

No contexto dos métodos SDDiP e DDiP, os cortes de Lagrange desempenham um papel fundamental, pois admitem representação linear e ainda garantem a convergência do algoritmo mesmo quando os subproblemas são MILPs. Além disso, apresentam propriedades que, em geral, resultam em um desempenho significativamente superior ao obtido com cortes Inteiros de Otimalidade. Entretanto, diversos autores observaram que a obtenção das soluções duais necessárias para a construção desses cortes é computacionalmente onerosa quando comparada às demais etapas dos algoritmos [43, 50, 65, 71], tornando essa fase um fator crítico para a aplicabilidade prática da SDDiP. Apesar de sua relevância, não foram encontrados trabalhos que investiguem a estrutura do problema dual com o objetivo de identificar soluções duais mais adequadas para a geração dos cortes. Assim, este capítulo tem como objetivo apresentar uma análise conceitual das soluções duais obtidas ao longo da execução dos métodos SDDiP e DDiP para a construção de cortes de Lagrange.

Para isso, são exploradas as propriedades do conjunto de soluções do problema de Lagrange (4.14), responsável por determinar os multiplicadores duais associados à restrição de cópia (Equação 4.10). Mostra-se que, embora os cortes de Lagrange sejam sempre válidos no domínio da função de recurso e exatos no ponto em que são gerados, diferentes cortes podem ser obtidos em função da solução dual selecionada ou da formulação adotada para a variável de cópia. De maneira geral, os cortes mais eficazes são aqueles menos relaxados sobre o restante do domínio da função de recurso, característica que influencia diretamente o desempenho dos métodos SDDiP e DDiP, tanto em termos de tempo computacional quanto de velocidade de convergência.

Por fim, são apresentados os métodos tradicionalmente utilizados para resolver o problema dual (4.14), bem como discutidas suas principais características e implicações quando empregados na construção de cortes de Lagrange.

6.1 Infinitas soluções duais e solução dual extrema

Ao analisar o problema de Lagrange (4.14), obtido pela relaxação das restrições de cópia, identifica-se uma característica relevante: o conjunto de soluções duais é infinito e ilimitado. Tal propriedade pode ser explorada para aprimorar o processo de construção dos cortes de Lagrange, contribuindo para um melhor desempenho da SDDiP. A seguir, apresenta-se uma análise que evidencia essa característica e discute como a escolha da solução dual influencia a qualidade dos cortes gerados.

Considera-se um exemplo extraído de [50] (Exemplo 2), no qual a função de recurso é definida por:

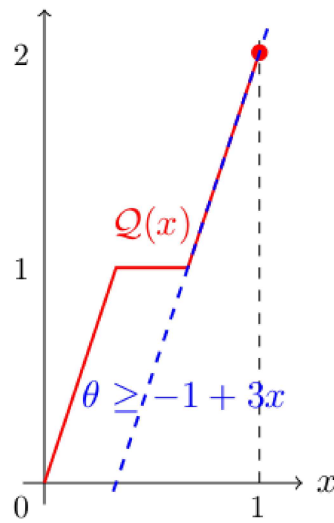


Figura 9 – Função de recurso $Q(x)$ retirada de [50]

$$Q(x) = \min y_1 + y_2 \quad (6.1)$$

$$s.t. \quad 2 \cdot y_1 + y_2 \geq 3 \cdot z \quad (6.2)$$

$$z = x \quad (6.3)$$

$$0 \leq y_1 \leq 2 \quad (6.4)$$

$$0 \leq y_2 \leq 3 \quad (6.5)$$

$$0 \leq z \leq 1 \quad (6.6)$$

$$y_1, z \in \mathbb{Z}, y_2 \in \mathbb{R} \quad (6.7)$$

A função Q depende da variável de recurso binária $x \in \{0, 1\}$, enquanto y_1 , y_2 e z são variáveis de decisão. A Equação (6.3) representa a restrição de cópia introduzida para viabilizar a construção dos cortes de Lagrange. Neste exemplo, a variável de cópia z é considerada inteira. Ressalta-se que [50] demonstra a validade e exatidão dos cortes de Lagrange tanto para a variável de cópia inteira quanto contínua, sem, contudo, detalhar as diferenças entre essas modelagens — aspecto que será discutido posteriormente neste capítulo. A forma da função de recurso $Q(x)$ é apresentada pelos traços em vermelho na Figura 9.

Os autores constroem o corte de Lagrange no ponto $x^* = 1$ por meio da relaxação da restrição de cópia (6.3), resultando na função Lagrangiana:

$$L(\mu, x^*) = \min y_1 + y_2 - \mu(z - 1) \quad (6.8)$$

$$s.t. \quad 2 \cdot y_1 + y_2 \geq 3 \cdot z \quad (6.9)$$

$$0 \leq y_1 \leq 2 \quad (6.10)$$

$$0 \leq y_2 \leq 3 \quad (6.11)$$

$$0 \leq z \leq 1 \quad (6.12)$$

$$y_1, z \in \mathbb{Z}, y_2 \in \mathbb{R} \quad (6.13)$$

O valor de $\mu \in \mathbb{R}$ que maximiza essa função corresponde a uma solução dual associada às restrições de cópia do problema primal $\mathcal{Q}(x^*)$:

$$z_D = \max_{\mu} L(\mu, x^*) \quad (6.14)$$

A partir da solução dual $\mu^* = 3$ (utilizada em [50]) e do valor $\mathcal{Q}(1) = 2$, obtém-se o corte de Lagrange:

$$\alpha \geq \mathcal{Q}(1) + \mu^* \cdot (x - x^*)$$

$$\alpha \geq 2 + 3 \cdot (x - 1)$$

$$\alpha \geq -1 + 3 \cdot x$$

conforme ilustrado pela reta azul na Figura 9.

Entretanto, o objetivo desta seção é analisar o processo de obtenção do valor dual e seu impacto na construção dos cortes. Para isso, considera-se o problema $L(\mu, x^*)$ definido em (6.8)-(6.13), que admite solução analítica simples, caracterizada por:

1. quando $\mu < 2$:

a solução primal é $y_1 = 0$, $y_2 = 0$ e $z = 0$, resultando em

$$L(\mu, x^*) = y_1 + y_2 - \mu(z - 1) = \mu$$

2. quando $\mu > 2$:

a solução primal é $y_1 = 1$, $y_2 = 1$ e $z = 1$, resultando em

$$L(\mu, x^*) = y_1 + y_2 - \mu(z - 1) = 2$$

3. quando $\mu = 2$:

ambas as soluções são ótimas, resultando em

$$L(\mu, x^*) = y_1 + y_2 - \mu(z - 1) = 2$$

Dessa forma, a função $L(\mu, x^*)$ pode ser expressa como:

$$L(\mu, x^*) = \begin{cases} \mu, & \text{se } \mu < 2 \\ 2, & \text{se } \mu \geq 2 \end{cases}$$

A Figura 10 (a) ilustra graficamente essa função. A maximização de $L(\mu, x^*)$, conforme (6.14), resulta em $z_D = 2$ para todo $\mu \in [2, \infty)$. Assim, o problema dual admite infinitas soluções, e qualquer valor nesse intervalo gera um corte de Lagrange válido e exato, conforme mostrado na Figura 11 (lado esquerdo).

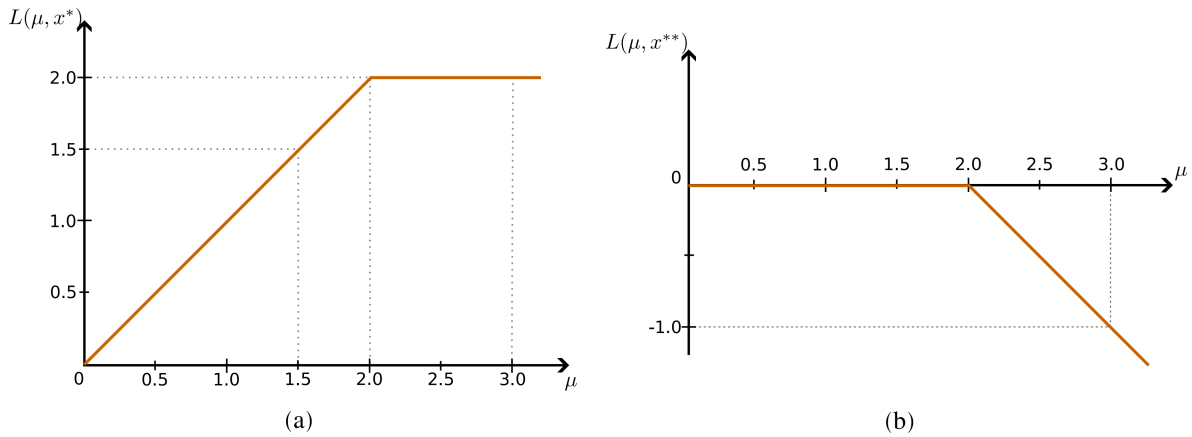


Figura 10 – Função de Lagrange para: (a) para $x = 1$ e (b) para $x = 0$.

Realizando procedimento semelhante para o ponto $x^{**} = 0$, obtém-se:

$$L(\mu, x^{**}) = \begin{cases} 0, & \text{se } \mu < 2 \\ 2 - \mu, & \text{se } \mu \geq 2 \end{cases}$$

A Figura 10 (b) apresenta o gráfico correspondente. Nesse caso, a maximização de $L(\mu, x^{**})$ implica que qualquer $\mu \in (-\infty, 2]$ é solução dual ótima para $\mathcal{Q}(x)$, novamente, o ponto considerado admite infinitos cortes válidos e exatos, conforme ilustrado na Figura 11 (lado direito).

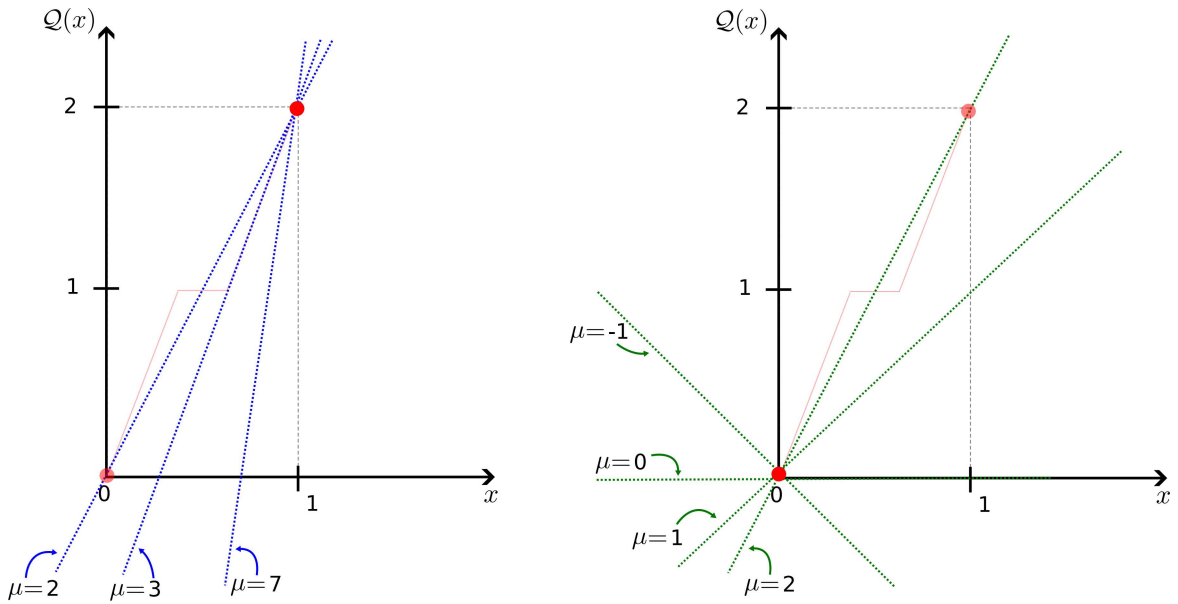


Figura 11 – Cortes de Lagrange para o problema exemplo de [50], considerando variável de cópia inteira. Em azul, cortes no ponto $x = 1$; em verde, cortes no ponto $x = 0$.

A natureza infinita e ilimitada do conjunto de soluções duais é intuitiva no caso unidimensional, mas pode ser estabelecida de forma geral. Considere a seguinte função de recurso, para um período t e cenário s :

$$Q_{t,s}(x_{t-1,s^{(a)}}) = \min_{y_{t,s}, z_{t,s}} f(y_{t,s}, z_{t,s}) \quad (6.15)$$

$$s.t. \quad (y_{t,s}, z_{t,s}) \in \mathcal{V} \quad (6.16)$$

$$z_{t,s} = x_{t-1,s^{(a)}}. \quad (6.17)$$

Esse subproblema corresponde à formulação apresentada em (4.7)-(4.10), escrita de forma condensada. A função f é linear e $\mathcal{V}$ representa um conjunto viável inteiro misto descrito por restrições afins. As variáveis são:

- $x_{t-1,s^{(a)}} \in \{0, 1\}^N$: vetor de variáveis de recurso, correspondente à decisão no período $t - 1$ no cenário antecessor $s^{(a)}$;
- $y_{t,s} \in \mathbb{R}^M \times \mathbb{Z}^P$: vetor de variáveis de decisão, composto por M variáveis contínuas e P inteiras;
- $z_{t,s} \in \{0, 1\}^N$: variáveis auxiliares introduzidas para representar a restrição de cópia.

A função Lagrangiana associada à relaxação das restrições de cópia é:

$$L_{t,s}(\mu_{t,s}, x_{t-1,s(a)}) = \min_{y_{t,s}, z_{t,s}} f(y_{t,s}, z_{t,s}) - \mu_{t,s}^\top \cdot (z_{t,s} - x_{t-1,s(a)}) \quad (6.18)$$

$$s.t. \quad (y_{t,s}, z_{t,s}) \in \mathcal{V}. \quad (6.19)$$

As soluções duais são obtidas pela resolução do problema:

$$(P) \quad \max_{\mu_{t,s}} L_{t,s}(\mu_{t,s}, x_{t-1,s(a)}), \quad (6.20)$$

cujos conjunto de soluções ótimas é definido por:

$$\mathcal{D}(x_{t-1,s(a)}) = \{\mu_{t,s} \in \mathbb{R}^N \mid \mu \in \arg \max_{\mu_{t,s}} L_{t,s}(\mu_{t,s}, x_{t-1,s(a)})\}. \quad (6.21)$$

Com base nessa formulação, estabelecem-se as seguintes proposições.

Proposição 1. *Considere, para um estágio t e cenário s fixos, o subproblema de recurso $\mathcal{Q}_{t,s}(x_{t-1,s(a)})$ dado por (6.15)-(6.17), no qual o vetor $x_{t-1,s(a)} \in \{0, 1\}^N$ atua apenas por meio da restrição de cópia (6.17). Seja (6.20) o problema dual de maximização obtido pela relaxação Lagrangiana dessa restrição, com multiplicadores $\mu_{t,s} \in \mathbb{R}^N$. Suponha que: (i) o conjunto viável misto-inteiro $\mathcal{V}$ desse subproblema é compacto; (ii) a propriedade de complete recourse é satisfeita, isto é, existe solução viável para o subproblema $\mathcal{Q}_{t,s}(x_{t-1,s(a)}) < +\infty$ para qualquer valor do vetor $x_{t-1,s(a)}$.*

Então, existe solução ótima finita para o problema dual (6.20). Ou seja, existe $\bar{M} \in \mathbb{R}^+$ e uma solução ótima $\bar{\mu}_{t,s} = (\bar{\mu}_1, \dots, \bar{\mu}_N)$ tal que, para cada componente i , vale $-\infty < -\bar{M} \leq \bar{\mu}_i \leq \bar{M} < +\infty$.

Demonstração. Denota-se o vetor de variáveis de recurso $x_{t-1,s(a)}$ por $x^{(a)}$, e os índices t, s serão suprimidos para simplificar a notação. Como o problema $\mathcal{Q}(x^{(a)})$ é linear, seu conjunto viável é compacto e a propriedade de *complete recourse* é satisfeita, existem constantes finitas $\underline{Q}, \bar{Q} \in \mathbb{R}$, tais que:

$$\underline{Q} \leq \mathcal{Q}(x^{(a)}) \leq \bar{Q}, \quad \forall x^{(a)} \in \{0, 1\}^N \quad (6.22)$$

e, portanto, $\Delta = \bar{Q} - \underline{Q} < +\infty$. Defina $\bar{M} = \Delta + \varepsilon$, para $\varepsilon > 0$, e o vetor $\bar{\mu} \in \mathbb{R}^N$, componente a componente, por

$$\bar{\mu}_i := (1 - 2x_i^{(a)})\bar{M}, \quad i = 1, \dots, N,$$

isto é, $\bar{\mu}_i = +\bar{M}$ se $x_i^{(a)} = 0$ e $\bar{\mu}_i = -\bar{M}$ se $x_i^{(a)} = 1$.

Considere agora a função de Lagrange (6.18)-(6.19) em $\bar{\mu}$, ou seja, $L(\bar{\mu}, x^{(a)})$. Para qualquer $(y, z) \in \mathcal{V}$, como $z, x^{(a)} \in \{0, 1\}^N$, tem-se

$$\bar{\mu}_i(z_i - x_i^{(a)}) = \begin{cases} 0, & \text{se } z_i = x_i^{(a)}, \\ \bar{M}, & \text{se } z_i \neq x_i^{(a)}, \end{cases}$$

logo $\bar{\mu}^\top (z - x^{(a)}) = \bar{M} \cdot m$, onde m é o número de componentes em que z difere de $x^{(a)}$.

Dado que a condição (6.22) é válida para qualquer valor de $x^{(a)}$, e que a variável z copia o valor de $x^{(a)}$, então os limites também são válidos para qualquer valor de z . Pela definição da função $L(\mu, x^{(a)})$, tem-se:

$$\underline{Q} + \bar{M} \cdot m \leq L(\bar{\mu}, x^{(a)}).$$

Pelo teorema da dualidade fraca, temos que:

$$\underline{Q} + \bar{M} \cdot m \leq L(\bar{\mu}, x^{(a)}) \leq Q(x^{(a)}).$$

Substituindo $\bar{M}$ pela sua definição e aplicando (6.22):

$$\begin{aligned} \underline{Q} + m \cdot \bar{Q} - m \cdot \underline{Q} + m \cdot \varepsilon &\leq L(\bar{\mu}, x^{(a)}) \leq Q(x^{(a)}) \leq \bar{Q} \\ \underline{Q} + m \cdot \bar{Q} - m \cdot \underline{Q} + m \cdot \varepsilon &\leq \bar{Q} \\ \underline{Q} - m \cdot \underline{Q} + m \cdot \bar{Q} - \bar{Q} + m \cdot \varepsilon &\leq 0 \\ (1 - m) \cdot (\underline{Q} - \bar{Q}) + m \cdot \varepsilon &\leq 0 \end{aligned} \tag{6.23}$$

Assim, para $m > 0$ a desigualdade (6.23) fica inválida, pois $\underline{Q} - \bar{Q} \leq 0$, logo $m = 0$. Como m corresponde ao número de componentes para as quais z difere de $x^{(a)}$, então $z_i = x_i^{(a)}, \forall i$. Logo, $\bar{\mu}$ é uma solução dual ótima e finita. $\square$

Proposição 2. *Considere, para um estágio t e cenário s fixos, o subproblema de recurso $Q_{t,s}(x_{t-1,s(a)})$ dado por (6.15)-(6.17), no qual o vetor $x_{t-1,s(a)} \in \{0, 1\}^N$ atua apenas por meio da restrição de cópia (6.17). Seja (6.20) o problema dual de maximização obtido pela relaxação Lagrangiana dessa restrição, com multiplicadores $\mu_{t,s} \in \mathbb{R}^N$. Suponha que: (i) o conjunto viável misto-inteiro $\mathcal{V}$ desse subproblema é compacto; (ii) a propriedade de recurso completo (complete recourse) é satisfeita, isto é, existe solução viável para o subproblema $(Q_{t,s}(x_{t-1,s(a)}) < +\infty)$ para qualquer valor do vetor $x_{t-1,s(a)}$.*

Então, o conjunto de soluções ótimas do problema dual é ilimitado nas direções compatíveis com o vetor de recurso, tais que, para cada direção i :

$$\begin{aligned} \sup_{\mu_{t,s} \in \mathcal{D}(x_{t-1,s(a)})} \mu_{i,t,s} &= +\infty \quad \text{se} \quad x_{i,t-1,s(a)} = 1 \\ \inf_{\mu_{t,s} \in \mathcal{D}(x_{t-1,s(a)})} \mu_{i,t,s} &= -\infty \quad \text{se} \quad x_{i,t-1,s(a)} = 0. \end{aligned}$$

Em particular, $\tilde{\mu} = (\tilde{\mu}_1, \dots, \tilde{\mu}_N)$ definido como:

$$\tilde{\mu}_i = \begin{cases} +\infty, & \text{se } x_{i,t-1,s(a)} = 1, \\ -\infty, & \text{se } x_{i,t-1,s(a)} = 0, \end{cases}$$

é um ponto limite do conjunto de soluções ótimas.

Demonstração. Denote $x_{t-1,s(a)}$ por $x^{(a)}$ e suprima os índices (t,s) para simplificar a notação. As hipóteses de compacidade do conjunto viável e de recurso completo correspondem às condições (A1) e (A2) de [50], as quais garantem dualidade forte para o problema dual (6.20). Em particular, o valor ótimo dual coincide com o valor da função de recurso $\mathcal{Q}(x^{(a)})$.

Considere agora uma solução ótima dual $\bar{\mu}$ finita associada ao parâmetro $\bar{x}^{(a)}$ para o problema (6.20). Fixe um índice $i \in \{1, \dots, N\}$. Pela Proposição 1, tem-se que $\bar{\mu}_i$ pode ser caracterizado por $-\infty < -\bar{M} \leq \bar{\mu}_i \leq \bar{M} < +\infty$.

Da igualdade entre os valores primal e dual, tem-se $\mathcal{Q}(\bar{x}^{(a)}) = L(\bar{\mu}, \bar{x}^{(a)})$, e da forma do Lagrangiano (6.18)-(6.19), segue que, em qualquer solução ótima, vale

$$\bar{\mu}^\top (z - \bar{x}^{(a)}) = 0.$$

Primeiramente, suponha que $\bar{x}_i^{(a)} = 0$. Neste caso, deve necessariamente ocorrer $z_i = 0$ em uma solução primal ótima associada. De fato, se $z_i = 1$, então o termo dual $\bar{\mu}_i(z_i - \bar{x}_i^{(a)})$ seria não nulo. Assim, o valor de $\bar{\mu}_i$ poderia ser alterado indefinidamente sem violar a viabilidade dual, resultando em um aumento indefinido do valor do Lagrangiano e contradizendo a dualidade forte. Pela hipótese de *complete recourse*, a escolha $z_i = 0$ é sempre viável.

Adicionalmente, visto que o valor $\bar{\mu}_i$ é suficiente para manter $z_i = 0$ em (6.18), se o componente $\bar{\mu}_i$ for arbitrariamente reduzido, z_i permanece em 0 uma vez que $z_i \in \{0, 1\}$ e observa-se que o valor da função dual não se altera. Portanto, o conjunto de soluções duais ótimas é não limitado inferiormente na componente i .

De forma análoga, suponha agora que $\bar{x}_i^{(a)} = 1$. Neste caso, deve ocorrer $z_i = 1$ em uma solução primal ótima associada. Caso contrário, se $z_i = 0$, o valor de $\bar{\mu}_i$ poderia ser alterado indefinidamente sem violar a viabilidade dual, resultando em um aumento indefinido do valor do Lagrangiano e contradizendo a dualidade forte. Pela hipótese de *complete recourse*, a escolha $z_i = 1$ é viável.

Adicionalmente, visto que o valor $\bar{\mu}_i$ é suficiente para manter $z_i = 1$ em (6.18), se o componente $\bar{\mu}_i$ for arbitrariamente aumentado, z_i permanece em 1 uma vez que $z_i \in \{0, 1\}$ e observa-se que o valor da função dual não se altera. Portanto, o conjunto de soluções duais ótimas é não limitado superiormente na componente i .

Como o argumento se aplica a cada componente i , conclui-se que o conjunto de soluções ótimas do problema dual é não limitado, o que implica a existência de direções ao longo das quais $\mu_i \rightarrow +\infty$ ou $\mu_i \rightarrow -\infty$ sem alteração do valor ótimo. Isso completa a prova. $\square$

As proposições 1 e 2 demonstram duas importantes propriedades do conjunto de soluções duais ótimas resolvido no processo da SDDiP, a partir dessas propriedades deriva-se o seguinte Corolário.

Corolário 1. *Considere as hipóteses estabelecidas nas Proposições 1 e 2. Então, o conjunto de soluções ótimas do problema dual dado por (6.21) é não vazio, convexo, ilimitado e infinito.*

Demonstração. Pela Proposição 1, existe ao menos uma solução dual ótima finita. Pela Proposição 2, o conjunto de soluções duais ótimas é ilimitado nas direções indicadas por $x^{(a)}$. Além disso, como $L(\cdot, x^{(a)})$ é côncava, o conjunto de maximizadores $\mathcal{D}(x^{(a)})$ é convexo. Portanto, $\mathcal{D}(x^{(a)})$ é não vazio, convexo, ilimitado e infinito.

□

A partir das Proposições 1 e 2 e do Corolário 1, estabelecem-se as seguintes propriedades:

Propriedade 1. *O corte de Lagrange construído a partir de qualquer solução dual ótima é válido e exato, conforme demonstrado em [50]. Consequentemente, para cada ponto avaliado, existe um número infinito de cortes de Lagrange.*

Propriedade 2. *Quanto maior o valor da componente dual $\bar{\mu}_i$ quando $x_i^{(a)} = 0$, ou quanto menor esse valor quando $x_i^{(a)} = 1$, mais o corte se aproxima da função de recurso em todo o domínio. Em outras palavras, soluções duais localizadas nos extremos do conjunto de soluções produzem cortes de Lagrange mais fortes.*

Com base na Propriedade 2, a Definição 1 introduz o conceito de *Ponto Extremo do Conjunto Dual Ótimo*, no contexto deste trabalho. Esse conceito tem por objetivo caracterizar soluções duais situadas nos extremos do conjunto $\mathcal{D}(x^{(a)})$ e será utilizado posteriormente na análise e construção das variantes dos cortes de Lagrange.

Definição 1 (Ponto extremo do conjunto dual ótimo). *Suprimindo os índices relativos ao período e cenário, considere o conjunto de soluções duais ótimas $\mathcal{D}(x^{(a)})$ para um dado $x^{(a)}$, conforme definido em (6.21).*

Seja $\Delta \in \mathbb{R}^N$ um vetor cujas componentes satisfazem

$$\begin{cases} \Delta_i \geq 0, & \text{se } x_i^{(a)} = 0, \\ \Delta_i \leq 0, & \text{se } x_i^{(a)} = 1. \end{cases} \quad (6.24)$$

Adicionalmente,

$$1^\top \Delta \neq 0. \quad (6.25)$$

Um ponto $\mu^* \in \mathcal{D}(x^{(a)})$ é dito um ponto extremo de $\mathcal{D}(x^{(a)})$ se, para todo vetor Δ que satisfaça (6.24) e (6.25),

$$\mu^* + \Delta \notin \mathcal{D}(x^{(a)}).$$

Denota-se por

$$\text{Ext}\left(\mathcal{D}(x^{(a)})\right) = \left\{ \mu^* \in \mathcal{D}(x^{(a)}) : \mu^* \text{ é um ponto extremo de } \mathcal{D}(x^{(a)}) \right\}$$

o conjunto dos pontos extremos de $\mathcal{D}(x^{(a)})$ conforme a Definição 1.

6.2 Impacto da consideração da variável de cópia contínua

Retomando o problema unidimensional apresentado em (6.1)–(6.7) e redefinindo a variável z na Equação (6.7) como contínua, isto é, $z \in [0, 1]$, observa-se que o problema primal preserva tanto o conjunto viável quanto a solução ótima do caso original. Isso decorre do fato de que z permanece fixada pela restrição de cópia, $z = x$, com $x \in \{0, 1\}$.

Em contrapartida, o problema dual $L(\mu, x^*)$, obtido a partir da relaxação da restrição de cópia, apresenta comportamento distinto. Nesse caso, o problema Lagrangiano passa a admitir três soluções primais candidatas:

1. quando $\mu < 1.5$:

a solução primal é $y_1 = 0$, $y_2 = 0$ e $z = 0$, resultando em

$$L(\mu, x^*) = y_1 + y_2 - \mu(z - 1) = \mu$$

2. quando $1.5 < \mu < 3$:

a solução primal é $y_1 = 1$, $y_2 = 0$ e $z = 2/3$, resultando em

$$L(\mu, x^*) = y_1 + y_2 - \mu(z - 1) = 1 + \frac{1}{3} \cdot \mu$$

3. quando $\mu > 3$:

a solução primal é $y_1 = 1$, $y_2 = 1$ e $z = 1$, resultando em

$$L(\mu, x^*) = y_1 + y_2 - \mu(z - 1) = 2$$

Assim, a função $L(\mu, x^*)$ pode ser escrita como:

$$L(\mu, x^*) = \begin{cases} \mu, & \text{se } \mu \leq 1.5 \\ 1 + \frac{1}{3} \cdot \mu, & \text{se } 1.5 < \mu < 3 \\ 2, & \text{se } \mu \geq 3 \end{cases}$$

A reta verde na Figura 12 (a) ilustra o comportamento dessa função. Ao maximizar $L(\mu, x^*)$, obtém-se um subconjunto mais restrito de soluções duais ótimas, dado por $\mu \in [3, \infty)$.

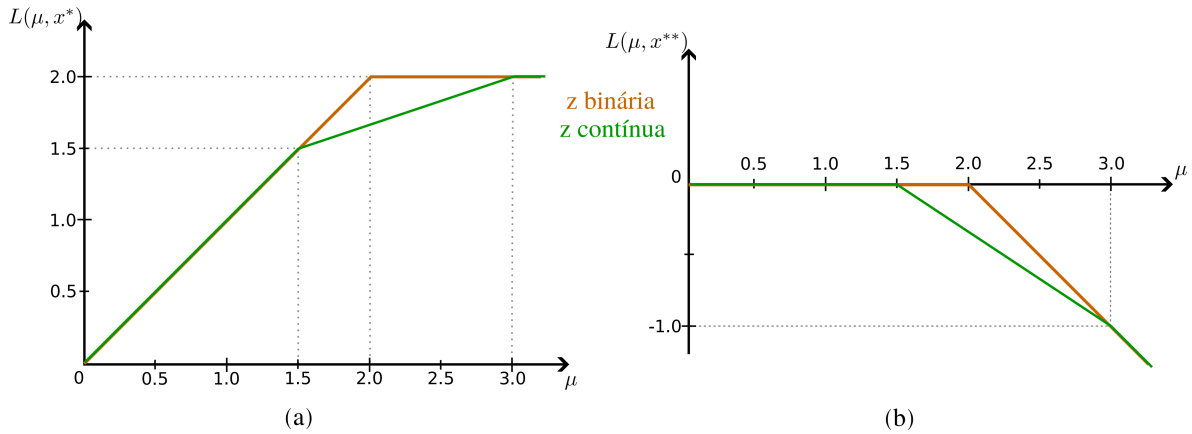


Figura 12 – Fun o de Lagrange para: (a) para $x = 1$ e (b) para $x = 0$. A curva laranja representa o caso com vari vel de c pia inteira, enquanto a curva verde corresponde   vari vel cont ua.

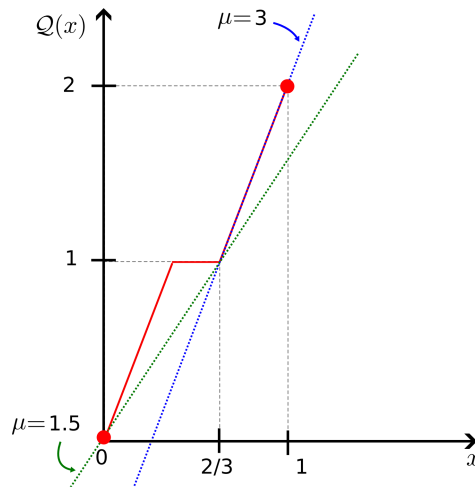


Figura 13 – Cortes de Lagrange (lim trofes) obtidos considerando a vari vel de c pia cont ua.

Isso ocorre porque, ao considerar z cont ua, o problema passa a explorar todo o intervalo $[0, 1]$, e n o apenas os pontos discretos $x = 0$ e $x = 1$ pertencentes ao dom nio original de $Q(x)$.

An lise an loga pode ser realizada para $x^{**} = 0$, resultando em:

$$L(\mu, x^{**}) = \begin{cases} 0, & \text{se } \mu \leq 1.5 \\ 1 - \frac{2}{3} \cdot \mu, & \text{se } 1.5 < \mu < 3 \\ 2 - \mu, & \text{se } \mu \geq 3 \end{cases}$$

A Figura 12 (b) apresenta o gr fico correspondente. Nesse caso, a maximiza  o conduz ao conjunto de solu  es  timas $\mu \in (-\infty, 1.5]$, que   estritamente menor do que o obtido quando $z \in \mathbb{Z}$. A Figura 13 mostra os cortes lim trofes — isto  , aqueles associados ao maior valor de μ para $x = 0$ e ao menor valor de μ para $x = 1$ — evidenciando que tais cortes permanecem v lidos para $Q(x)$ em todo o intervalo $x \in [0, 1]$.

Em síntese, conforme argumentado em [50], tanto a representação inteira quanto a contínua para a variável de cópia produzem cortes de Lagrange válidos e exatos. Contudo, a análise do exemplo evidencia que a função Lagrangiana difere entre essas abordagens, como ilustrado na Figura 12. No caso geral, pode-se demonstrar que o conjunto de cortes obtido com a variável de cópia inteira contém aquele correspondente à formulação contínua. A partir dessa observação, derivam-se propriedades adicionais, apresentadas na sequência.

Proposição 3. *Considere as seguintes funções de recurso*

$$\mathcal{Q}^I(x^{(a)}) = \min_{y,z} f(y,z) \quad (6.26)$$

$$s.t. (y,z) \in \mathcal{V}^I = \{(y,z) \in \mathcal{V} : z \in \{0,1\}^N\} \quad (6.27)$$

$$z = x^{(a)}, \quad (6.28)$$

$$\mathcal{Q}^C(x^{(a)}) = \min_{y,z} f(y,z) \quad (6.29)$$

$$s.t. (y,z) \in \mathcal{V}^C = \{(y,z) \in \mathcal{V} : z \in [0,1]^N\} \quad (6.30)$$

$$z = x^{(a)}, \quad (6.31)$$

onde f é linear e $\mathcal{V}$ é o conjunto viável misto-inteiro definido por restrições afins, as variáveis de decisão y são um vetor inteiro misto e a variável de recurso $x^{(a)} \in \{0,1\}^N$. Ambos os problemas possuem recurso completo (complete recourse) e $\mathcal{V}$ é compacto.

Considere as funções lagrangianas associadas à relaxação da restrição de cópia:

$$L^I(\mu, x^{(a)}) = \min_{(y,z) \in \mathcal{V}^I} \left\{ f(y,z) - \mu^\top (z - x^{(a)}) \right\},$$

$$L^C(\mu, x^{(a)}) = \min_{(y,z) \in \mathcal{V}^C} \left\{ f(y,z) - \mu^\top (z - x^{(a)}) \right\},$$

Os conjuntos de soluções duais para um dado $\bar{x}^{(a)}$ são dados por:

$$\mathcal{D}^I(\bar{x}^{(a)}) = \{\mu \in \mathbb{R}^N \mid \mu = \arg \max_{\mu} L^I(\mu, \bar{x}^{(a)})\}$$

$$\mathcal{D}^C(\bar{x}^{(a)}) = \{\mu \in \mathbb{R}^N \mid \mu = \arg \max_{\mu} L^C(\mu, \bar{x}^{(a)})\}$$

Então:

$$(a) \mathcal{D}^C(\bar{x}^{(a)}) \subseteq \mathcal{D}^I(\bar{x}^{(a)}) \quad \forall \bar{x}^{(a)};$$

$$(b) \text{ não necessariamente } \mathcal{D}^I(\bar{x}^{(a)}) \subseteq \mathcal{D}^C(\bar{x}^{(a)}) \text{ para todo } \bar{x}^{(a)};$$

Demonstração. Como $x^{(a)} \in \{0,1\}^N$ e, no problema primal, a variável z está fixa ao valor de $x^{(a)}$, tem-se, para todo $x^{(a)}$:

$$\mathcal{Q}^I(x^{(a)}) = \mathcal{Q}^C(x^{(a)}).$$

Já o problema de Lagrange, pelo fato de que $\mathcal{V}^I \subseteq \mathcal{V}^C$, e se tratando de um problema de minimização, tem-se:

$$L^C(\mu, x^{(a)}) \leq L^I(\mu, x^{(a)}). \quad (6.32)$$

As hipóteses de compacidade do conjunto viável e de *complete recourse* correspondem às condições (A1) e (A2) de [50], as quais garantem dualidade forte, ou seja, dado um valor $\bar{x}^{(a)}$:

$$\max_{\mu} L^I(\mu, \bar{x}^{(a)}) = \mathcal{Q}^I(\bar{x}^{(a)}) = \mathcal{Q}^C(\bar{x}^{(a)}) = \max_{\mu} L^C(\mu, \bar{x}^{(a)}).$$

Assim,

$$\forall \bar{\mu} \in \mathcal{D}^C(\bar{x}^{(a)}) \rightarrow L^C(\bar{\mu}, \bar{x}^{(a)}) = \mathcal{Q}^C(\bar{x}^{(a)}) = \mathcal{Q}^I(\bar{x}^{(a)}).$$

Combinando com (6.32), obtém-se que

$$\forall \bar{\mu} \in \mathcal{D}^C(\bar{x}^{(a)}) \rightarrow L^C(\bar{\mu}, \bar{x}^{(a)}) = \mathcal{Q}^C(\bar{x}^{(a)}) = \mathcal{Q}^I(\bar{x}^{(a)}) \leq L^I(\bar{\mu}, \bar{x}^{(a)}),$$

e pelo teorema da dualidade fraca: $\mathcal{Q}^I(x^{(a)}) \geq L^I(\mu, x^{(a)})$, portanto a relação anterior só pode ocorrer na igualdade, ou seja:

$$\forall \bar{\mu} \in \mathcal{D}^C(\bar{x}^{(a)}) \rightarrow L^C(\bar{\mu}, \bar{x}^{(a)}) = \mathcal{Q}^C(\bar{x}^{(a)}) = \mathcal{Q}^I(\bar{x}^{(a)}) = L^I(\bar{\mu}, \bar{x}^{(a)}).$$

Logo:

$$\forall \bar{\mu} \in \mathcal{D}^C(\bar{x}^{(a)}) \rightarrow \bar{\mu} \in \mathcal{D}^I(\bar{x}^{(a)}),$$

o que prova (a). Para provar (b), utiliza-se como contraexemplo o caso unidimensional descrito neste capítulo. Nesse exemplo, $\mu = 2 \in \mathcal{D}^I(\bar{x}^{(a)})$, mas $\mu = 2 \notin \mathcal{D}^C(\bar{x}^{(a)})$. Portanto, não necessariamente $\mathcal{D}^I \subseteq \mathcal{D}^C$. $\square$

Com base nos resultados anteriores, podem ser estabelecidas as seguintes propriedades:

Propriedade 3. *Pela Proposição 3, o conjunto de soluções duais associado à representação inteira das variáveis de cópia contém aquele correspondente à formulação contínua. Como consequência, a modelagem inteira possibilita a obtenção de cortes de Lagrange potencialmente mais fortes.*

No caso unidimensional, por exemplo, o multiplicador dual $\mu = 2$ é ótimo na formulação inteira, mas não na contínua. Ademais, o corte associado a esse valor é exato não apenas no ponto x em que foi gerado, mas também no outro ponto admissível do domínio.

Propriedade 4. *Na formulação com variável de cópia contínua, os cortes de Lagrange obtidos, além de exatos, são válidos não apenas nos pontos discretos $x \in \{0, 1\}$, mas também ao longo de todo o intervalo contínuo $x \in (0, 1)$. Essa propriedade é particularmente relevante em contextos nos quais a variável de estado não é inerentemente binária.*

6.3 Infinitas soluções duais extremas

A análise da Seção 6.1 evidencia que existem infinitas soluções duais associadas à restrição de cópia, o que implica a existência de infinitos cortes válidos e exatos. Apesar disso, observa-se a presença de uma solução dual extrema ($\mu = 2$) que produz um corte particularmente forte, sugerindo, à primeira vista, que a escolha da solução dual seria trivial.

Entretanto, em problemas multidimensionais, tal solução limite pode não ser única. Nesses casos, há um conjunto infinito de soluções duais extremas, sem um elemento dominante. Para ilustrar esse cenário, considere a função de recurso bidimensional $Q(x_1, x_2)$, onde as variáveis de recurso são $x_1 \in \{0, 1\}$ e $x_2 \in \{0, 1\}$:

$$\begin{aligned} Q(x_1, x_2) = \min \quad & y_1 + y_2 \\ \text{s.t.} \quad & 2 \cdot y_1 + y_2 \geq 3 \cdot z_1 + 4 \cdot z_2 \\ & z_1 = x_1 \\ & z_2 = x_2 \\ & 0 \leq y_1 \leq 2 \\ & 0 \leq y_2 \leq 3 \\ & 0 \leq z_1 \leq 1 \\ & 0 \leq z_2 \leq 1 \\ & y_1, z_1, z_2 \in \mathbb{Z}, y_2 \in \mathbb{R}. \end{aligned}$$

O Lagrangiano obtido ao relaxar as restrições de cópia é dado por:

$$\begin{aligned} L(\mu_1, \mu_2, x_1, x_2) = \min \quad & y_1 + y_2 - \mu_1(z_1 - x_1) - \mu_2(z_2 - x_2) \\ \text{s.t.} \quad & 2 \cdot y_1 + y_2 \geq 3 \cdot z_1 + 4 \cdot z_2 \\ & 0 \leq y_1 \leq 2 \\ & 0 \leq y_2 \leq 3 \\ & 0 \leq z_1 \leq 1 \\ & 0 \leq z_2 \leq 1 \\ & y_1, z_1, z_2 \in \mathbb{Z}, y_2 \in \mathbb{R}. \end{aligned}$$

A solução do problema para cada ponto do domínio é:

$$Q(0, 0) = 0 \quad \text{e} \quad z_D = \max_{\mu} L(\mu_1, \mu_2, 0, 0) = 0 \quad (6.33)$$

$$Q(1, 0) = 2 \quad \text{e} \quad z_D = \max_{\mu} L(\mu_1, \mu_2, 1, 0) = 2 \quad (6.34)$$

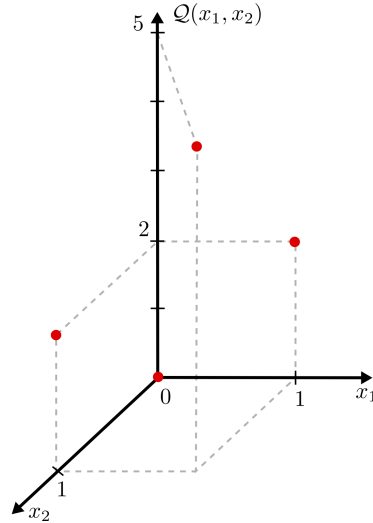


Figura 14 – Função de recurso do exemplo bidimensional

$$Q(0, 1) = 2 \quad \text{e} \quad z_D = \max_{\mu} L(\mu_1, \mu_2, 0, 1) = 2 \quad (6.35)$$

$$Q(1, 1) = 5 \quad \text{e} \quad z_D = \max_{\mu} L(\mu_1, \mu_2, 1, 1) = 5 \quad (6.36)$$

Verifica-se, portanto, ausência de gap de dualidade. A Figura 14 apresenta a função de recurso nos quatro pontos do domínio.

6.3.1 Análise no espaço dual

Conforme estabelecido no Corolário 1, o conjunto de soluções duais ótimas, $\mathcal{D}(x_1, x_2)$, é infinito e não limitado. A Figura 15 ilustra as regiões no espaço dual que definem $\mathcal{D}(x_1, x_2)$, para cada ponto (x_1, x_2) .

As regiões em verde claro e escuro correspondem aos conjuntos de soluções duais. A região em verde escuro representa o conjunto de soluções duais obtido quando as variáveis de cópia são tratadas no domínio contínuo, ou seja, $\mathcal{D}^C(x_1, x_2)$, estando estritamente contido na região em verde claro, associada ao caso inteiro, ou seja, $\mathcal{D}^I(x_1, x_2)$, para os 4 valores de x_1 e x_2 , conforme a Proposição 3.

É possível constatar que as regiões correspondentes ao conjunto ótimo são ilimitadas em um dos sentidos, para cada dimensão, sendo que para $x_i = 0$ a região é ilimitada para $\mu_i \rightarrow -\infty$ e para $x_i = 1$ a região é ilimitada para $\mu_i \rightarrow +\infty$, conforme a Proposição 2.

De acordo com a mesma propriedade, os cortes de Lagrange mais fortes, isto é, aqueles com melhor aderência global à função de recurso, são obtidos nos pontos extremos (Definição 1) dessas regiões. Tais extremos estão destacados como pontos isolados (solução extrema trivial) ou por segmentos de reta (infinitas soluções extremas) indicados na Figura em verde claro, representação inteira, e em verde escuro, representação contínua da variável de cópia.

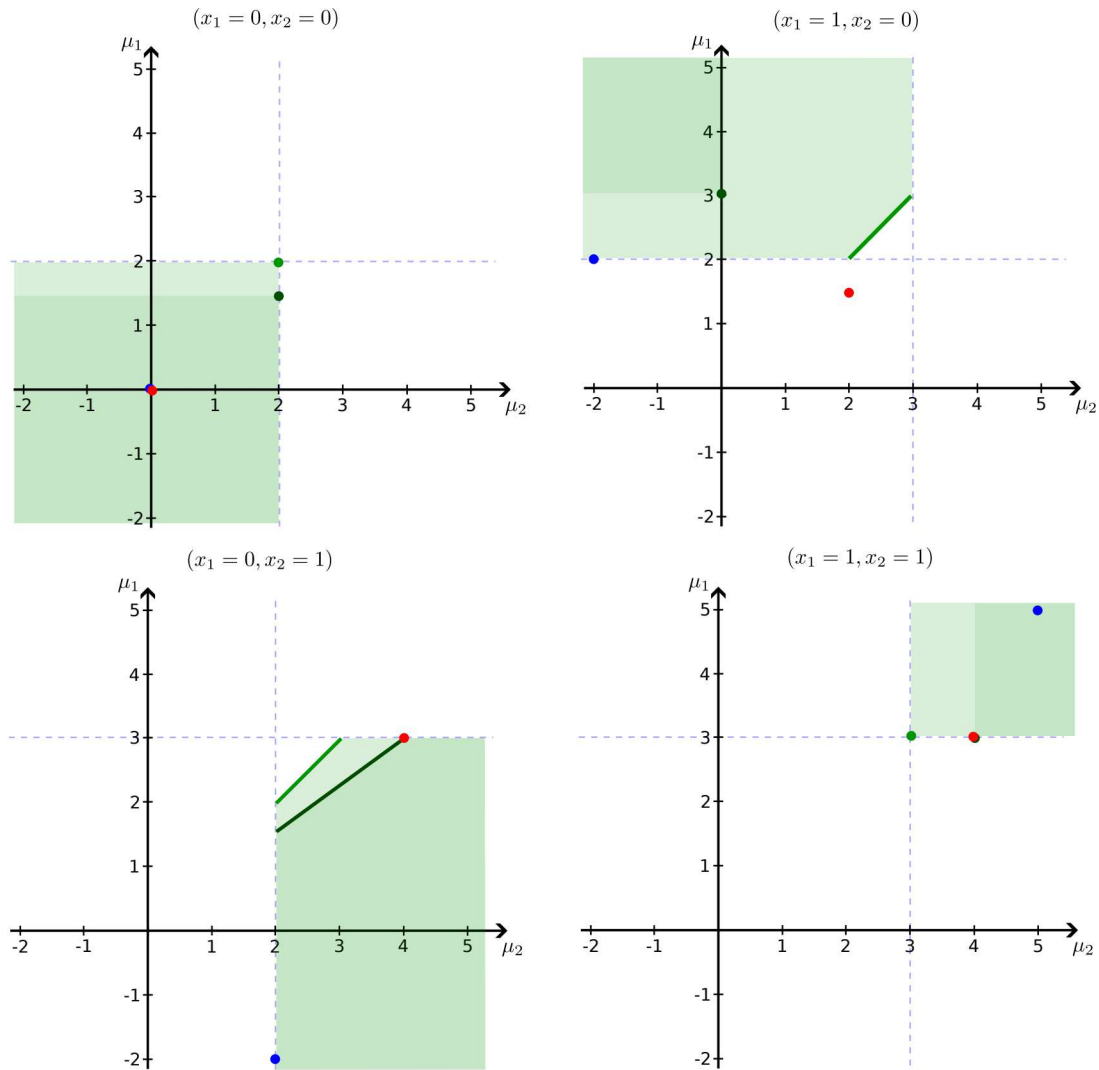


Figura 15 – Regiões de otimalidade das variáveis duais associadas aos problemas de maximização do Lagrangiano: (6.33), (6.34), (6.35) e (6.36).

Adicionalmente, são destacados dois tipos de soluções: (i) pontos vermelhos, que representam soluções do problema linearmente relaxado, podendo ou não pertencer ao conjunto ótimo, e (ii) pontos azuis, associados aos cortes inteiros de otimalidade (Seção 4.4) que sempre pertencem a $\mathcal{D}^I(x_1, x_2)$ (região verde clara), por isso geram cortes exatos, mas nem sempre pertencem a $\mathcal{D}^C(x_1, x_2)$ (região verde escura).

6.3.2 Análise no espaço primal

De modo a ilustrar como as soluções duais identificadas no espaço dual se traduzem em cortes de Lagrange no espaço primal, a Figura 16 apresenta, para as variáveis de recurso $(x_1 = 0, x_2 = 0)$, os cortes gerados no espaço primal (à direita) a partir dos respectivos coeficientes de Lagrange destacados no espaço dual (à esquerda).

Observa-se que o corte azul, associado ao dual $(\mu_1 = 0, \mu_2 = 0)$, embora seja válido e

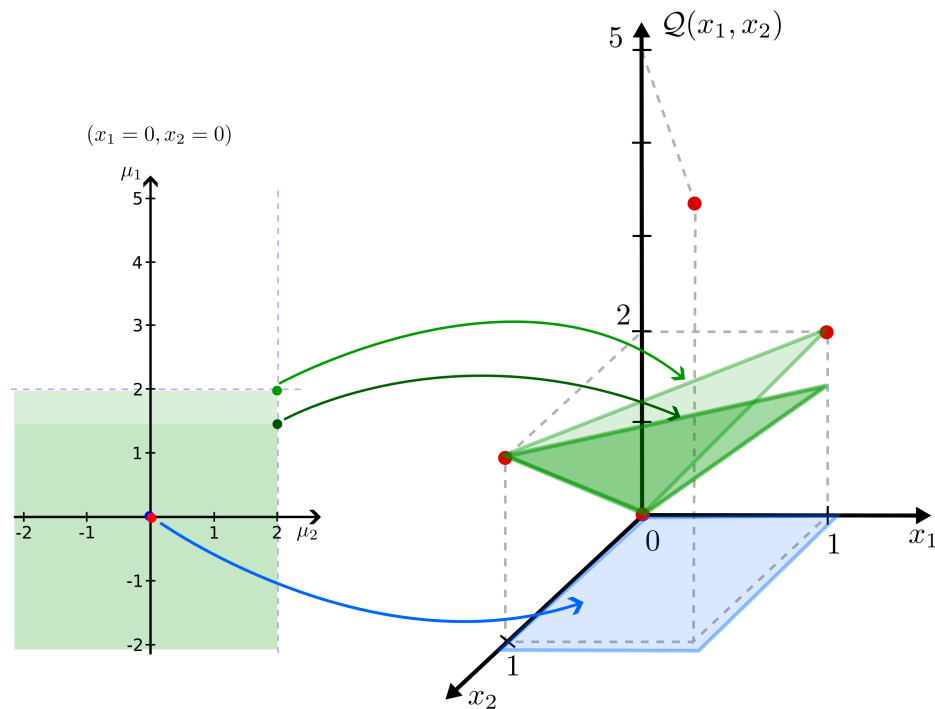


Figura 16 – Relação entre soluções no espaço dual e cortes de Lagrange no espaço primal para o ponto $(x_1 = 0, x_2 = 0)$

exato no ponto consultado, não fornece uma indicação adequada da característica de variação da função, conseqüentemente não aproxima satisfatoriamente a função de recurso nos demais pontos do domínio. Este corte corresponde ao corte de Benders, Benders Reforçados e Inteiros de Otimalidade construídos para esse ponto.

O corte em verde escuro, gerado a partir do dual $(\mu_1 = 1.5, \mu_2 = 2)$ que é extremo do conjunto $\mathcal{D}^C(x_1, x_2)$, tangencia a função de recurso não apenas no ponto consultado, mas também em $(x_1 = 0, x_2 = 1)$, resultando em uma aproximação mais acurada.

Por fim, o corte em verde claro, associado ao dual $(\mu_1 = 2, \mu_2 = 2)$ que é um ponto extremo do conjunto $\mathcal{D}^I(x_1, x_2)$, tangencia simultaneamente a função de recurso em três pontos. Como esse corte é exato no ponto consultado $(x_1 = 0, x_2 = 0)$, ele constitui o melhor corte de otimalidade que pode ser obtido nesse contexto.

Por sua vez, a Figura 17 ilustra a região de otimalidade no espaço dual quando as variáveis de recurso são $(x_1 = 1, x_2 = 0)$. Nesta situação, $\mathcal{D}^I(x_1, x_2)$ possui infinitos pontos extremos, que estão posicionados na aresta destacada em verde. São ilustrados, no espaço primal, três cortes de Lagrange construídos a partir de soluções pertencentes a esta aresta.

Os cortes associados aos duais indicados nos pontos (a) e (c) tangenciam simultaneamente três pontos do domínio; contudo, para isso, afastam-se do quarto ponto, resultando em uma aproximação menos equilibrada. Em contraste, o corte correspondente ao ponto (b), embora seja exato em apenas dois pontos do domínio, apresenta melhor aderência para os pontos onde o

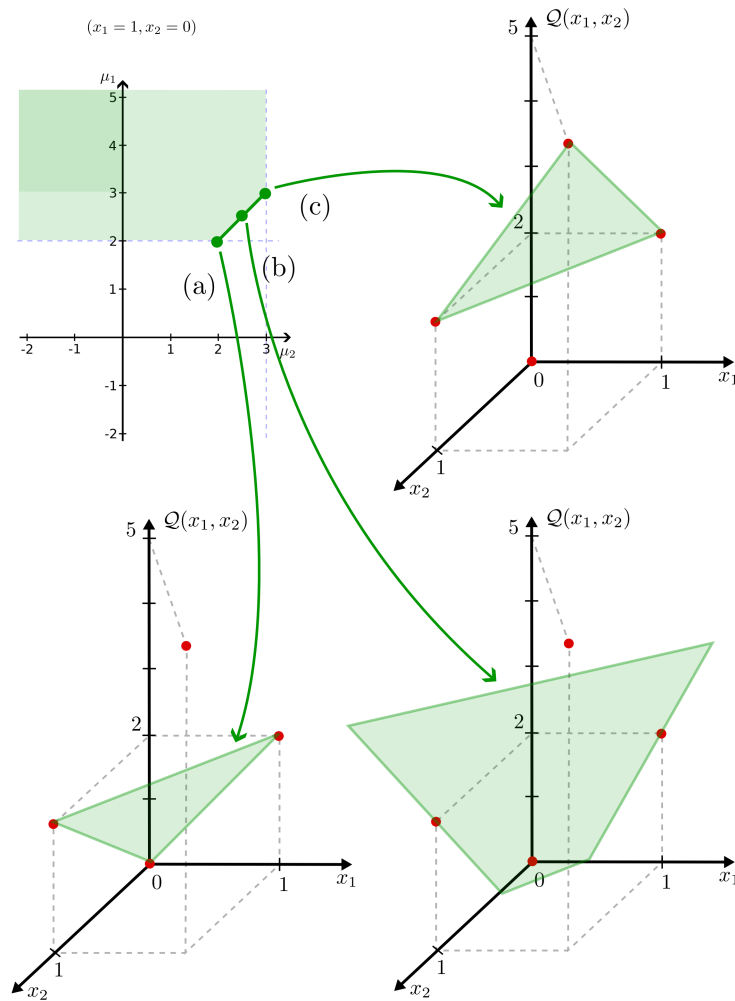


Figura 17 – Relação entre soluções no espaço dual e cortes de Lagrange no espaço primal para o ponto $(x_1 = 1, x_2 = 0)$

corte não é exato, aproximando-se de forma mais uniforme destes pontos. Em conclusão, todos os duais da região extrema de $\mathcal{D}^I(x_1, x_2)$ fornecem cortes de Lagrange que aproximam bem a função de recurso, porém com diferentes compromissos.

Adicionalmente, destacam-se:

- O dual do problema linearmente relaxado (pontos vermelhos na Figura 15) pertence ao conjunto de duais ótimos para o problema original para os casos $(x_1, x_2) \in \{(0, 0), (0, 1), (1, 1)\}$, mas não é um dual ótimo para a situação $(x_1, x_2) \in \{(1, 0)\}$. De fato, resolvendo o problema linearmente relaxado, encontramos $Q_{LR}(0, 0) = 0$, $Q_{LR}(1, 0) = 1.5$, $Q_{LR}(0, 1) = 2$ e $Q_{LR}(1, 1) = 5$. Dessa forma, observa-se que os cortes de Benders (Seção 4.3), construídos com base nestas soluções, produzem cortes válidos e exatos nas situações $(x_1, x_2) \in \{(0, 0), (0, 1), (1, 1)\}$, mas o corte será válido e frouxo (não exato) para a situação $(x_1, x_2) \in \{(1, 0)\}$.
- Os cortes Inteiros de Otimalidade (Seção 4.4) são válidos e exatos e sua inclinação sempre

corresponde a um dual ótimo para a restrição de cópia do problema $\mathcal{Q}(x)$, quando são consideradas variáveis de cópia binárias. No entanto, quando as variáveis de cópia são contínuas, isso pode não ser verdade, como por exemplo o ponto $(x_1 = 1, x_2 = 0)$ ilustrado na Figura 15.

6.4 Busca dos valores duais para construção dos cortes de Lagrange

A construção dos cortes de Lagrange depende da determinação das soluções duais $\mu_{t,s} \in \mathbb{R}^N$ associadas às restrições de cópia, obtidas pela resolução do problema (6.20).

O problema (6.20) é convexo, irrestrito e não diferenciável. Para sua resolução, podem ser empregados métodos clássicos da literatura, como o método do subgradiente [87], métodos de planos cortantes [88] e *bundle methods* [89], amplamente utilizados na determinação de soluções duais em problemas SDDiP e DDiP. Esses métodos serão discutidos na sequência.

Para simplificar a notação, omitem-se os índices t, s e o parâmetro $x_{t-1,s(a)}$, de modo que $L_{t,s}(\mu_{t,s}, x_{t-1,s(a)})$ passa a ser denotada por $L(\mu)$.

6.4.1 Método do Subgradiente

Nas aplicações de SDDiP, o método do Subgradiente [87] é amplamente utilizado para a determinação das variáveis duais e construção dos cortes de Lagrange [43, 50, 65, 71, 90].

O método consiste em iterativamente atualizar a solução dual na direção de crescimento da função, dada por um subgradiente, com o objetivo de encontrar um ponto para o qual o critério de parada estabelecido seja atingido. Seus principais passos são:

- Escolha de um ponto inicial: $\mu_0 \in \mathbb{R}^N$.
- Cálculo do tamanho do passo: $\alpha_k \in \mathbb{R}^+$.
- Verificação do critério de parada, idealmente: $0 \in \partial L(\mu_k)$.

Neste contexto, $\partial L(\mu_k) \subset \mathbb{R}^N$ corresponde ao subdiferencial da função L no ponto μ_k . Assim, a iteração do método é dada por:

$$\mu_{k+1} = \mu_k - \alpha_k \cdot g_k.$$

Nesta expressão, $g_k \in \partial L(\mu_k)$. O método é conhecido por realizar iterações computacionalmente pouco custosas, porém apresenta convergência lenta.

6.4.2 Método de Planos Cortantes

O método de planos cortantes [88] constrói aproximações afins superiores da função L , resultando em um modelo linear por partes que é refinado iterativamente. A cada iteração k , resolve-se o seguinte problema:

$$(\mu_k, \beta_k) = \arg \max_{\mu \in \mathbb{R}^N, \beta \in \mathbb{R}} \beta \quad (6.37)$$

$$s.t. \quad (6.38)$$

$$\beta \leq L(\mu_j) + g_j \cdot (\mu - \mu_j), \forall j = 0, \dots, k-1 \quad (6.39)$$

$$\beta \leq \bar{\beta} \quad (6.40)$$

$$\underline{\mu} \leq \mu \leq \bar{\mu} \quad (6.41)$$

Após a obtenção de μ_k , um novo plano de corte é definido a partir da avaliação de $L(\mu_k)$ e de um subgradiente $g_k \in \partial L(\mu_k)$.

Para a aplicação do método, são necessários:

- Um ponto inicial $\mu_0 \in \mathbb{R}^N$.
- Limites $\underline{\mu}, \bar{\mu} \in \mathbb{R}^N$ e $\bar{\beta} \in \mathbb{R}$.
- Um critério de parada, idealmente $0 \in \partial L(\mu_k)$.

Em comparação ao método do subgradiente, o método de planos cortantes apresenta maior custo computacional por iteração, pois requer a resolução de um problema de otimização linear. Em contrapartida, suas iterações incorporam informação acumulada das iterações anteriores, o que tende a melhorar a qualidade das aproximações nas etapas subsequentes.

6.4.3 Bundle methods

Os *bundle methods* são extensões do método de planos cortantes que incorporam informações de iterações anteriores para restringir o espaço de busca no domínio. Isso é feito por meio de termos de penalização ou restrições adicionais, o que favorece a exploração de regiões onde a função está melhor aproximada e reduz a instabilidade observada no método clássico [89].

Uma variante amplamente utilizada é o *bundle method proximal*, cuja iteração k é definida por:

$$(\mu_k, \beta_k) = \arg \max_{\mu \in \mathbb{R}^N, \beta \in \mathbb{R}} \beta - \frac{1}{2} \cdot u_k \cdot |\mu - \mu_i|^2, i \in \{1, k-1\} \quad (6.42)$$

$$s.t. \quad \beta \leq L(\mu_j) + g_j \cdot (\mu - \mu_j), \forall j = 0, \dots, k-1 \quad (6.43)$$

Onde u_k é o parâmetro de penalização que pode ser atualizado com regras simples. Outra alternativa são os *Level methods*, que controlam explicitamente o nível da função objetivo. Nesse caso, a iteração k é dada por:

$$(\mu_k, \beta_k) = \arg \min_{\mu \in \mathbb{R}^N, \beta \in \mathbb{R}} \frac{1}{2} \cdot |\mu - \mu_{k-1}|^2 \quad (6.44)$$

$$s.t. \beta \leq L(\mu_j) + g_j \cdot (\mu - \mu_j), \forall j = 0, \dots, k-1 \quad (6.45)$$

$$\beta \geq l_k \quad (6.46)$$

Assim, a cada iteração o algoritmo escolhe um valor de μ_k tal que minimize a distância ao μ_{k-1} , utilizado como referência, porém que garanta que o nível da função seja maior que l_k . Os valores de atualização de l_k podem seguir regras simples.

Esses métodos apresentam, em geral, convergência mais rápida e comportamento mais estável do que o método de planos cortantes. Por outro lado, cada iteração requer a resolução de um problema de otimização quadrática, o que pode aumentar o custo computacional.

7 Cortes de Lagrange e uma nova variante

Conforme discutido no Capítulo 6, a construção de cortes de Lagrange depende da obtenção de soluções duais ótimas do subproblema Lagrangiano, que, no contexto do SDDiP, formam um conjunto infinito e ilimitado, o que possibilita a geração de cortes com características bastante distintas.

A determinação desses coeficientes é um aspecto central na SDDiP, pois a metodologia adotada pode impactar significativamente a convergência. Por outro lado, esse processo é, em geral, computacionalmente oneroso, podendo comprometer a viabilidade prática do algoritmo. Este capítulo tem como objetivo discutir e classificar os tipos de cortes exatos para a SDDiP conforme a forma como os coeficientes são obtidos. Por fim, este capítulo propõe uma nova variante de corte denominada *Tighter Lagrangian Cuts*.

Ao longo deste capítulo, as abordagens sempre se aplicam a um subproblema específico da árvore, para determinado estágio t e cenário s , para o qual o corte está sendo construído. Assim, os índices t, s serão suprimidos para simplificar a notação.

7.1 Cortes de Lagrange Tradicionais

Os cortes de Lagrange tradicionais [50] são obtidos pela resolução do problema dual definido em (4.14), utilizando métodos gerais de otimização convexa irrestrita, como subgradiente, planos cortantes e métodos do tipo *Bundle*, descritos na Seção 6.4.

É importante ressaltar que definimos como cortes de Lagrange tradicionais aqueles que buscam uma solução dual arbitrária, sem apresentar uma tendência específica nessa busca, sendo essa a forma como os cortes são geralmente construídos na maioria das aplicações da SDDiP.

Devido à existência de múltiplas soluções duais ótimas, a qualidade dos cortes gerados pode variar, dependendo da posição da solução dual em relação à fronteira do conjunto ótimo. Consequentemente, a convergência do SDDiP com cortes tradicionais é sensível a: (i) características do problema; (ii) método empregado na resolução dual; e (iii) ponto inicial adotado.

Por outro lado, essa multiplicidade introduz propriedades úteis para a construção dos cortes, destacadas a seguir.

1. Derivada nula:

Como a função de Lagrange é constante sobre o conjunto de soluções duais ótimas, ou seja, para todo $\mu_1, \mu_2 \in \mathcal{D}(\bar{x}^{(a)})$:

$$L(\mu_1, \bar{x}^{(a)}) = L(\mu_2, \bar{x}^{(a)}),$$

onde $\mathcal{D}(\bar{x}^{(a)})$ denota o conjunto de soluções duais para um dado valor de recurso $\bar{x}^{(a)}$. Consequentemente:

$$0 \in \partial L(\bar{\mu}, \bar{x}^{(a)}) \quad \forall \bar{\mu} \in \mathcal{D}(\bar{x}^{(a)}).$$

Denota-se por $\partial L(\mu, x^{(a)})$ o superdiferencial da função côncava $L(\mu, x^{(a)})$ em μ . Assim, em particular,

$$\partial L(\bar{\mu}, \bar{x}^{(a)}) = \{0\} \quad \forall \bar{\mu} \in \text{ri}\mathcal{D}(\bar{x}^{(a)}),$$

onde ri denota o interior relativo do conjunto.

2. Valor ótimo conhecido:

Pela dualidade forte, o valor máximo de L é conhecido, isto é,

$$\max_{\mu} L(\mu, \bar{x}^{(a)}) = Q(\bar{x}^{(a)}).$$

O método do subgradiente [87] é particularmente adequado para a obtenção dos cortes de Lagrange tradicionais, pois este problema admite critérios de parada naturais, como:

$$\|g\| \leq \varepsilon \text{ ou } |Q(\bar{x}^{(a)}) - L(\mu, \bar{x}^{(a)})| \leq \varepsilon \quad (7.1)$$

Além disso, o conhecimento do valor ótimo (limite superior) permite o uso de passos eficientes, como o de Polyak:

$$\alpha_k = \frac{Q(\bar{x}^{(a)}) - L(\mu_k, \bar{x}^{(a)})}{\|g_k\|^2}.$$

Uma escolha natural para o ponto inicial é a solução dual do problema linearmente relaxado, isto é,

$$\mu_0 = \mu_{LR}. \quad (7.2)$$

Embora aplicações de SDDiP baseadas em subgradiente [43, 50, 65, 71, 90] apresentem poucos detalhes de implementação, acredita-se que seu uso é amplamente motivado pela simplicidade e pelas propriedades descritas acima.

Por outro lado, a aplicação do método de planos cortantes é menos direta devido ao fato de o conjunto de soluções ótimas ser ilimitado (Proposição 2). Como o método tende a explorar pontos extremos da região viável, ao longo do processo iterativo os limites $\underline{\mu}$ e $\bar{\mu}$ são frequentemente atingidos, o que pode levar à instabilidade numérica e ao mau condicionamento. Além disso, torna-se necessário arbitrar esses limites com valores suficientemente elevados, uma vez que a variável dual é ilimitada. Embora técnicas de regularização possam mitigar esse efeito, o método é, em geral, considerado menos estável [89].

Nesse contexto, métodos do tipo *Bundle* [89] apresentam desempenho mais robusto. Esses métodos também podem empregar o critério de parada (7.1) e o ponto inicial (7.2). Em aplicações de SDDiP [65, 50], o *Level method* foi utilizado para determinação dos cortes de Lagrange tradicionais e mostrou convergência mais rápida, compensando o maior custo por iteração e resultando em melhor desempenho global.

7.2 Square-minimization cuts (SMC)

Outro tipo de corte de Lagrange, proposto mais recentemente em [61] e [73], consiste em uma versão especializada dos cortes de Lagrange tradicionais. Nesse caso, o procedimento de obtenção da solução dual, além de buscar o máximo da função dual, também minimiza a soma quadrática dos componentes do vetor de variáveis duais. As Equações (7.3)-(7.4) descrevem o problema resolvido na construção dos cortes SMC, sendo que a restrição (7.4) garante a otimalidade dos valores duais em virtude da forte dualidade.

$$\min_{\mu} \mu^T \mu \quad (7.3)$$

$$s.t. \quad L(\mu, x^{(a)}) = Q(x^{(a)}) \quad (7.4)$$

Assim, a qualidade dos cortes SMC depende do ponto avaliado: a inclinação em cada dimensão será mais próxima ao extremo do conjunto quando o respectivo componente dual não admite zero como solução. Essas situações podem ser ilustradas com base na Figura 15:

- ponto $(x_1, x_2) = (0, 0)$: o SMC utilizaria a solução dual $(\mu_1, \mu_2) = (0, 0)$, que pertence ao interior da região de otimalidade, resultando em um corte fraco;
- ponto $(x_1, x_2) = (1, 1)$: como $(\mu_1, \mu_2) = (0, 0)$ não é uma solução dual ótima, o SMC determina a solução dual $(\mu_1, \mu_2) = (3, 3)$, correspondente a um ponto extremo da região de otimalidade (conforme a Definição 1), produzindo, assim, um corte forte.

Assim, cortes fortes e fracos necessariamente fazem parte do conjunto de cortes SMC, e esse método apresenta potencial para melhorar a convergência do SDDiP. Embora os autores não detalhem a metodologia nem as técnicas empregadas na resolução do problema (7.3)-(7.4), a obtenção desses cortes envolve a solução de um problema dual modificado, cujo conjunto de soluções é mais restrito, o que pode acarretar um aumento no custo computacional.

7.3 Cortes exatos não Lagrangianos

Alternativamente, cortes exatos podem ser computados sem a necessidade de resolver explicitamente o problema dual, ainda que seus coeficientes sejam soluções duais, uma vez que os cortes são exatos. Nessa classe de cortes estão:

- os cortes Inteiros de Otimalidade [48] têm coeficientes determinados por meio de uma expressão algébrica (ver Seção 4.4). São cortes extremamente simples de se computar, mas os coeficientes encontrados são em geral distantes da região limítrofe do conjunto dual ótimo, proporcionando cortes fracos.

- cortes *ReLU* [58] e os cortes escalados (*Scaled*) [56], quando aplicados a problemas com variáveis de estado binárias (como na SDDiP), produzem cortes exatos e, adicionalmente, buscam coeficientes que resultam em cortes fortes. Embora não exijam a resolução explícita do problema dual, a obtenção desses coeficientes envolve a reformulação e a resolução de uma sequência de problemas, o que também acarreta elevado custo computacional.

7.4 Nova variante de corte de Lagrange: *Tighter Lagrangian Cuts*

Este trabalho propõe um tipo de corte que pode ser classificado na primeira categoria de métodos, a qual resolve uma formulação dual do subproblema de MILP. Mais especificamente, busca-se selecionar uma solução dual específica na fronteira do conjunto dual ótimo. Para isso, os valores duais são obtidos por meio da maximização ou minimização de seus valores, dentro do conjunto de soluções que garantam a propriedade de dualidade forte (soluções ótimas). Isso pode ser feito pela resolução do problema (7.5)-(7.6).

$$\max \sum_{i=1}^N k_i \cdot \mu_i \quad (7.5)$$

$$\text{sujeito a } L(\mu, x^{(a)}) = Q(x^{(a)}), \quad (7.6)$$

em que:

$$k_i = \begin{cases} 1, & \text{se } x_i^{(a)} = 0, \\ -1, & \text{se } x_i^{(a)} = 1. \end{cases} \quad (7.7)$$

Como demonstrado em [50], a propriedade de dualidade forte $L(\mu^*, x^{(a)}) = Q(x^{(a)})$ é satisfeita, o que implica que qualquer solução μ para (7.5)-(7.6) é também uma solução dual para $Q(x^{(a)})$. Portanto, o *Tighter Lagrangian cut* é um corte de Lagrange, cujas propriedades de convergência para o SDDiP são estabelecidas em [50]. Adicionalmente, a combinação de sinais imposta por (7.7) garante a obtenção de cortes mais fortes, ou seja, associados a pontos extremos do conjunto dual ótimo, conforme a Propriedade 2.

No entanto, a resolução do problema (7.5)-(7.6) não é direta. Uma forma de tratá-lo é modificar a função de Lagrange por meio da soma de uma função linear com inclinação positiva ou negativa, de forma a se obter uma função modificada $\hat{L}(\mu, x^{(a)})$ cujo máximo esteja contido na fronteira da região ótima da função original, conforme ilustrado na Figura 18. Dessa forma, métodos de maximização irrestrita, como subgradiente, planos cortantes e *Bundle* são aplicados à função modificada, resolvendo o seguinte problema:

$$\max_{\mu} \hat{L}(\mu, x^{(a)}), \quad (7.8)$$

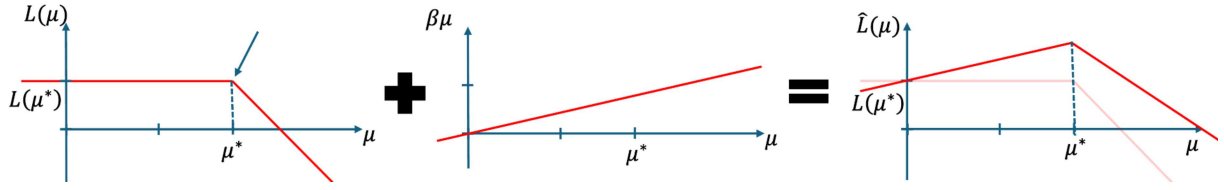


Figura 18 – Modificação da função de Lagrange para obtenção da solução dual na fronteira do conjunto ótimo

onde:

$$\hat{L}(\mu, x^{(a)}) = L(\mu, x^{(a)}) - \beta^\top \mu. \quad (7.9)$$

A Proposição 4 mostra que é possível obter a função $\hat{L}(\mu, x^{(a)})$ com essas propriedades se a escolha do parâmetro β for adequada.

Proposição 4. *Seja $L(\mu, x^{(a)})$, para um dado $x^{(a)}$, a função de Lagrange definida por (6.18)-(6.19). Suponha que $\text{Ext}(\mathcal{D}(x^{(a)})) \neq \emptyset$. Então existe $\beta \in \mathbb{R}^N$ tal que, definindo:*

$$\hat{L}(\mu, x^{(a)}) = L(\mu, x^{(a)}) - \beta^\top \mu,$$

o conjunto de soluções para o problema:

$$(\hat{P}) \quad \max \hat{L}(\mu, x^{(a)}),$$

denotado por $\hat{\mathcal{D}}(x^{(a)})$, satisfaz:

$$\hat{\mathcal{D}}(x^{(a)}) \subseteq \text{Ext}(\mathcal{D}(x^{(a)})) \quad e \quad \hat{\mathcal{D}}(x^{(a)}) \neq \emptyset.$$

Ou seja, $(\hat{P})$ admite solução e todas as suas soluções são pontos extremos de $\mathcal{D}(x^{(a)})$, conforme a Definição 1.

Demonstração. A validade da proposição decorre da verificação das afirmações seguintes, que serão demonstradas individualmente na sequência.

1. **Existe $\mu^* \in \text{Ext}(\mathcal{D}(x^{(a)}))$ tal que $\mu^* \in \hat{\mathcal{D}}(x^{(a)})$.**

Seja

$$\mu^* \in \text{Ext}(\mathcal{D}(x^{(a)})).$$

Pela Definição 1, para todo vetor $\Delta \in \mathbb{R}^N$ que satisfaça (6.24) e (6.25), tem-se

$$\mu^* + \Delta \notin \mathcal{D}(x^{(a)}).$$

E, como $\mu^* \in \mathcal{D}(x^{(a)})$, tem-se

$$L(\mu^*, x^{(a)}) > L(\mu^* + \Delta, x^{(a)}).$$

Além disso, como $L(\cdot, x^{(a)})$ é uma função côncava e poliedral, e μ^* está localizado na fronteira da região plana de máximo, em um ponto de não diferenciabilidade, fixando Δ tal que $|\Delta_i| > 0$ para todo i e suficientemente pequeno, existe uma face inclinada adjacente à região ótima na direção definida por Δ , à qual pertence o ponto $\mu^* + \Delta$. Logo, existe um vetor do superdiferencial de $L(\cdot, x^{(a)})$ associado a ela. Em particular, existe $\alpha \in \partial L(\mu^*, x^{(a)})$ tal que, para cada componente i ,

$$\begin{cases} \alpha_i < 0, & \text{se } x_i^{(a)} = 0, \\ \alpha_i > 0, & \text{se } x_i^{(a)} = 1. \end{cases} \quad (7.10)$$

Como μ^* é uma solução dual ótima, tem-se também que

$$0 \in \partial L(\mu^*, x^{(a)}) \text{ e } \alpha \in \partial L(\mu^*, x^{(a)}).$$

Pela convexidade do superdiferencial de uma função côncava, segue que

$$\lambda \alpha \in \partial L(\mu^*, x^{(a)}), \quad \forall \lambda \in [0, 1].$$

Toma-se $\beta \in \mathbb{R}^N$, tal que $\beta = \lambda' \alpha$ para um valor de $\lambda' \in (0, 1)$. Então:

$$\beta \in \partial L(\mu^*, x^{(a)}),$$

logo:

$$0 \in \partial L(\mu^*, x^{(a)}) - \beta.$$

Pela definição de $\hat{L}(\mu^*, x^{(a)})$, o superdiferencial desta função é dado por:

$$\partial \hat{L}(\mu^*, x^{(a)}) = \partial L(\mu^*, x^{(a)}) - \beta,$$

logo:

$$0 \in \partial \hat{L}(\mu^*, x^{(a)}).$$

Como $\hat{L}(\cdot, x^{(a)})$ é côncava, então μ^* é um maximizador global de $\hat{L}(\cdot, x^{(a)})$. Logo,

$$\mu^* \in \hat{\mathcal{D}}(x^{(a)}).$$

2. Se $\bar{\mu} \notin \mathcal{D}(x^{(a)})$ então $\bar{\mu} \notin \hat{\mathcal{D}}(x^{(a)})$.

Considere o ponto μ^* e o vetor β construído na etapa anterior, é preciso mostrar que:

$$\hat{L}(\bar{\mu}, x^{(a)}) < \hat{L}(\mu^*, x^{(a)}) \quad \forall \bar{\mu} \notin \mathcal{D}(x^{(a)})$$

O que é equivalente a:

$$L(\bar{\mu}, x^{(a)}) < L(\mu^*, x^{(a)}) + \beta^\top (\bar{\mu} - \mu^*) \quad \forall \bar{\mu} \notin \mathcal{D}(x^{(a)}) \quad (7.11)$$

Da desigualdade fundamental do supergradiente de uma função côncava, como $\beta, \alpha \in \partial L(\mu^*, x^{(a)})$, tem-se que:

$$L(\bar{\mu}, x^{(a)}) \leq L(\mu^*, x^{(a)}) + \alpha^\top (\bar{\mu} - \mu^*) \quad (7.12)$$

$$L(\bar{\mu}, x^{(a)}) \leq L(\mu^*, x^{(a)}) + \beta^\top (\bar{\mu} - \mu^*) \quad (7.13)$$

Para mostrar que a Equação (7.13) deve ser atendida estritamente, resultando em (7.11), por absurdo suponha que $\exists \bar{\mu} \notin \mathcal{D}(x^{(a)})$ tal que:

$$L(\bar{\mu}, x^{(a)}) = L(\mu^*, x^{(a)}) + \beta^\top (\bar{\mu} - \mu^*) \quad (7.14)$$

Substituindo (7.14) em (7.12), tem-se:

$$L(\mu^*, x^{(a)}) + \beta^\top (\bar{\mu} - \mu^*) \leq L(\mu^*, x^{(a)}) + \alpha^\top (\bar{\mu} - \mu^*)$$

Como $\beta = \lambda' \alpha$:

$$L(\mu^*, x^{(a)}) + \beta^\top (\bar{\mu} - \mu^*) \leq L(\mu^*, x^{(a)}) + \frac{1}{\lambda'} \beta^\top (\bar{\mu} - \mu^*)$$

$$\beta^\top (\bar{\mu} - \mu^*) \leq \frac{1}{\lambda'} \beta^\top (\bar{\mu} - \mu^*)$$

$$(1 - \frac{1}{\lambda'}) \beta^\top (\bar{\mu} - \mu^*) \leq 0 \quad (7.15)$$

Como temos que $\lambda' \in (0, 1)$, segue que:

$$(1 - \frac{1}{\lambda'}) < 0 \quad (7.16)$$

Pela definição de $\bar{\mu}$ e μ^* :

$$L(\bar{\mu}, x^{(a)}) < L(\mu^*, x^{(a)})$$

$$L(\bar{\mu}, x^{(a)}) - L(\mu^*, x^{(a)}) < 0$$

Usando esta desigualdade estrita em (7.14), tem-se:

$$\beta^\top (\bar{\mu} - \mu^*) < 0 \quad (7.17)$$

Como os dois fatores em (7.15) são estritamente negativos, o produto é estritamente positivo, contradizendo (7.15). Assim:

$$\hat{L}(\bar{\mu}, x^{(a)}) < \hat{L}(\mu^*, x^{(a)}),$$

e isso é válido para todo $\bar{\mu} \notin \mathcal{D}(x^{(a)})$.

3. Se $\tilde{\mu} \in \mathcal{D}(x^{(a)})$ e $\tilde{\mu} \notin \text{Ext}(\mathcal{D}(x^{(a)}))$ então $\tilde{\mu} \notin \hat{\mathcal{D}}(x^{(a)})$.

Pela Definição 1, $\exists \Delta$, denotado $\tilde{\Delta}$, tal que:

$$\tilde{\mu} + \tilde{\Delta} \in \mathcal{D}(x^{(a)}),$$

o que leva à:

$$L(\tilde{\mu}, x^{(a)}) = L(\tilde{\mu} + \tilde{\Delta}, x^{(a)}). \quad (7.18)$$

Ao mesmo tempo, pela regra de formação de β em (7.10) e de $\tilde{\Delta}$ em (6.24), tem-se:

$$\beta^\top \tilde{\Delta} < 0. \quad (7.19)$$

Pela definição da função $\hat{L}(\mu, x^{(a)})$ no enunciado desta proposição, temos:

$$\hat{L}(\tilde{\mu} + \tilde{\Delta}, x^{(a)}) = L(\tilde{\mu} + \tilde{\Delta}, x^{(a)}) - \beta^\top (\tilde{\mu} + \tilde{\Delta}). \quad (7.20)$$

Substituindo (7.18) em (7.20), obtemos:

$$\begin{aligned} \hat{L}(\tilde{\mu} + \tilde{\Delta}, x^{(a)}) &= L(\tilde{\mu}, x^{(a)}) - \beta^\top (\tilde{\mu} + \tilde{\Delta}), \\ \hat{L}(\tilde{\mu} + \tilde{\Delta}, x^{(a)}) &= \hat{L}(\tilde{\mu}, x^{(a)}) - \beta^\top \tilde{\Delta}. \end{aligned} \quad (7.21)$$

Devido à (7.19), a igualdade (7.21) implica:

$$\hat{L}(\tilde{\mu} + \tilde{\Delta}, x^{(a)}) > \hat{L}(\tilde{\mu}, x^{(a)}).$$

O que, por sua vez, implica $\tilde{\mu} \notin \hat{\mathcal{D}}(x^{(a)})$.

Portanto, pelas três etapas anteriores, toda solução ótima de $(\hat{P})$ pertence a $\mathcal{D}(x^{(a)})$ e é necessariamente um ponto extremo desse conjunto. Logo,

$$\hat{\mathcal{D}}(x^{(a)}) \subseteq \text{Ext}(\mathcal{D}(x^{(a)})).$$

Além disso, pela primeira etapa, $\hat{\mathcal{D}}(x^{(a)}) \neq \emptyset$. Isso conclui a demonstração. □

No entanto, a maximização da função de Lagrange modificada ($\hat{L}(\mu, x^{(a)})$) para determinação dos *Tighter Lagrangian Cuts*, pode ser mais desafiadora do que a maximização da função de Lagrange original ($L(\mu, x^{(a)})$) para obtenção dos cortes de Lagrange Tradicionais. Isso porque as características 1 e 2 observadas na Seção 7.1 (Derivada nula e Valor ótimo conhecido) podem não ser válidas para o problema modificado. No entanto, os critérios de parada (7.1) ainda podem ser aplicados consultando a função original.

O impacto do uso da função modificada nos métodos citados anteriormente para maximização da função dual é descrito abaixo:

1. **subgradiente**: a utilização deste método no problema modificado fica comprometida, pois a ausência de um limite superior conhecido para a função $\hat{L}(\mu, x^{(a)})$ impede a utilização de tamanhos de passos como o de Polyak.
2. **planos cortantes**: com a função modificada, o conjunto de soluções não é mais ilimitado, o que facilita a aplicação deste método, em comparação à sua aplicação na função original.
3. **Bundle**: podem ser aplicados tanto na função original quanto na função modificada, utilizando os mesmos critérios.

7.5 Comparação entre cortes lineares Lagrangianos e não-Lagrangianos

A Tabela 30 compara os *Tighter Lagrangian cuts* propostos com os demais cortes exatos da literatura. Adicionalmente, são comparados os cortes não exatos mais comumente utilizados para solução de MS-MILPs via DDiP ou SDDiP. A análise considera a complexidade de implementação e de cálculo de cada tipo de corte, bem como sua propriedade de exatidão (*tightness*). Por fim, os cortes são classificados como “fracos” ou “fortes”. Algumas abordagens produzem simultaneamente cortes fracos e fortes, enquanto outras tendem a gerar predominantemente um ou outro, dependendo das características do problema e da estratégia de busca dual adotada, incluindo sua inicialização.

Como evidenciado na tabela, nosso objetivo é alcançar um equilíbrio adequado entre a qualidade dos cortes e o esforço computacional, tanto do ponto de vista de implementação quanto de desempenho computacional.

Classe	Corte	Exato	Característica do corte	Complexidade computacional	Complexidade de implementação
Cortes de Lagrange	Tradicionais	✓	fraco ou forte	baixa	média
	SMC	✓	fraco e forte	média	média
	<i>Tighter</i>	✓	forte	média	média
Cortes não Lagrangianos	ReLU	✓	fraco ou forte	baixa	alta
	Scaled	✓	forte	alta	alta
	Int. de Otimalidade	✓	fraco	baixíssima	baixíssima
Cortes não exatos	Benders	×	fraco ou forte	baixíssima	baixíssima
	Bend.Reforçados	×	fraco ou forte	baixa	baixíssima

Tabela 30 – Comparação entre cortes lineares utilizados nos métodos da SDDiP e DDiP.

8 Avaliação dos cortes de Lagrange

Nos Capítulos 6 e 7 demonstrou-se que os cortes de Lagrange utilizados nos algoritmos DDiP e SDDiP não são únicos, uma vez que o problema dual associado admite múltiplas soluções ótimas. Também foi discutido, sob uma perspectiva teórica, como diferentes escolhas de multiplicadores duais podem resultar em cortes com características distintas e, conseqüentemente, impactar o desempenho do algoritmo. Neste capítulo, tais questões são investigadas por meio de experimentos computacionais, com o objetivo de avaliar a influência do tipo de corte, do método empregado para obtenção das soluções duais e da escolha do ponto inicial de busca sobre a convergência da DDiP. Adicionalmente, é analisado o efeito da utilização de variáveis de cópia contínuas ou inteiras na construção do problema dual e na qualidade dos cortes de Lagrange gerados.

Todos os experimentos computacionais foram conduzidos em um computador equipado com processador Intel i7 de 11ª geração, contendo 8 núcleos de processamento e 32 GB de memória RAM. Para a solução dos problemas de otimização, foi utilizado o solver IBM CPLEX, configurado para execução com dois *threads*. Os problemas lineares inteiros mistos foram resolvidos por meio do algoritmo *branch-and-bound*, enquanto os problemas lineares foram solucionados pelo método *simplex*. Os procedimentos de subgradiente e de geração de planos cortantes foram implementados no contexto desse trabalho e integrados à implementação em C++ da SDDiP/DDiP. Já o método *Bundle* utilizado nos experimentos foi obtido a partir de uma biblioteca amplamente reconhecida na literatura especializada, desenvolvida pela Profa. Claudia Sagastizábal.

8.1 Estudo de caso

Para os experimentos computacionais, foi considerado o caso “reduced-system-cond-constr”, descrito na Seção 5.2, cujos os resultados gerais são apresentados na Tabela 28. Como o objetivo deste capítulo é avaliar o desempenho de cortes exatos, em particular, variantes dos cortes de Lagrange, adotou-se uma representação com expansão binária reduzida, viabilizando a obtenção de tais resultados. Conforme indicado na Seção 5.2 (Tabela 29), o uso de 5 variáveis binárias apresenta um compromisso adequado entre precisão e custo computacional. Essa configuração foi, portanto, adotada nas análises deste capítulo.

8.2 Análise comparativa e sensibilidade

São comparados os cortes exatos cuja complexidade de implementação e de cálculo é considerada média, baixa ou baixíssima, ou seja, os cortes Tradicionais, SMC, Inteiros de Otimidade e *Tighter*. Em teoria, todos esses cortes podem levar à solução ótima via DDiP.

Contudo, conforme discutido na Seção 5.2, os cortes Inteiros de Otimalidade não foram capazes de resolver o problema em tempo computacional razoável, sendo, portanto, excluídos da análise.

Assim, foram realizadas análises de sensibilidade em relação ao método de busca dos duais para construção dos cortes, ponto inicial fornecido à esse método e tipo de variável de cópia considerada, conforme listado a seguir:

1. Método de busca da solução dual:

- cortes de Lagrange **Tradicionais** obtendo as variáveis duais pelo método do **Subgradiente**, amplamente utilizado na literatura para esta variante [50, 65, 43, 71, 90];
- cortes de Lagrange **Tradicionais** obtendo as variáveis duais pelo método **Bundle**;
- cortes **SMC** [61] obtendo as variáveis duais pelo método de planos cortantes (**PC**);
- cortes **Tighter** obtendo as variáveis duais pelo método de planos cortantes (**PC**);
- cortes **Tighter** obtendo as variáveis duais pelo método **Bundle**;

2. Ponto inicial fornecido ao método de busca da solução dual:

- vetor nulo ($\mu_0 = 0$);
- solução dual da relaxação linear do problema ($\mu_0 = \mu_{LR}$);
- inclinações dos cortes Inteiros de Otimalidade ($\mu_0 = \mu_{IO}$).

3. Variável de cópia:

- variável de cópia contínua;
- variável de cópia inteira.

8.3 Resultados

As Tabelas 31 e 32 mostram, para as variáveis de cópia contínuas e inteiras, respectivamente, o *gap* de convergência da DDiP, o número de iterações necessárias para atingir este *gap*, o tempo computacional e o tempo por iteração para cada uma das sensibilidades analisadas.

Os Cortes de Lagrange Tradicionais mostraram-se altamente sensíveis à escolha do ponto inicial, tanto no caso de variáveis cópia inteiras quanto contínuas. Esse comportamento decorre do fato de que a solução dual ótima obtida pelo método de busca dual depende do ponto de partida adotado, resultando na geração de cortes distintos. Dependendo dos multiplicadores duais

	Ponto inicial busca dual	Gap (%)	Numero de iterações	Tempo (s)	Tempo por iteração (s)
Tradicionais (Subgradiente)	0	0.0	235	1938	9.0
Tradicionais (Subgradiente)	μ_{LR}	0.0	12	53	5.1
Tradicionais (Subgradiente)	μ_{IO}	10537*	354*	2148*	6.6
Tradicionais (<i>Bundle</i>)	0	0.0	356	14574	41.0
Tradicionais (<i>Bundle</i>)	μ_{LR}	0.0	39	253	6.5
Tradicionais (<i>Bundle</i>)	μ_{IO}	10584*	357*	3590*	10.0
SMC (PC)	0	0.0	28	610	21.8
SMC (PC)	μ_{LR}	0.0	28	565	20.2
SMC (PC)	μ_{IO}	0.0	28	573	20.5
<i>Tighter Cuts</i> (PC)	0	0.0	9	900	100.0
<i>Tighter Cuts</i> (PC)	μ_{LR}	0.0	9	905	100.5
<i>Tighter Cuts</i> (PC)	μ_{IO}	0.0	9	1012	112.4
<i>Tighter Cuts</i> (<i>Bundle</i>)	0	0.0	8	633	79.1
<i>Tighter Cuts</i> (<i>Bundle</i>)	μ_{LR}	0.0	7	223	31.8
<i>Tighter Cuts</i> (<i>Bundle</i>)	μ_{IO}	0.0	14	1340	95.7

Tabela 31 – Resultados da DDiP utilizando diferentes tipos de cortes Lagrangianos com variável de cópia contínua e a busca dual com diferentes pontos iniciais (* execução interrompida devido ao excesso de tempo).

selecionados, os cortes produzidos podem ser classificados como “fortes”, quando associados a multiplicadores localizados próximos à fronteira do conjunto de soluções duais ótimas, ou como “fracos”, quando associados a multiplicadores situados em regiões mais centrais desse conjunto.

Embora os resultados tenham sido satisfatórios quando se utilizou como ponto inicial a solução dual do problema linearmente relaxado, não há garantia de que essa estratégia conduza sistematicamente à geração de cortes fortes. Em outras palavras, não é possível assegurar que o multiplicador de Lagrange obtido esteja próximo da fronteira do conjunto de soluções ótimas. Como consequência, a qualidade dos cortes gerados pode variar significativamente de acordo com as características do problema analisado.

	Ponto inicial busca dual	Gap (%)	Numero de iterações	Tempo (s)	Tempo por iteração (s)
Tradicionais (Subgradiente)	0	0.0	279	13301	47.7
Tradicionais (Subgradiente)	μ_{LR}	0.0	94	625	156.2
Tradicionais (Subgradiente)	μ_{IO}	10537*	501*	1445*	2.8
SMC (PC)	0	0.0	19	1392	73.3
SMC (PC)	μ_{LR}	0.0	18	663	36.8
SMC (PC)	μ_{IO}	0.0	17	1222	71.9
<i>Tighter Cuts</i> (PC)	0	0.0	5	2937	587.4
<i>Tighter Cuts</i> (PC)	μ_{LR}	0.0	5	3196	639.2
<i>Tighter Cuts</i> (PC)	μ_{IO}	0.0	5	2776	555.2

Tabela 32 – Resultados da DDiP utilizando diferentes tipos de cortes Lagrangianos com variável de cópia inteira e a busca dual com diferentes pontos iniciais (* execução interrompida devido ao excesso de tempo).

No caso estudado, os resultados sugerem que a boa qualidade dos cortes gerados a partir desse ponto inicial está associada à similaridade entre a função de recurso do problema linearmente relaxado e a função de recurso original. Nessa situação, a solução dual do problema relaxado tende a fornecer uma aproximação adequada da estrutura dual do problema original, favorecendo a obtenção de multiplicadores localizados em regiões associadas à geração de cortes mais fortes. Assim, concluímos que o desempenho dos Cortes de Lagrange Tradicionais está fortemente condicionado à representatividade da solução dual utilizada para inicializar o processo de busca. Consequentemente, essa abordagem tende a apresentar desempenho consistente principalmente em situações nas quais o método de Benders clássico já produz aproximações de boa qualidade para a função de recurso.

Os cortes SMC mostraram-se pouco sensíveis ao ponto inicial empregado pelo método de busca no que diz respeito à qualidade dos cortes gerados. Como consequência, o número de iterações necessárias para a convergência da DDiP permaneceu relativamente estável entre os diferentes pontos iniciais analisados. Embora esse número tenha sido superior ao obtido com os Cortes de Lagrange Tradicionais quando inicializados com $\mu_0 = \mu_{LR}$, foi significativamente inferior ao observado para os demais pontos iniciais considerados. Esse comportamento corrobora as conclusões apresentadas na Seção 7.2 e os resultados da Tabela 30, indicando que a metodologia SMC é capaz de produzir cortes “fortes” e “fracos” simultaneamente, independentemente da

solução dual utilizada para inicialização do procedimento de busca.

Os *Tighter Lagrangian Cuts*, por sua vez, também não apresentaram sensibilidade ao ponto inicial adotado. Além disso, foi o único método capaz de conduzir a DDiP à convergência com o menor número de iterações em todos os casos avaliados. Esse resultado sugere que os cortes gerados são efetivamente mais fortes, fornecendo uma aproximação mais precisa da função de recurso ao longo de todo o domínio de interesse. Entretanto, tal ganho em qualidade foi acompanhado de um aumento significativo no tempo computacional. Isso ocorre porque a construção desses cortes exige a obtenção de soluções duais com características específicas, processo substancialmente mais custoso do que a simples determinação de uma solução dual ótima arbitrária.

Por fim, a comparação entre as formulações com variáveis cópia inteiras e contínuas permite destacar dois aspectos principais:

- O número de iterações da DDiP necessárias para atingir a convergência foi, de maneira geral, menor quando se utilizaram variáveis cópia inteiras. Esse resultado sugere que, independentemente da variante considerada, os cortes de Lagrange gerados nesse contexto tendem a apresentar maior qualidade, fornecendo aproximações mais precisas da função de recurso. Tal comportamento está em consonância com a análise conceitual desenvolvida na Seção 6.2;
- Em contrapartida, o custo computacional associado à geração desses cortes foi significativamente superior. Isso ocorre porque a presença de variáveis cópia inteiras torna o problema dual mais complexo e, conseqüentemente, mais difícil de resolver. Dessa forma, embora a redução do número de iterações contribua para acelerar o processo de convergência da DDiP, esse benefício pode ser parcialmente ou totalmente compensado pelo aumento do esforço computacional necessário para a construção dos cortes.

Com relação aos métodos de busca dual, observou-se que, para os Cortes de Lagrange Tradicionais, o método do Subgradiente apresentou, em geral, menor tempo computacional do que o método *Bundle*. Conforme discutido anteriormente, a estrutura particular do problema analisado oferece condições favoráveis à aplicação do método do Subgradiente, permitindo a obtenção de boas soluções com reduzido esforço computacional. Por outro lado, embora o método *Bundle* tenha apresentado desempenho semelhante quando empregado um ponto inicial favorável, $\mu_0 = \mu_{LR}$, seu desempenho deteriorou-se para pontos iniciais mais distantes da solução desejada. Esse comportamento sugere que o seu desempenho pode melhorar com ajustes nos parâmetros que controlam o raio ou a região de busca utilizada durante o processo iterativo.

No caso dos cortes SMC e dos *Tighter Lagrangian Cuts* calculados por meio do método de Plano de Cortes (PC), observou-se um comportamento mais estável e previsível. Em particular, o método PC apresentou baixa sensibilidade ao ponto inicial, tanto em termos do número de

iterações quanto do tempo computacional. Tal característica decorre do fato de que, já nas primeiras iterações, o método consulta pontos relativamente distantes da solução inicial, fazendo com que a influência dessa informação seja rapidamente reduzida ao longo do procedimento.

Quando o método *Bundle* foi aplicado aos *Tighter Lagrangian Cuts*, verificou-se uma redução do número de iterações quando utilizado $\mu_0 = \mu_{LR}$, acompanhada por uma diminuição significativa do tempo total de processamento. Esses resultados indicam que, para cortes mais estruturados, a utilização de um método dual mais sofisticado pode compensar o maior custo computacional por iteração, resultando em um desempenho global superior.

Comparando os diferentes métodos de busca dual, pode-se concluir que:

- o método do Subgradiente apresenta elevada eficiência quando é possível definir adequadamente o tamanho de passo e os critérios de parada. Nessas condições, o método tende a produzir soluções de boa qualidade com baixo custo computacional, embora apresente certa sensibilidade ao ponto inicial adotado;
- o método de Planos de Corte é significativamente menos sensível ao ponto inicial, produzindo um comportamento mais consistente entre diferentes execuções. Em contrapartida, cada iteração é computacionalmente mais onerosa. Ainda assim, esse método mostrou-se particularmente adequado para a construção dos cortes SMC, cuja obtenção está associada à busca de soluções duais de menor magnitude;
- o método *Bundle*, apesar de apresentar maior custo computacional por iteração, mostrou-se mais eficiente que o método de Planos de Cortes na obtenção dos *Tighter Lagrangian Cuts*, reduzindo, de forma geral, o tempo total de processamento. Contudo, os resultados também indicam que seu desempenho é mais dependente da escolha do ponto inicial.

De forma geral, os resultados deste capítulo evidenciam que o desempenho da DDiP/SDDiP é determinado pela interação entre o tipo de corte de Lagrange empregado, o método de busca dual utilizado e a escolha do ponto inicial. Métodos mais robustos do ponto de vista teórico, como os *Tighter Cuts*, de fato produzem cortes mais fortes e reduzem significativamente o número de iterações, mas a um custo computacional elevado por iteração. Por outro lado, abordagens mais simples, como os cortes Tradicionais, podem apresentar bom desempenho apenas em condições favoráveis, mas são altamente sensíveis à parametrização. Por fim, a escolha do método mais adequado depende do balanço desejado entre tempo total de processamento e robustez da convergência. Em problemas de maior escala ou com maior complexidade estrutural, os resultados sugerem que o investimento em cortes mais fortes e métodos de busca dual mais sofisticados pode ser justificável, desde que acompanhado de implementações eficientes.

9 Estudo de casos de grande porte

Os estudos apresentados nos capítulos anteriores foram conduzidos em problemas de pequena escala, permitindo a análise detalhada das propriedades dos diferentes cortes e métodos de decomposição. Entretanto, uma das motivações desta tese está associada à resolução de problemas de planejamento hidrotérmico de grande porte, cujas dimensões são compatíveis com aplicações reais.

Neste capítulo, avalia-se o desempenho dos métodos estudados em um modelo baseado no sistema elétrico brasileiro, contemplando todas as usinas hidrelétricas e termelétricas, bem como um conjunto representativo de restrições hidráulicas e elétricas. Inicialmente, são apresentados a descrição do caso e os resultados obtidos pela resolução monolítica do problema, utilizados como referência para comparação.

Em seguida, analisa-se o comportamento da decomposição via DDP em problemas lineares e em problemas contendo restrições condicionais. Posteriormente, avalia-se a aplicação da versão exata da DDiP utilizando cortes de Lagrange, destacando as limitações computacionais observadas. Por fim, investiga-se o impacto da expansão binária sobre o esforço computacional dos subproblemas, fornecendo subsídios para a proposta metodológica apresentada no capítulo seguinte.

9.1 Descrição do caso

O caso de grande porte considerado neste trabalho é baseado no sistema elétrico brasileiro e contempla todas as usinas hidrelétricas e termelétricas, bem como diversas restrições elétricas e hidráulicas representadas no planejamento da operação realizado pelo Operador Nacional do Sistema Elétrico (ONS). As principais características e dimensões desse caso são apresentadas na Tabela 33.

Como todas essas características são modeladas por variáveis contínuas e restrições lineares, cada subproblema da árvore de cenários corresponde a um problema de programação linear (LP), associado a um período t e a um cenário s . As dimensões desses subproblemas são apresentadas na Tabela 34.

O problema é representado conforme uma árvore multi-estágios com cenários estocásticos cuja dimensão será variada conforme descrito na Tabela 35.

Ao problema-base foram adicionadas duas restrições condicionais, definidas para cada patamar de carga, período e cenário. Essas restrições são baseadas em regras operativas estabelecidas pela Agência Nacional de Águas e Saneamento Básico (ANA), conforme descrito a seguir:

- **Usina Hidrelétrica de Emborcação:** imposição de um limite mínimo de defluência em

Total de usinas hidráulicas	162
Número de usinas hidráulicas com reservatórios	67
Número de usinas térmicas	128
Número de submercados de energia	5
Número de reservatórios equivalentes de energia	12
Número de patamares de carga por submercado	3
Número de restrições de volume	20
Número de restrições de vazão	107
Número de restrições elétricas	51

Tabela 33 – Características do caso de grande porte

Número total de variáveis contínuas	≈ 6200
Restrição de balanço hidráulico	162
Restrições de atendimento a demanda	15
Restrições de função de produção hidráulica	2880
Restrições de vazão	321
Restrições de volume	20
Restrições elétricas	153
Restrições de energia armazenada por REE (presente nos subproblemas do último estágio)	12
Restrições da FCF multivariada por REE (presente nos subproblemas do último estágio)	6000
Total de restrições subproblema sem e com FCF	3551 / 9563

Tabela 34 – Descrição das restrições de cada subproblema da árvore de cenários

Número de períodos (t)	4 / 5 / 6 / 7
Número de aberturas por período	2
Total de nós (subproblemas) na árvore (t, s)	15 / 31 / 63 / 127

Tabela 35 – Dimensões e características das árvores de cenários consideradas

todos os patamares de carga p , condicionado ao volume inicial do respectivo período e cenário, conforme:

$$q_{t,s}^p \geq 80 \text{ m}^3/\text{s}, \quad \text{se } vi_{t,s} > 45\%, \quad \forall p, \forall t, s. \quad (9.1)$$

- **Usina Hidrelétrica de Furnas:** imposição de um limite máximo de defluência em todos os patamares de carga p , condicionado ao volume inicial do respectivo período e cenário, conforme:

$$q_{t,s}^p \leq 300 \text{ m}^3/\text{s}, \quad \text{se } vi_{t,s} < 50\%, \quad \forall p, \forall t, s. \quad (9.2)$$

Além disso, foram consideradas as restrições associadas à operação dos vertedouros das usinas. Essas restrições decorrem de limitações físicas, uma vez que o vertimento não pode ocorrer quando o nível do reservatório se encontra abaixo da soleira do vertedouro. Para

representar essa condição, as restrições foram modeladas em função do volume médio do reservatório no respectivo período e cenário, conforme a equação a seguir:

$$s_{t,s}^p \leq 0, \quad \text{se } \frac{vi_{t,s} + vf_{t,s}}{2} < v_{sol}, \quad \forall p, \forall t, s. \quad (9.3)$$

Onde v_{sol} representa o volume mínimo a partir do qual o vertedouro pode ser acionado. Essas restrições afetam 22 reservatórios do sistema e são consideradas em todos os patamares de carga, períodos e cenários.

Assim, o caso de grande porte, em suas versões com 4, 5, 6 e 7 períodos, foi resolvido considerando-se três variantes:

- **no-constr**: caso sem as restrições condicionais, correspondendo diretamente ao caso-base;
- **cond-constr**: caso-base com as restrições condicionais (9.1), (9.2) e (9.3). Em relação ao caso-base, essa variante adiciona 144 variáveis binárias e 264 restrições lineares por subproblema;
- **with-constr**: caso-base com as restrições (9.1), (9.2) e (9.3) consideradas de forma incondicional, ou seja, independentemente do volume do reservatório. Nessa variante, não são introduzidas variáveis adicionais em relação ao caso-base, sendo acrescentadas apenas 72 restrições lineares por subproblema.

Em todas as execuções, foi imposto um limite máximo de 72 horas de processamento, sendo cada execução interrompida ao atingir esse limite. A arquitetura de implementação e o solver utilizados foram os mesmos empregados nos experimentos descritos nos Capítulos 5 e 8.

Todos os experimentos foram executados em um servidor equipado com processador Intel® Xeon® Silver 4216, operando a 2,10 GHz, com 32 núcleos e 500 GB de memória RAM. Para cada execução, foram disponibilizados dois processos, coordenados por um sistema de filas.

9.2 Solução global do problema de grande porte

Inicialmente, o problema de grande porte foi resolvido de forma monolítica. Os resultados referentes ao custo total de operação e ao tempo computacional são apresentados, respectivamente, nas Tabelas 37 e 36.

Observa-se que essa abordagem de solução torna-se computacionalmente inviável à medida que a árvore de cenários cresce. Mesmo para árvores ainda relativamente pequenas, como a de 127 nós correspondente ao caso com 7 períodos, foram necessárias mais de 5 horas para obtenção de uma solução do problema linear (*LP-Global*). Já para a variante contendo restrições condicionais (*MILP-Global*), o limite máximo de 72 horas de processamento foi atingido sem que a solução ótima fosse comprovada.

	no-constr (LP-Global)	cond-constr (MILP-Global)	with-constr (LP-Global)
4 períodos	4.6 min	13.2 min	4.2 min
5 períodos	30 min	4.0 horas	22 min
6 períodos	2.1 horas	22.3 horas	1.9 horas
7 períodos	9.3 horas	72* horas	5.9 horas

Tabela 36 – Tempo computacional para a resolução monolítica de cada variante do caso de grande porte.

	no-constr (LP-Global)	cond-constr (MILP-Global)	with-constr (LP-Global)
4 períodos	34796.7×10^7	35135.8×10^7	35218.7×10^7
5 períodos	34854.9×10^7	35213.9×10^7	35352.5×10^7
6 períodos	34945.3×10^7	35318.0×10^7	35464.9×10^7
7 períodos	35113.4×10^7	-	35632.2×10^7

Tabela 37 – Custo da solução, em R\$, obtido pela resolução monolítica de cada variante do caso de grande porte.

Com relação ao custo da solução (sc), observa-se, conforme esperado, que valores mais elevados são obtidos à medida que o problema se torna mais restritivo. Em particular,

$$sc_{no} \leq sc_{cond} \leq sc_{with}.$$

Esses resultados são utilizados como referência para a avaliação das soluções obtidas pelos métodos de decomposição apresentados na sequência. Nas análises subsequentes, tais valores serão tratados como soluções *exatas*, embora estejam associados à tolerância de otimalidade adotada pelo solver.

9.3 Decomposição do problema de grande porte linear

Nesta seção, as variantes lineares do problema de grande porte, **no-constr** e **with-constr**, são resolvidas por meio da decomposição via DDP com cortes de Benders. Foi adotado como critério de parada um *gap* relativo inferior a $1 \times 10^{-3}\%$, o mesmo critério utilizado pelo Operador Nacional do Sistema Elétrico (ONS) no planejamento da operação do sistema brasileiro.

Os resultados de tempo computacional, número de iterações necessárias para a convergência da DDP e custo da solução obtida são apresentados nas Tabelas 38 e 39.

Em todos os casos analisados, a DDP apresentou desempenho superior ao método *LP-Global*, especialmente para árvores de cenários de maior dimensão, evidenciando sua melhor escalabilidade à medida que o tamanho do problema aumenta. Além disso, ainda seria possível obter reduções significativas no tempo computacional por meio da paralelização do algoritmo. Em problemas com estruturas de árvore semelhantes, [91] demonstrou reduções de tempo entre 6 e 30 vezes, dependendo da estratégia de paralelização empregada. Dessa forma, a decomposição

	no-constr		with-constr	
	Tempo	Iter.	Tempo	Iter.
4 períodos	2.7 min	52	3.9 min	59
5 períodos	13.9 min	127	14.5 min	136
6 períodos	44.1 min	220	58.5 min	269
7 períodos	1.3 horas	188	2.9 horas	365

Tabela 38 – Tempo computacional e número de iterações necessários para a convergência da DDP nos casos de grande porte.

	no-constr	with-constr
4 períodos	34796.9×10^7	35218.9×10^7
5 períodos	34855.1×10^7	35352.7×10^7
6 períodos	34945.5×10^7	35465.1×10^7
7 períodos	35113.6×10^7	35632.4×10^7

Tabela 39 – Custo da solução, em R\$, obtida por meio da DDP para os casos de grande porte.

mostra-se uma alternativa computacionalmente bastante atrativa e, para árvores de cenários maiores, pode tornar-se indispensável.

Os custos obtidos foram muito próximos daqueles encontrados na resolução monolítica dos problemas correspondentes, permanecendo dentro das tolerâncias adotadas. Esse resultado confirma a capacidade da DDP de convergir para a solução ótima em problemas lineares.

Em relação ao número de iterações, observa-se uma tendência de crescimento conforme aumentam o número de períodos e a quantidade de nós da árvore de cenários. Adicionalmente, o tempo médio por iteração também se eleva, uma vez que o número de subproblemas resolvidos em cada passagem do algoritmo aumenta com o tamanho da árvore.

9.4 Decomposição do problema de grande porte com restrições condicionadas

9.4.1 Solução inexata via DDP

O problema de grande porte foi avaliado utilizando cortes de Benders e cortes de Benders Reforçados. Os resultados obtidos são apresentados nas Tabelas 40 e 41.

Observa-se que, em nenhum dos casos analisados, a DDP foi capaz de convergir para a solução ótima com o *gap* de convergência especificado. Em todos os casos ocorreu estagnação do processo de convergência após determinado número de iterações, a partir do qual não se verificaram melhorias significativas nos limitantes nem na qualidade das soluções obtidas.

Nos casos com 6 e 7 períodos, a melhor solução encontrada para a variante com restrições condicionais apresentou custo superior ao da variante **with-constr**, na qual as mesmas restrições são consideradas de forma incondicional. Esse resultado evidencia a incapacidade

	Benders		Benders Reforçado	
	Tempo	#Iter.	Tempo	#Iter.
4 períodos	4.0 min	22	5.3 min	20
5 períodos	6.7 min	17	11.0 min	19
6 períodos	25.9 min	32	52.5 min	44
7 períodos	61.9 min	39	180.2 min	74

Tabela 40 – Tempo computacional e número de iterações da DDP até a estagnação da solução para os casos de grande porte com restrições condicionais (**cond-constr**), utilizando cortes de Benders e Benders Reforçados.

	Benders		Benders Reforçado	
	Custo (R\$)	Gap(%)	Custo (R\$)	Gap(%)
4 períodos	35174.6×10^7	0.8	35192.2×10^7	0.8
5 períodos	35332.7×10^7	1.1	35316.2×10^7	1.1
6 períodos	35498.7×10^7	1.4	35494.9×10^7	1.4
7 períodos	35706.0×10^7	1.6	35693.4×10^7	1.6

Tabela 41 – Custo da solução e *gap* de otimalidade obtidos pela DDP até a estagnação da solução para os casos de grande porte com restrições condicionais (**cond-constr**), utilizando cortes de Benders e Benders Reforçados.

dos cortes de Benders tradicionais e reforçados de representar adequadamente o impacto das restrições condicionais, comprometendo a qualidade da aproximação da função de custo futuro e, conseqüentemente, o processo de convergência da DDP.

9.4.2 Solução exata via DDiP

A resolução do caso de grande porte por meio da DDiP utilizando cortes de Lagrange mostrou-se inviável mesmo para o menor dos casos analisados. Foi empregada uma expansão binária simplificada, com apenas cinco variáveis binárias por usina, discretizando os níveis dos reservatórios em 32 valores possíveis. Ainda assim, não foi possível observar a convergência do algoritmo dentro do limite máximo de 72 horas de processamento.

Nessa execução, foram utilizados cortes de Lagrange tradicionais gerados por meio do método do subgradiente. Por outro lado, ao empregar *Tighter Lagrangian Cuts* com o método dos planos cortantes, observou-se elevada instabilidade numérica e algorítmica, impossibilitando a obtenção de resultados consistentes.

	Tempo	Iter.	Custo (R\$)	Gap(%)
4 períodos	72* horas	29	36428.0×10^7	27.5

Tabela 42 – Tempo computacional, número de iterações, custo da solução e *gap* de otimalidade obtidos pela DDiP para o caso de grande porte com 4 períodos.

De fato, na literatura, aplicações de métodos de decomposição a problemas de grande

porte envolvendo variáveis inteiras geralmente utilizam cortes inexatos, como cortes de Benders ou Benders Reforçados, cuja convergência para a solução ótima não é garantida [65, 43, 69, 71, 66, 67]. Os autores argumentam que, embora esses métodos possam não atingir a solução ótima, sua utilização se torna necessária devido ao elevado custo computacional associado ao uso de cortes de Lagrange. Essa constatação motivou a análise do impacto da expansão binária em problemas de grande porte, apresentada na seção seguinte.

9.5 Impacto da expansão binária no problema de grande porte

Avalia-se, nesta seção, o impacto do aumento da granularidade da expansão binária na resolução do problema de grande porte por meio da análise do tempo de CPU necessário para resolver o subproblema associado a um único nó da árvore de cenários. Para isolar o efeito da expansão binária, as restrições condicionais foram desconsideradas nesta análise, de modo que as únicas variáveis binárias presentes são aquelas introduzidas pela expansão binária utilizada para representar os volumes armazenados nos 67 reservatórios do sistema.

Para cada configuração do número de variáveis binárias por usina, o subproblema foi resolvido utilizando 21 instâncias geradas aleatoriamente considerando diferentes condições de afluência para todas as usinas. Esse procedimento permitiu estimar o tempo computacional médio de resolução. Além disso, foi imposto um limite máximo de 10 minutos (600 segundos) para cada execução.

Os resultados apresentados na Tabela 43 mostram, para diferentes granularidades da expansão binária, o tempo médio de CPU obtido nas 21 instâncias avaliadas, bem como o número de execuções que atingiram o limite máximo de tempo. É importante destacar que, à medida que o número de variáveis binárias aumenta, o tempo médio observado tende a subestimar o esforço computacional real, uma vez que parte das execuções é interrompida ao atingir o limite de tempo sem que o subproblema seja efetivamente resolvido.

Variáveis binárias por usina	Número total de variáveis binárias	Tempo médio de CPU (s)	Número de timeouts
1	67	9	0
3	201	52	1
5	335	53	1
7	469	228	7
9	603	527	16
11	737	593	19
13	871	600	21

Tabela 43 – Tempo médio de CPU para a resolução de um subproblema de grande porte com diferentes níveis de granularidade da expansão binária.

Observa-se um crescimento acentuado do tempo computacional à medida que a granularidade da expansão binária aumenta, especialmente nas transições de 5 para 7 e de 7 para 9

variáveis binárias por usina. Mesmo para configurações relativamente modestas, como aquelas com 3 ou 5 variáveis binárias por usina, o tempo necessário para resolver um único subproblema já se aproxima de um minuto.

Considerando que, no procedimento DDiP, não apenas esse subproblema, mas todos os subproblemas associados aos diferentes nós da árvore de cenários precisam ser resolvidos repetidamente ao longo das iterações do algoritmo, o esforço computacional total torna-se proibitivo. Esse comportamento corrobora os resultados observados na aplicação da DDiP com a configuração *Lagr_5bin*, apresentados na Tabela 42, que demandou elevado esforço computacional sem produzir resultados competitivos.

Dessa forma, com base nos resultados apresentados nesta seção e na Seção 9.4.2, conclui-se que a aplicação da versão exata da DDiP não é viável para o problema de grande porte considerado neste trabalho. Em contrapartida, os resultados apresentados na Seção 9.4.1 mostram que os métodos inexatos baseados em cortes de Benders apresentam desempenho computacional adequado, embora à custa da perda de garantias de otimalidade. Essa dicotomia entre qualidade da solução e esforço computacional motiva a metodologia híbrida proposta no capítulo seguinte, que busca combinar as principais vantagens de ambas as abordagens.

10 Proposta de decomposição em duas etapas para MS-MILPs

As dificuldades observadas no Capítulo 9 evidenciam que a resolução de problemas de grande porte frequentemente fica restrita a métodos inexatos, nos quais a solução ótima não é alcançada, ou a abordagens que demandam custos computacionais extremamente elevados, muitas vezes proibitivos na prática. Nesse contexto, propõe-se um procedimento de solução em duas etapas, concebido para combinar o elevado desempenho computacional das abordagens de Benders e Benders Reforçado com a exatidão do DDiP baseado em cortes Lagrangianos.

Embora a metodologia proposta ainda possua caráter heurístico, mostra-se capaz de produzir soluções de melhor qualidade do que aquelas obtidas pelas duas primeiras abordagens, ao mesmo tempo em que evita o elevado custo computacional associado ao DDiP quando este é empregado em conjunto com a expansão binária.

10.1 Descrição da metodologia

A metodologia proposta consiste em substituir a expansão binária tradicional, que utiliza N variáveis binárias para representar os volumes das usinas hidrelétricas, por uma formulação simplificada baseada em uma única variável binária por usina. Nessa formulação, cada variável binária permite representar dois níveis distintos de volume, denotados por V_1 e V_2 . Dessa forma, a expansão binária descrita em (4.17) é substituída por:

$$vf = V_1 + (V_2 - V_1) \cdot b \quad (10.1)$$

Com essa simplificação, o processo de decomposição passa a ser composto por duas etapas principais:

1. Determinação dos vetores de volumes $V_{1t,s} \in \mathbb{R}^H$ e $V_{2t,s} \in \mathbb{R}^H$, onde H representa o número de usinas hidrelétricas consideradas no sistema, para cada período t e cenário s , a partir de soluções aproximadas do problema original.
2. Resolução do problema por meio do algoritmo DDiP, utilizando a representação simplificada dos volumes dos reservatórios, baseada em uma única variável binária por usina, com objetivo de melhorar as soluções aproximadas obtidas.

10.1.1 Etapa 1: Determinação de soluções aproximadas

Aplica-se inicialmente a abordagem DDP para resolver, de forma independente (e possivelmente em paralelo), as três seguintes aproximações contínuas do problema:

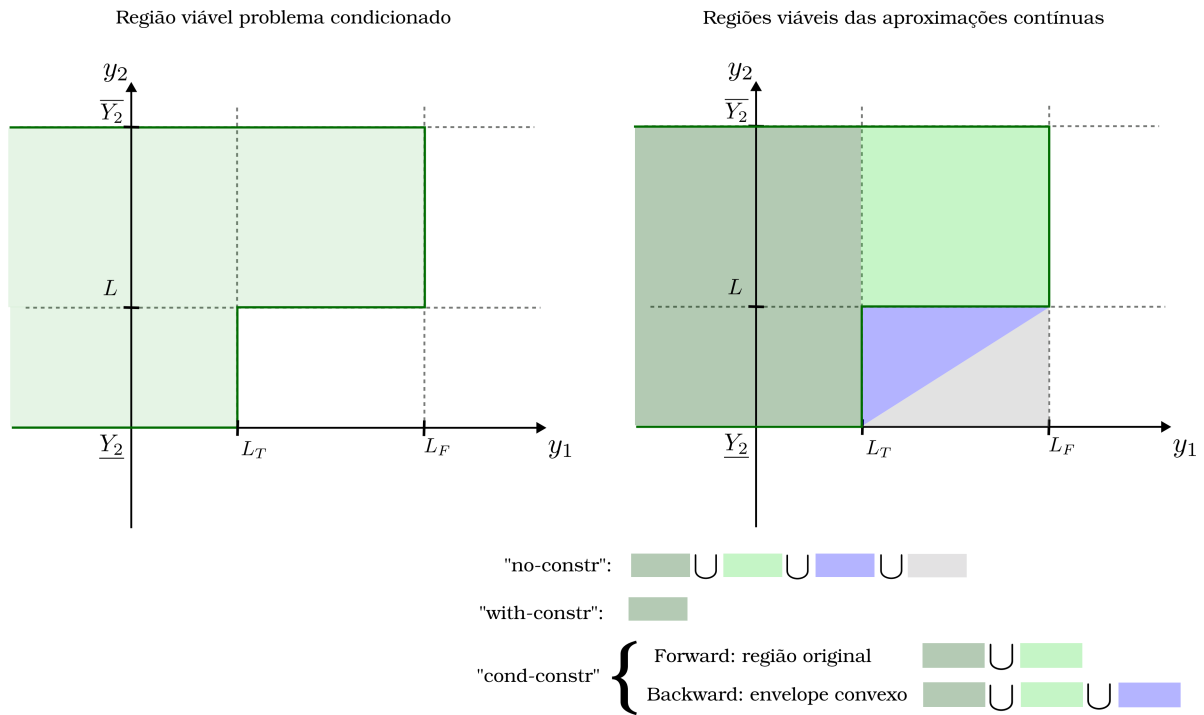


Figura 19 – Regiões viáveis induzidas para os diferentes tipos de problemas considerados na Etapa 1 da abordagem proposta.

- problema irrestrito (“**no-constr**”): onde as restrições condicionais são convertidas em restrições lineares considerando apenas o limite mais relaxado;
- problema incondicional (“**with-constr**”): onde as restrições condicionais são convertidas em restrições lineares considerando apenas o limite menos relaxado;
- problema condicional (“**cond-constr**”): onde as restrições condicionais são consideradas no passo *forward*, mas, no passo *backward* da DDP, as restrições de integralidade associadas às variáveis binárias são relaxadas.

Os dois primeiros problemas são programas lineares e podem ser resolvidos de forma exata (dentro de uma tolerância aceitável) via DDP tradicional, enquanto o último, ao ser resolvido com a DDP, não possui convergência garantida.

A Figura 19 ilustra a região viável da restrição condicionada descrita na Seção 2.5 e as respectivas regiões viáveis para os subproblemas associados às três aproximações contínuas consideradas na Etapa 1. Observa-se que a região viável dos subproblemas resolvidos nos passos *backward* da abordagem “cond-constr”, na qual as restrições de integralidade das variáveis binárias b_T e b_F são relaxadas (de forma a permitir a aplicação de cortes de Benders e Benders Reforçado), corresponde à envoltória convexa da região viável do problema original. Assim, esta aproximação corresponde a uma versão convexificada do problema MS-MILP original.

Como a principal motivação desta primeira etapa é obter trajetórias candidatas de volumes dos reservatórios, consideram-se soluções associadas a essas diferentes funções de recurso, que limitam inferior e superiormente a função de recurso (não convexa) do problema MS-MILP original. Denota-se por $V_{no\ i}^{t,s}$, $V_{with\ i}^{t,s}$ e $V_{cond\ i}^{t,s}$ o armazenamento final no reservatório i no período t e cenário s na solução de cada um desses problemas, respectivamente.

Dado que as soluções obtidas por Benders ou Benders Reforçados são frequentemente consideradas adequadas pela literatura, e que muitas usinas podem não estar sujeitas a restrições condicionais, as soluções do problema “cond-constr” são consideradas preferíveis. Neste sentido, adota-se o seguinte critério para determinar os dois valores de armazenamento de cada usina i representados pelos elementos V_{1t,s_i} e V_{2t,s_i} :

$$\text{se } V_{no\ i}^{t,s} \neq V_{with\ i}^{t,s} \neq V_{cond\ i}^{t,s} \neq V_{no\ i}^{t,s} : \quad V_{1t,s_i} = \min\{V_{no\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (10.2)$$

$$V_{2t,s_i} = \max\{V_{no\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (10.3)$$

$$\text{se } V_{no\ i}^{t,s} = V_{with\ i}^{t,s} = V_{cond\ i}^{t,s} : \quad V_{1t,s_i} = V_{2t,s_i} = V_{cond\ i}^{t,s} \quad (10.4)$$

Caso contrário, dois destes três valores serão iguais, e utiliza-se naturalmente:

$$V_{1t,s_i} = \min\{V_{no\ i}^{t,s}, V_{with\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (10.5)$$

$$V_{2t,s_i} = \max\{V_{no\ i}^{t,s}, V_{with\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (10.6)$$

Por fim, considere usinas hidrelétricas sujeitas a restrições condicionais dependentes do volume inicial do reservatório. Nessas situações, o limite aplicado à variável de decisão $y_{t,s}$ depende da comparação entre o volume inicial vi_{t,s_i} e um valor crítico $v_{crit\ i}$, que define um ponto de transição da restrição. Tais restrições podem ser de limite superior:

$$y_{t,s} \leq \begin{cases} L & \text{se } vi_{t,s_i} < v_{crit\ i} \\ L + M & \text{caso contrário} \end{cases} \quad (10.7)$$

ou inferior:

$$y_{t,s} \geq \begin{cases} L & \text{se } vi_{t,s_i} > v_{crit\ i} \\ L - M & \text{caso contrário} \end{cases} \quad (10.8)$$

No caso da restrição de limite superior, o valor L é aplicado quando o nível do reservatório está abaixo do volume crítico, caracterizando a condição mais restritiva. Caso contrário, a restrição é relaxada para $L + M$. De forma análoga, na restrição de limite inferior, L representa o limite mais restritivo quando o volume está acima do nível crítico, sendo relaxado para $L - M$ quando essa condição não é satisfeita.

Dessa forma, é conveniente que o valor v_{criti} seja incluído no conjunto de possíveis estados para a usina i . A introdução explícita desse ponto de transição é importante para garantir a possibilidade de operação da usina exatamente sobre o limite onde a restrição muda de regime. Assim, são consideradas as regras adicionais:

- para usinas com restrições (10.7):

$$\text{se } v_{criti} < V_{1t,s_i} \rightarrow V_{1t,s_i} = v_{criti}; V_{2t,s_i} = V_{condi}^{t,s} \quad (10.9)$$

- para usinas com restrições (10.8):

$$\text{se } V_{2t,s_i} < v_{criti} \rightarrow V_{1t,s_i} = V_{condi}^{t,s}; V_{2t,s_i} = v_{criti} \quad (10.10)$$

A escolha dos níveis de volumes que definem as trajetórias candidatas dos reservatórios é crucial para o desempenho do método de decomposição proposto em duas etapas. O Apêndice E apresenta as vantagens das regras propostas em (10.2)-(10.10) para a obtenção das soluções candidatas, em comparação com outras regras alternativas.

10.1.2 Etapa 2: Melhoria da solução aproximada

As soluções obtidas na etapa anterior podem ser inviáveis, como no caso da variante “no-constr”, ou viáveis, porém não necessariamente ótimas, como nas variantes “cond-constr” e “with-constr”. Assim, nesta segunda etapa, são introduzidas variáveis binárias adicionais $b_{t,s} \in \{0,1\}^H$ e variáveis de recurso $b_{t-1,s(a)} \in \{0,1\}^H$ para representar, respectivamente, os volumes finais e iniciais das H usinas hidrelétricas em cada período t e cenário s , por meio das seguintes relações de expansão binária:

$$vf_{t,s} = V_{1t,s} + (V_{2t,s} - V_{1t,s})b_{t,s} \quad (10.11)$$

$$vi_{t,s} = V_{1t-1,s(a)} + (V_{2t-1,s(a)} - V_{1t-1,s(a)})b_{t-1,s(a)} \quad (10.12)$$

Observe que, quando $V_{1t,s_i} = V_{2t,s_i}$, a variável binária associada à usina i no subproblema (t,s) torna-se desnecessária, uma vez que o volume final correspondente fica fixado. Com essa nova representação, obtém-se um problema MS-MILP com variáveis de recurso binárias, o qual pode ser resolvido exatamente pela DDiP utilizando cortes de Lagrange.

A solução obtida ao final da segunda etapa apresenta as seguintes propriedades:

- os volumes finais de cada reservatório ficam restritos aos valores candidatos $V_{1t,s}$ e $V_{2t,s}$ definidos na Etapa 1, reduzindo significativamente o espaço de busca do problema;

- como a solução obtida pela variante “cond-constr” pertence ao conjunto factível da segunda etapa, o valor ótimo do problema resultante é necessariamente menor ou igual ao custo obtido por essa variante;
- o problema resultante pode ser resolvido de forma exata e com convergência finita pela DDiP, em virtude da presença de variáveis de recurso binárias;
- a abordagem se beneficia das informações produzidas na primeira etapa, que fornecem valores candidatos de boa qualidade para os volumes dos reservatórios. Essas informações direcionam a busca da segunda etapa, resultando em soluções de menor custo operativo já nas primeiras iterações. Assim, a primeira etapa desempenha o papel de um *warm start* para a segunda etapa.

A Figura 20 ilustra a abordagem de decomposição em duas etapas proposta. Na primeira etapa, são obtidas as soluções aproximadas associadas às variantes “cond-constr”, “no-constr” e “with-constr”, cujos custos são denotados por sc_{cond} , sc_{no} e sc_{with} , respectivamente. Na segunda etapa, resolve-se exatamente o problema simplificado baseado na discretização binária dos volumes, obtendo-se a solução de custo sc_{lagr} . Suponha que o custo ótimo do problema original seja conhecido e denotado por sc_{exact} . Nessa situação, verificam-se as seguintes relações:

- Seja sc_{cond} o custo da solução do problema “cond-constr”. Como a convergência não é garantida neste caso, tem-se

$$sc_{\text{exact}} \leq sc_{\text{cond}}.$$

- Seja sc_{no} o custo da solução do problema “no-constr”. Como esta formulação corresponde a uma relaxação do problema original, segue-se que

$$sc_{\text{no}} \leq sc_{\text{exact}}.$$

- Seja sc_{with} o custo da solução do problema “with-constr”. Como este problema é mais restritivo do que o problema original, obtém-se

$$sc_{\text{exact}} \leq sc_{\text{with}}.$$

- Finalmente, seja sc_{lagr} o custo da solução obtida pela decomposição em duas etapas. Como esta abordagem se baseia numa aproximação binária com uma única variável, e uma das trajetórias de volume consideradas na segunda etapa reproduz a solução obtida pela variante “cond-constr”, tem-se

$$sc_{\text{no}} \leq sc_{\text{exact}} \leq sc_{\text{lagr}} \leq sc_{\text{cond}}.$$

Em síntese, a abordagem proposta fornece um limitante superior potencialmente mais próximo do custo ótimo do problema original do que aquele obtido diretamente pela variante

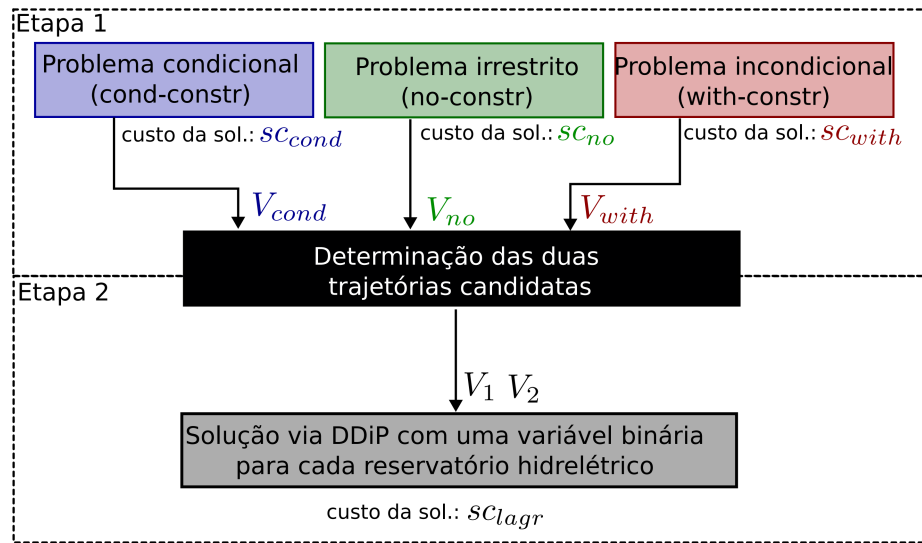


Figura 20 – Representação da abordagem de decomposição em duas etapas

“cond-constr”, abordagem usualmente empregada na literatura. De fato, ao explorar simultaneamente diferentes trajetórias candidatas para os volumes dos reservatórios, a segunda etapa é capaz de identificar soluções de menor custo dentro do espaço discreto considerado. Além disso, como os volumes provenientes da variante “with-constr” também podem ser incorporados ao processo de busca, a metodologia ainda tem potencial de obtenção de solução mais acurada do que esta variante.

10.2 Resultados computacionais para o problema reduzido

O problema “reduced-system” contendo as restrições condicionais (vide Seção 5.2) foi resolvido por meio da decomposição em duas etapas. Na primeira etapa, utilizou-se a solução obtida pela DDP com cortes de Benders para as três variantes consideradas (“no-constr”, “with-constr” e “cond-constr”). Na segunda etapa, aplicou-se a DDiP com cortes de Lagrange tradicionais e método do subgradiente.

Para avaliar a qualidade das soluções obtidas, utiliza-se o “gap de otimalidade” em R\$, definido como a diferença entre o custo da solução encontrada e o custo da solução ótima obtida pelo MILP global:

$$\text{gap de otimalidade} = UB - sc_{exact} \quad (10.13)$$

Define-se também o “gap de otimalidade relativo”, correspondente ao percentual do gap de otimalidade em relação ao custo presente de operação (obtido a priori pela solução exata). O custo futuro não é considerado nesse cálculo, pois sua magnitude é significativamente superior à do custo presente e tende a mascarar as diferenças entre as soluções avaliadas. Assim:

$$\text{gap de otimalidade relativo} = 100 \times \frac{UB - sc_{exact}}{\text{custo presente}_{exact}} \quad (10.14)$$

Os resultados de tempo computacional, gap de otimalidade e número de iterações são apresentados na Tabela 44. Para fins de comparação, também são incluídos os resultados apresentados na Seção 5.2 para o mesmo problema resolvido por meio dos seguintes métodos:

- DDP com cortes de Benders
- DDP com cortes de Benders Reforçados
- DDiP com cortes de Lagrange com expansão binária considerando 5, 6 e 7 variáveis binárias

Método	Número de Iterações	Tempo Total (sec)	Gap de Otim. (\$)	Gap de Otim. (%)
2Step	2	4 + 6	360,736	0.10
DDiP Lagr. 7bin	180	2993	1,925,155	0.56
Benders	4	4	2,485,826	0.72
SBenders	3	8	2,485,826	0.72
DDiP Lagr. 6bin	45	556	9,558,329	2.78
DDiP Lagr. 5bin	12	97	102,379,205	29.79

Tabela 44 – Performance dos métodos de decomposição aplicados ao problema “reduced-system”

A Figura 21 apresenta o processo de convergência (limites inferior e superior) ao longo das iterações. Os resultados da DDiP com cortes de Lagrange não foram incluídos, pois apresentam escalas de custo e número de iterações significativamente distintas.

Diante dos resultados pode-se observar que:

- todas as abordagens analisadas apresentaram tempos de processamento reduzidos quando comparadas à DDiP com expansão binária tradicional;
- com cortes de Benders ou Benders Reforçados, a DDP apresenta estagnação e não converge, permanecendo com diferença entre os limites superior e inferior ($UB > LB$);
- a decomposição em duas etapas alcança convergência ao utilizar a DDiP na segunda fase. Entretanto, o método ainda apresenta um gap de otimalidade residual devido à aproximação introduzida pela representação binária simplificada;
- as execuções com cortes de Lagrange e DDiP requerem um número significativamente maior de iterações. A perda de precisão associada ao uso de poucas variáveis binárias somente é superada quando são utilizadas pelo menos sete variáveis binárias, tornando o método competitivo em relação à DDP;

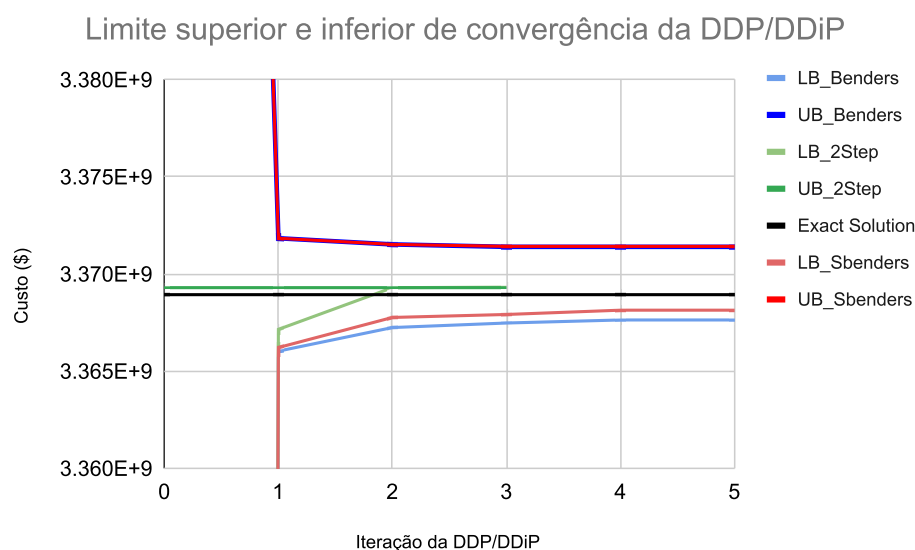


Figura 21 – Convergência da DDP/DDiP para o problema “reduced-system” considerando os métodos não exatos.

Além disso, a abordagem proposta reduz substancialmente o gap de otimalidade em comparação com os métodos baseados exclusivamente em cortes de Benders, ao custo de um aumento moderado no tempo computacional. Dessa forma, a decomposição em duas etapas constitui uma alternativa atrativa, combinando a rapidez dos métodos inexatos com o refinamento proporcionado pela DDiP.

10.3 Resultados computacionais para o problema de grande porte

Nesta seção, a metodologia proposta é avaliada no problema de grande porte descrito na Seção 9.1, com o objetivo de analisar sua aplicabilidade e escalabilidade em um problema real. A Figura 22 apresenta a evolução dos limites inferior (*LB*) e superior (*UB*) ao longo das iterações da DDP/DDiP para cada abordagem considerada, bem como o valor ótimo obtido pela resolução do MILP global (linha preta).

Observa-se, inicialmente, que a DDiP com cinco variáveis na expansão binária e cortes de Lagrange tradicionais apresenta uma convergência substancialmente mais lenta que as demais abordagens. Em razão do elevado esforço computacional, sua execução foi interrompida após 29 iterações, ao atingir o limite de 72 horas de processamento.

Conforme esperado, os cortes de Benders Reforçados produzem limites inferiores mais elevados do que aqueles obtidos com os cortes de Benders tradicionais, indicando uma aproximação mais precisa da função de custo futuro. Esse ganho, entretanto, é acompanhado por um aumento de aproximadamente 40% no tempo de CPU. Apesar dessa diferença, os custos das soluções obtidas pelos dois métodos permanecem bastante próximos, sugerindo que as decisões

operativas subótimas identificadas por ambas as abordagens são semelhantes.

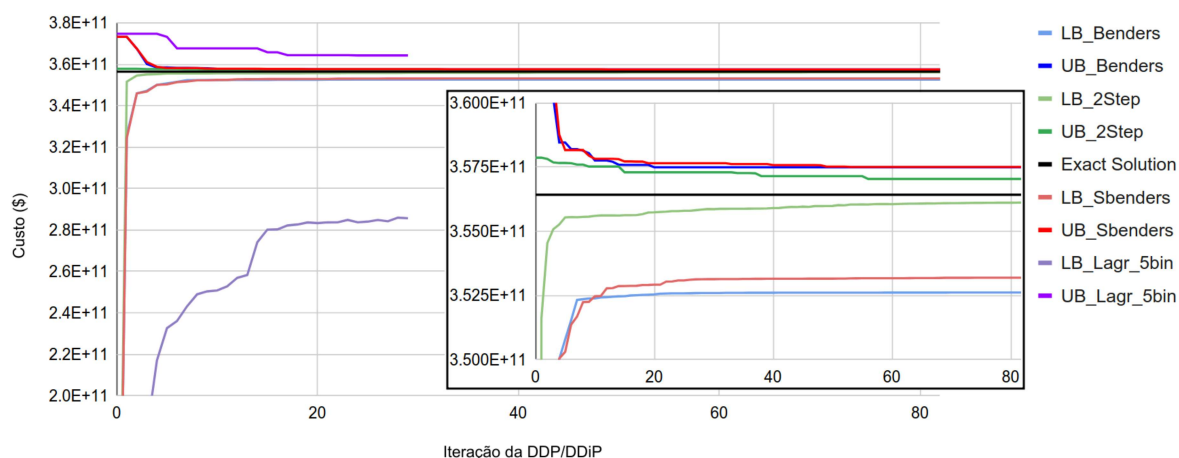


Figura 22 – Convergência da DDP/DDiP para o problema de grande porte.

Em contraste, a etapa da DDiP da decomposição em duas etapas apresenta uma convergência melhor, produzindo soluções (*UB*) menos custosas já nas primeiras iterações e um aumento muito mais rápido do limite inferior. Ainda assim, apesar da convergência comprovada, o método continua a apresentar dificuldades para reduzir o gap de convergência além de determinado ponto, o que exigiria um número significativamente maior de iterações e tempo computacional.

A Tabela 45 apresenta o número de iterações, o tempo de processamento e o gap de otimalidade obtidos por cada abordagem. Os resultados da decomposição em duas etapas são reportados para diferentes quantidades de iterações executadas na segunda fase. Observa-se que mesmo um número reduzido de iterações já é suficiente para produzir soluções de melhor qualidade quando comparadas àquelas obtidas exclusivamente pela DDP.

À medida que o número de iterações da segunda etapa aumenta, verificam-se reduções adicionais no gap de otimalidade. Naturalmente, esses ganhos são acompanhados por um aumento no esforço computacional. Ainda assim, os tempos de processamento permanecem moderados quando comparados ao elevado custo associado à aplicação direta da DDiP com expansão binária completa.

Por outro lado, as abordagens baseadas exclusivamente em cortes de Benders e Benders Reforçados apresentam tempos de execução significativamente menores, porém sem capacidade de reduzir o gap de otimalidade após a estabilização dos limites, caracterizando a estagnação do processo iterativo. Em contraste, a estratégia de decomposição em duas etapas converte o acréscimo de esforço computacional em melhorias progressivas na qualidade da solução, reduzindo gradualmente o gap de otimalidade à medida que novas iterações da DDiP são realizadas. Assim, com o método proposto é possível estabelecer um compromisso entre a qualidade da solução e o tempo computacional desejável/disponível.

Método	Iterações DDP/DDiP	Tempo total (min)	Gap de otimalidade (\$)	Gap relativo de otimalidade (%)
2Step	15	8 + 32	873,055,567	7.64
2Step	38	8 + 131	725,017,739	6.35
2Step	56	8 + 284	616,266,468	5.40
Benders	48	8	1,069,557,857	9.43
Sbenders	53	14	1,081,152,007	9.46
Lagr_5bin	29	4320*	7,857,128,667	68.80

Tabela 45 – Resultados para o problema de grande porte

Os tempos computacionais apresentados neste trabalho foram obtidos sem o uso de processamento paralelo. Portanto, o tempo de execução dos métodos de decomposição pode ser reduzido substancialmente na prática.

10.3.1 Resultados de operação

Além da redução dos custos de operação, a análise da operação de usinas específicas mostra que a abordagem de decomposição em duas etapas tem potencial de evitar distorções operativas significativas presentes nas soluções primais obtidas pelos métodos de Benders e Benders Reforçados. Esses métodos apresentam dificuldades para adaptar a operação às restrições condicionais, principalmente porque os cortes são construídos a partir da relaxação linear do problema, o que limita a propagação, para os estágios anteriores, dos efeitos associados à ativação das restrições condicionais

Essas diferenças são ilustradas na Figura 23 para a Usina Hidrelétrica de Furnas, cujo reservatório desempenha papel fundamental na regularização das aflúências do Sistema Interligado Nacional. À esquerda, apresenta-se a operação obtida pela resolução exata do problema como um MILP-global. As linhas azuis representam os volumes iniciais do reservatório em cada estágio, sendo que cada tonalidade corresponde a um dos oito cenários considerados. De forma análoga, as linhas vermelhas representam as vazões turbinadas. A Figura mostra que a estratégia ótima consiste em reduzir a vazão turbinada no estágio 3 para permitir que o volume do reservatório ultrapasse 50% no início do estágio seguinte, possibilitando assim a utilização de vazões superiores a $300\text{m}^3/\text{s}$ nesse estágio.

O gráfico central da Figura 23 apresenta a operação obtida por meio da decomposição de Benders. Observa-se que o método não consegue alcançar uma solução semelhante à ótima: as vazões turbinadas permanecem limitadas a $300\text{m}^3/\text{s}$ em todos os cenários e períodos, e o volume do reservatório nunca ultrapassa o limite de 50%. Isso ocorre porque os cortes de Benders são derivados do problema com relaxação das restrições de integralidade e, portanto, não propagam adequadamente, ao longo da árvore de cenários, informações relacionadas à ativação ou desativação da restrição condicional.

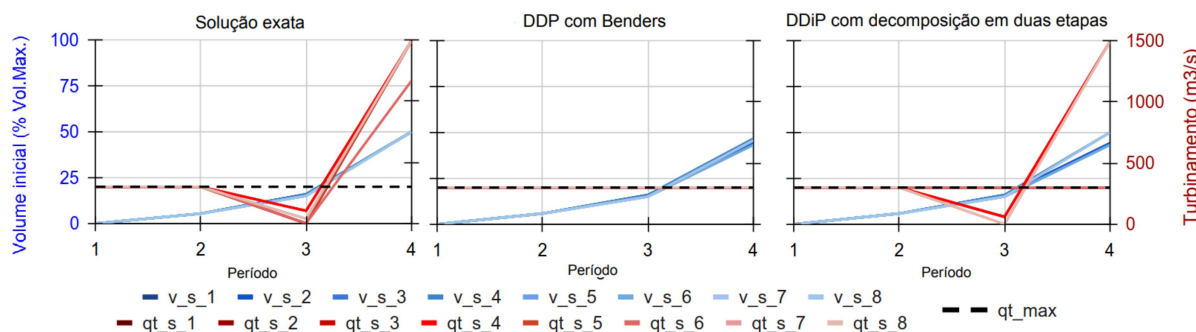


Figura 23 – Operação obtida para a Usina Hidrelétrica de Furnas

Em contraste, a metodologia proposta, ilustrada à direita da Figura 23, produz uma operação semelhante à solução ótima em metade dos cenários (3, 4, 7 e 8), enquanto nos demais cenários apresenta comportamento próximo ao obtido pela decomposição de Benders.

É importante destacar que esses resultados indicam que a metodologia proposta não alcança a solução ótima do problema, uma vez que possui natureza heurística. Ainda assim, ela demonstra potencial para melhorar as decisões operativas obtidas por meio da DDP com decomposição de Benders. No pior caso, o método reproduz a mesma operação obtida por Benders, pois uma das trajetórias de volume selecionadas é derivada diretamente dessa solução.

10.3.2 Validação de desempenho em múltiplas instâncias

Com o objetivo de avaliar a robustez da metodologia proposta, foram geradas 21 instâncias do problema de grande porte por meio da variação aleatória das condições de afluência dos reservatórios. Mais especificamente, as vazões naturais afluentes das 162 usinas hidrelétricas foram perturbadas aleatoriamente em todos os cenários e estágios, de modo a representar diferentes condições hidrológicas e, conseqüentemente, distintas estratégias operativas para o sistema.

Todas as instâncias foram inicialmente resolvidas até a otimalidade utilizando a formulação MILP-global e, em seguida, ordenadas de acordo com o custo total de operação obtido. Posteriormente, cada instância foi resolvida por meio de abordagens de decomposição baseadas em DDP e DDiP, incorporando os três tipos de cortes avaliados neste trabalho: Benders, Benders Reforçado e a decomposição em duas etapas.

Para os cortes de Benders e Benders Reforçados, como a convergência não pode ser alcançada, o procedimento DDP foi interrompido após 60 iterações em todas as instâncias, momento em que o algoritmo já apresentava comportamento de estagnação. De forma semelhante, embora a segunda etapa (DDiP) possua propriedades teóricas de convergência, não foi possível observar tal comportamento dentro de um tempo computacional considerado razoável para o caso de grande porte analisado. Por essa razão, são comparados os resultados obtidos pela

metodologia proposta após 20 e 60 iterações. Os valores do gap relativo de otimalidade de cada instância são apresentados na Figura 24.

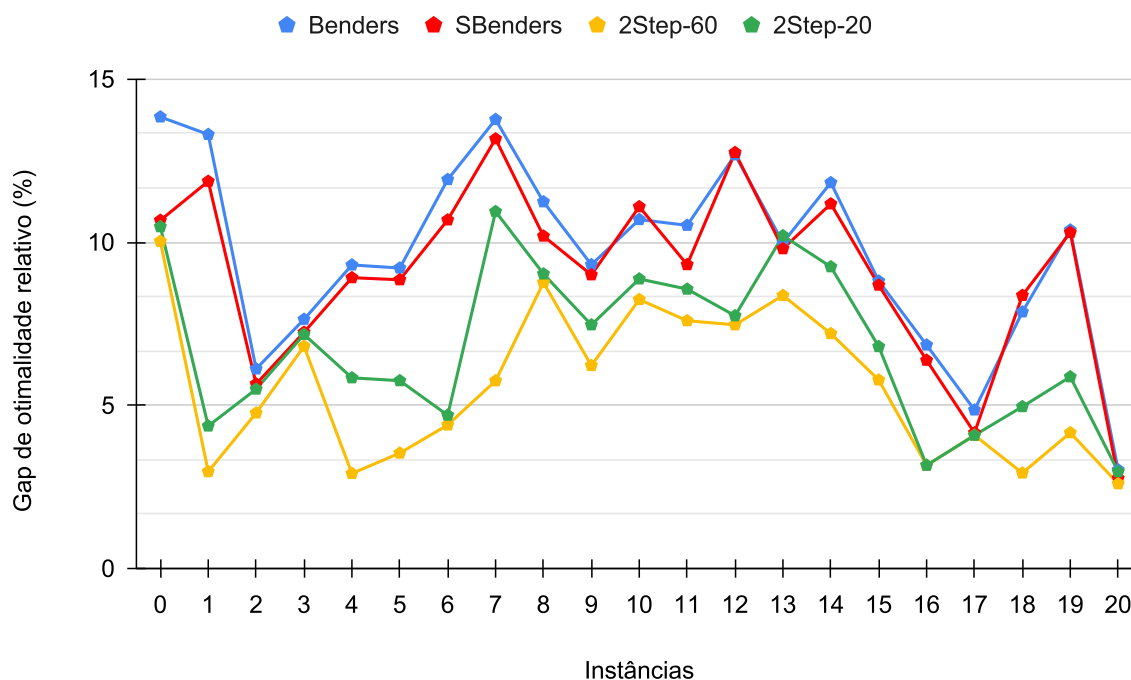


Figura 24 – Gap relativo de otimalidade para cada instância do problema de grande porte.

Observa-se que o DDP com cortes de Benders e Benders Reforçados produz gaps de otimalidade muito semelhantes, sendo que o segundo apresenta um desempenho médio ligeiramente superior. A decomposição em duas etapas proporciona uma redução adicional do gap de otimalidade, e esse ganho se torna mais expressivo à medida que mais iterações do DDiP são executadas.

Além disso, foram registrados os tempos computacionais de cada método. Os valores médios obtidos para as 21 instâncias são apresentados na Tabela 46. Para a metodologia em duas etapas, o tempo reportado corresponde à soma do esforço computacional necessário na primeira etapa (solução baseada em cortes de Benders) e na segunda etapa (DDiP com cortes de Lagrange).

Método	Iterações	Tempo médio (min)
2-step	20	10 + 53
2-step	60	10 + 321
Benders	60	10
S-Benders	60	16

Tabela 46 – Tempo computacional médio para as 21 instâncias do caso de grande porte.

Com base nos resultados computacionais obtidos, observa-se que a abordagem proposta constitui uma alternativa viável para a resolução do problema de grande porte analisado, para o

qual a versão exata do método DDiP não apresentou desempenho computacional compatível com a aplicação prática. Além disso, a metodologia produziu soluções de menor custo quando comparada às abordagens aproximadas baseadas em cortes de Benders e Benders Reforçados, ao custo de um aumento moderado do tempo computacional.

11 Conclusões

Esta tese abordou o problema de planejamento da operação hidrotérmica de médio prazo na presença de restrições condicionais, uma classe de restrições amplamente presente em sistemas energéticos reais e que introduz a necessidade da representação de variáveis inteiras nos modelos de otimização. Essas características dificultam significativamente a aplicação dos métodos clássicos de decomposição, uma vez que comprometem a construção de aproximações precisas para a função de custo futuro, elemento fundamental nos algoritmos de programação dinâmica estocástica.

Inicialmente, foi realizada uma revisão dos principais métodos encontrados na literatura para a resolução de problemas estocásticos multiestágio lineares inteiros mistos, com destaque para as abordagens de decomposição fundamentadas em aproximações da função de recurso. Nesse contexto, foram analisados e comparados os diferentes tipos de cortes propostos na literatura para a representação dessa função.

Posteriormente, foi conduzida uma revisão bibliográfica abrangente sobre o método *Stochastic Dual Dynamic Integer Programming* (SDDiP). Foram examinados os trabalhos que serviram de base para a proposição da técnica, bem como as contribuições posteriores que apresentam extensões, aprimoramentos e variantes do método. Adicionalmente, foram investigadas as diversas aplicações do SDDiP reportadas. Todos os trabalhos identificados na literatura, tanto aqueles relacionados ao desenvolvimento metodológico quanto às aplicações da técnica, são apresentados e discutidos ao longo desta revisão bibliográfica.

O desenvolvimento da tese teve por base a implementação, em um mesmo arcabouço computacional, dos principais métodos encontrados na literatura, incluindo os cortes de Benders, Benders Reforçados, Inteiros de Otimalidade e Lagrangeanos Tradicionais, *Square-minimization cuts*. Essa abordagem permitiu a realização de uma avaliação uniforme e comparativa dessas técnicas sob diferentes perspectivas. Para tal comparação e em todos os experimentos desta tese foi utilizada a técnica da DDiP de forma a isolar o efeito amostral na convergência do método, observando-se a capacidade de convergência de cada tipo de corte.

A avaliação inicial consistiu em uma análise detalhada de um caso tutorial, no qual foi possível investigar de forma isolada o impacto de uma restrição condicional sobre as características do problema e sobre o comportamento dos métodos de solução. Posteriormente, foi considerado um problema representativo de um sistema elétrico de potência real, porém com dimensões reduzidas, com o objetivo de avaliar a eficiência computacional e a qualidade das soluções produzidas por cada método.

Os resultados obtidos indicaram que os cortes inexatos demandam um esforço computacional significativamente menor que os métodos exatos. Entretanto, por não representarem de forma precisa a função de recurso, podem fornecer sinais incorretos ao processo de otimização, conduzindo a soluções substancialmente distantes da ótima. Em contrapartida, os métodos exatos

apresentam maior fidelidade na representação do problema, porém à custa de uma robustez computacional limitada, tornando-se rapidamente proibitivos à medida que a complexidade do problema aumenta.

Adicionalmente, foi realizada uma investigação empírica acerca dos impactos de diferentes tipos de restrições condicionais sobre o comportamento das funções de recurso. Os resultados evidenciaram que a introdução desse tipo de restrição pode provocar o surgimento de descontinuidades e regiões de não convexidade, resultando em funções convexas apenas por partes. Nessas condições, a função de recurso deixa de admitir uma representação exata por meio de cortes lineares.

Uma das principais contribuições desta tese consiste na caracterização e derivação de propriedades estruturais do espaço dual associado aos subproblemas resolvidos no contexto do método SDDiP. A análise teórica desenvolvida permitiu estabelecer relações entre a geometria do conjunto dual ótimo, as diferentes formas de representação das variáveis de cópia e a qualidade dos cortes gerados para os algoritmos de decomposição. Esses resultados fornecem uma compreensão mais profunda dos aspectos que devem ser explorados nos mecanismos de buscas da informação dual, esclarecendo questões que permaneciam pouco explorados na literatura.

Com base nesses resultados teóricos, a principal contribuição original da tese foi a proposta uma nova família de cortes de Lagrange: *Tighter Lagrangian cuts*, construídos a partir de pontos extremos do conjunto dual ótimo. Diferentemente das abordagens existentes, os cortes desenvolvidos exploram de forma mais eficiente a informação dual disponível, produzindo aproximações mais aderentes à estrutura da função de recurso na presença de decisões discretas e restrições condicionais. Como consequência, os cortes propostos fornecem sinais mais informativos ao algoritmo de solução, contribuindo para a redução do número de iterações necessárias à convergência e para a obtenção de políticas de melhor qualidade nos métodos SDDiP/DDiP.

Dessa forma, as contribuições apresentadas nesta tese atuam em duas frentes complementares. A primeira é de natureza teórica, ao ampliar significativamente o entendimento sobre a estrutura dual dos problemas em questão. A segunda é de natureza metodológica, ao converter esse conhecimento em uma nova estratégia para construção de cortes de Lagrange mais eficazes. Em conjunto, esses resultados representam um avanço em relação ao estado da arte, tanto pelo esclarecimento de propriedades fundamentais do problema dual quanto pela proposição de mecanismos capazes de melhorar o desempenho computacional e a qualidade das aproximações utilizadas pelos algoritmos de decomposição. A relevância dessas contribuições é corroborada pelos experimentos computacionais, envolvendo diferentes análises de sensibilidade e estudos comparativos, onde foi observada a superioridade dos cortes propostos em termos de qualidade de aproximação e eficiência na convergência.

A partir das limitações observadas nos métodos de decomposição quando aplicados a problemas de grande porte com restrições condicionais, em especial do custo computacional

da obtenção das soluções em face do critério de convergência, foi desenvolvida uma segunda contribuição original desta tese: uma metodologia híbrida de decomposição estruturada em duas etapas. Na primeira etapa, são empregadas aproximações convexas construídas por meio de cortes de Benders, explorando sua elevada robustez computacional para guiar eficientemente o processo de busca. Na segunda etapa, essas aproximações são refinadas por meio da resolução exata dos subproblemas utilizando o método DDiP com cortes de Lagrange, permitindo capturar com maior fidelidade os efeitos das decisões discretas e das restrições condicionais sobre a função de custo futuro.

A principal inovação dessa abordagem está na combinação de duas características tradicionalmente conflitantes na literatura: a escalabilidade computacional proporcionada pelos métodos aproximativos e a qualidade de representação decorrente dos métodos exatos. Dessa forma, a metodologia proposta busca aproveitar as vantagens de cada classe de técnicas, mitigando simultaneamente suas principais limitações. Enquanto os cortes de Benders fornecem uma aproximação computacionalmente eficiente para problemas de grande dimensão, os cortes Lagrangianos introduzem informações adicionais capazes de representar de maneira mais precisa as não convexidades induzidas pelas variáveis inteiras e pelas restrições condicionais.

Os resultados computacionais obtidos em instâncias de grande porte, representativas do planejamento da operação de sistemas hidrotérmicos reais, demonstraram a efetividade da estratégia proposta. Em comparação com abordagens baseadas exclusivamente em cortes de Benders e Benders Reforçados, a metodologia híbrida produziu soluções de menor custo esperado e políticas operativas significativamente mais próximas daquelas obtidas pela resolução exata do problema. Além disso, os experimentos evidenciaram que a abordagem mantém viabilidade computacional mesmo em cenários de elevada complexidade, nos quais a aplicação direta de métodos exatos se torna impraticável. Esses resultados demonstram que a metodologia proposta representa uma alternativa promissora para o tratamento de problemas estocásticos multiestágio lineares inteiros mistos de grande escala, conciliando qualidade de solução e eficiência computacional em um contexto onde tais objetivos são frequentemente antagônicos.

Como perspectivas para trabalhos futuros, destaca-se:

- O desenvolvimento de métodos mais eficientes para a identificação de soluções duais associadas a pontos extremos do conjunto dual ótimo. Embora os *Tighter Lagrangian Cuts* tenham demonstrado potencial para produzir aproximações de maior qualidade da função de recurso, sua construção ainda requer esforço computacional adicional. Nesse contexto, a investigação de estratégias capazes de explorar essas soluções duais de forma mais eficiente pode ampliar significativamente os ganhos de desempenho observados nos métodos SDDiP e DDiP.
- A extensão da metodologia híbrida de decomposição em duas etapas para outras classes de restrições envolvendo variáveis inteiras em problemas de planejamento. Um exemplo par-

tualmente relevante é a representação de políticas operativas dependentes de indicadores agregados do sistema, como procedimentos em que determinadas usinas termelétricas devem ser despachadas em níveis mínimos ou máximos quando o armazenamento energético global do sistema se encontra abaixo de um determinado limiar. Problemas dessa natureza apresentam características semelhantes às restrições condicionais estudadas nesta tese e constituem um ambiente promissor para a aplicação e evolução das técnicas propostas.

- A aplicação dos conceitos desenvolvidos nesta tese a outros problemas de Pesquisa Operacional. Em particular, destacam-se os problemas de gestão de estoques e planeamento da produção, nos quais decisões discretas induzem descontinuidades e não convexidades nas funções de valor. A adaptação dos conceitos de caracterização dual, cortes Lagrangianos reforçados e decomposição híbrida para esse contexto representa uma direção de pesquisa promissora.

Por fim, espera-se que os resultados apresentados contribuam para o avanço dos métodos de decomposição aplicados a problemas estocásticos de grande porte com variáveis inteiras, bem como para o desenvolvimento de ferramentas computacionais capazes de representar de forma mais adequada as complexidades operativas presentes em sistemas energéticos reais.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WOOD, A.; WOLLEMBERG, B. *Power Generation Operation and Control*. 3rd.. ed. NewYork, NY, USA: John Wiley and Sons, 2014.
- [2] MACEIRA, M. 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. *20th Power Systems Computation Conference (PSCC)*, 2018.
- [3] WOLFGANG, O. et al. Hydro reservoir handling in Norway before and after deregulation. *Energy*, v. 34, n. 10, p. 1642–1651, 2009. ISSN 0360-5442. 11th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360544209003119>>.
- [4] 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, 2016.
- [5] CÔTÉ, P.; ARSENAULT, R. Efficient Implementation of Sampling Stochastic Dynamic Programming Algorithm for Multireservoir Management in the Hydropower Sector. *Journal of Water Resources Planning and Management*, v. 145, n. 4, p. 05019005, 2019.
- [6] SHAPIRO, A. On a time consistency concept in risk averse multistage stochastic programming. *Oper. Res. Letters*, v. 37, n. 3, p. 143–147, 2009.
- [7] MACEIRA, M. et al. Chain of Optimization Models for Setting the Energy Dispatch and Spot Price in the Brazilian System. In: *Power System Computation Conference*. [S.l.: s.n.], 2002.
- [8] SANTOS, T. et al. Hourly pricing and day-ahead dispatch setting in Brazil: The dessem model. *Electric Power Systems Research*, v. 189, p. 106709, 12 2020.
- [9] JUNGER, M. et al. *50 Years of Integer Programming 1958-2008*. [S.l.]: Springer Berlin, 2009.
- [10] FISHER, M. L. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Manage. Sci.*, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 50, n. 12 Supplement, p. 1861–1871, dez. 2004. ISSN 0025-1909. Disponível em: <<http://dx.doi.org/10.1287/mnsc.1040.0263>>.
- [11] MORRISON, D. R. et al. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, v. 19, p. 79–102, 2016. ISSN 1572-5286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1572528616000062>>.
- [12] BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. Association for Computing Machinery, New York, NY, USA, v. 35, n. 3, p. 268–308, sep 2003. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/937503.937505>>.

- [13] BIRGE, J. R.; LOUVEAUX, F. *Introduction to Stochastic Programming*. New York, NY, USA: Springer-Verlag, 1997.
- [14] DINIZ, A. L. et al. Short/Mid-Term Hydrothermal Dispatch and Spot Pricing for Large-Scale Systems-the Case of Brazil. *20th Power Systems Computation Conference (PSCC)*, 2018.
- [15] Coordinador Eléctrico Nacional. Energy planning models in Chile. 2024. Disponível em: <<https://www.coordinador.cl>>.
- [16] FOSSO, O. et al. Generation scheduling in a deregulated system. The Norwegian case. *IEEE Transactions on Power Systems*, v. 14, n. 1, p. 75–81, 1999.
- [17] SHAWWASH, Z.; SIU, T.; RUSSEL, S. The BC Hydro short term hydro scheduling optimization model. In: *Proceedings of the 21st International Conference on Power Industry Computer Applications. Connecting Utilities. PICA 99. To the Millennium and Beyond (Cat. No.99CH36351)*. [S.l.: s.n.], 1999. p. 183–189.
- [18] ROMISCH, W.; SCHULTZ, R. Multistage Stochastic Integer Programs: An Introduction. In: GROTSCH, M.; KRUMKE, S. O.; RAMBAU, J. (Ed.). *Online Optimization of Large Scale Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 581–600. ISBN 978-3-662-04331-8. Disponível em: <https://doi.org/10.1007/978-3-662-04331-8_29>.
- [19] CRUZ, C. B. da; BRANDAO, L. C.; DINIZ, A. L. Water values and marginal costs for power generation planning under more restrictive constraints and a more detailed representation of hydro plants. *Electric Power Systems Research*, v. 235, p. 110819, 2024. ISSN 0378-7796. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378779624007053>>.
- [20] SCHÄFFER, L. E. et al. Modelling of Environmental Constraints for Hydropower Optimization Problems – a Review. In: *2020 17th International Conference on the European Energy Market (EEM)*. [S.l.: s.n.], 2020. p. 1–7.
- [21] Agência Nacional de Águas. Resolução N 193, DE 10 DE MAIO DE 2024. *Diário Oficial da União, Edição: 91, Seção: 1, Página: 146*. Disponível em: <<https://www.gov.br/ana/pt-br/legislacao/resolucoes/resolucoes-regulatorias/2024/193>>.
- [22] Agência Nacional de Águas. Resolução N 70, DE 19 DE ABRIL DE 2021. *Diário Oficial da União, Edição: 74, Seção 1, Página 158*. Disponível em: <<https://www.gov.br/ana/pt-br/legislacao/resolucoes/resolucoes-regulatorias/2021/70>>.
- [23] Agência Nacional de Águas. Resolução N 132, DE 10 DE OUTUBRO DE 2022. *Diário Oficial da União, Edição: 196, Seção 1, Página 63*. Disponível em: <<https://www.gov.br/ana/pt-br/legislacao/resolucoes/resolucoes-regulatorias/2022/132>>.
- [24] Agência Nacional de Águas. Resolução N 194, DE 10 DE OUTUBRO DE 2022. *Diário Oficial da União, Edição: 91, Seção: 1, Página: 146*. Disponível em: <<https://www.gov.br/ana/pt-br/legislacao/resolucoes/resolucoes-regulatorias/2024/194>>.
- [25] PEREIRA-BONVALLET, E. et al. Optimizing Hydrothermal Scheduling with Non-Convex Irrigation Constraints: Case on the Chilean Electricity System. *Energy Procedia*, v. 87, p. 132–140, 2016. ISSN 1876-6102. 5th International Workshop on Hydro Scheduling in Competitive Electricity Markets. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1876610215030246>>.

- [26] LITLABO, T. et al. Modelling overflow using mixed integer programming in short-term hydropower scheduling. *Energy Systems*, 2023.
- [27] SLYKE, R. M. V.; 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, 1969.
- [28] BIRGE, J. R. Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs. *Operations Research*, v. 33, n. 5, p. 989–1007, 1985.
- [29] 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, 1988. ISSN 0377-2217.
- [30] DEMPSTER, M.; THOMPSON, R. Parallelization and aggregation of nested Benders decomposition. *Operations Research*, v. 81, p. 163–188, 1998. Disponível em: <<https://doi.org/10.1023/A:1018996821817>>.
- [31] 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, Aug 2009. ISSN 0885-8950.
- [32] SHERALI, H. D.; LUNDAY, B. J. On generating maximal nondominated Benders cuts. *Annals of Operations Research*, v. 210, n. 1, p. 57–72, 2013. ISSN 1572-9338. Disponível em: <<http://dx.doi.org/10.1007/s10479-011-0883-6>>.
- [33] 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, 1991. ISSN 1436-4646. Disponível em: <<http://dx.doi.org/10.1007/BF01582895>>.
- [34] GJELSVIK, A.; MO, B.; HAUGSTAD, A. Long- and medium-term operations planning and stochastic modelling in hydro-dominated power systems based on stochastic dual dynamic programming. In: PARDALOS, P. M. et al. (Ed.). *Handbook of Power Systems I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 33–55. ISBN 978-3-642-02493-1. Disponível em: <https://doi.org/10.1007/978-3-642-02493-1_2>.
- [35] PHILPOTT, A.; DE MATOS, V. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of Operational Research*, v. 218, n. 2, p. 470–483, 2012. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221711010332>>.
- [36] SHAPIRO, A. et al. Risk neutral and risk averse Stochastic Dual Dynamic Programming method. *European Journal of Operational Research*, v. 224, n. 2, p. 375–391, 2013. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221712006455>>.
- [37] 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, 2015. ISSN 0377-0427. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377042715002794>>.
- [38] BRANDI, R. B. S. et al. A Convergence Criterion for Stochastic Dual Dynamic Programming: Application to the Long-Term Operation Planning Problem. *IEEE Trans. Power Syst.*, v. 33, n. 4, p. 3678–3690, 2018.

- [39] HOMEM-DE-MELLO, T.; DE MATOS, V. L.; FINARDI, E. C. Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Systems*, v. 2, p. 1–31, 2011.
- [40] SHAPIRO, A. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, v. 209, n. 1, p. 63–72, 2011. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221710005448>>.
- [41] CERISOLA, S.; C., J. L.; RAMOS, A. Stochastic dual dynamic programming applied to nonconvex hydrothermal models. *European Journal of Operational Research*, v. 218, p. 687–697, 05 2012.
- [42] STEEGER, G.; REBENNACK, S. Dynamic convexification within nested Benders decomposition using Lagrangian relaxation: An application to the strategic bidding problem. *European Journal of Operational Research*, v. 257, n. 2, p. 669–686, 2017. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037722171630621X>>.
- [43] LARA, C. L. et al. Deterministic electric power infrastructure planning: Mixed-integer programming model and nested decomposition algorithm. *European Journal of Operational Research*, v. 271, n. 3, p. 1037–1054, 2018. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221718304466>>.
- [44] LARA, C. L.; SIIROLA, J. D.; GROSSMANN, I. E. Electric power infrastructure planning under uncertainty: stochastic dual dynamic integer programming (SDDiP) and parallelization scheme. *Optimization and Engineering*, v. 21, p. 461–502, 2020. ISSN 1573-2924. Disponível em: <<https://doi.org/10.1007/s11081-019-09471-0>>.
- [45] COLONETTI, B. et al. Parallel Dual Dynamic Integer Programming for Large-Scale Hydrothermal Unit-Commitment. *IEEE Transactions on Power Systems*, v. 38, n. 3, p. 2926–2938, 2023.
- [46] 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, 2008.
- [47] BRANDAO, L. C. et al. A Data-Driven Representation of Aggregate Efficiency Curves of Hydro Units for the Mid-Term Hydrothermal Coordination Problem. *Electric Power Systems Research*, v. 212, p. 108511, 2022. ISSN 0378-7796. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378779622006162>>.
- [48] LAPORTE, G.; LOUVEAUX, F. V. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, v. 13, n. 3, p. 133–142, 1993. ISSN 0167-6377. Disponível em: <<https://www.sciencedirect.com/science/article/pii/016763779390002X>>.
- [49] ANGULO, G.; AHMED, S.; DEY, S. Improving the Integer L-Shaped Method. *INFORMS Journal on Computing*, v. 28, p. 483–499, 07 2016.
- [50] ZOU, J.; AHMED, S.; SUN, X. A. Stochastic dual dynamic integer programming. *Math. Program.*, v. 175, p. 461–502, 2019. ISSN 1436-4646. Disponível em: <<https://doi.org/10.1007/s10107-018-1249-5>>.

- [51] CARØE, C. C.; TIND, J. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, v. 83, n. 1, p. 451–464, 1998. ISSN 1436-4646.
- [52] PHILPOTT, A.; WAHID, F.; BONNANS, J. MIDAS: A mixed integer dynamic approximation scheme. *Mathematical Programming*, v. 181, 02 2019.
- [53] AHMED, S.; CABRAL, F. G.; COSTA, B. Freitas Paulo da. Stochastic Lipschitz dynamic programming. *Math. Program.*, Springer-Verlag, v. 191, n. 2, p. 755–793, feb 2022. ISSN 0025-5610. Disponível em: <<https://doi.org/10.1007/s10107-020-01569-z>>.
- [54] FEIZOLLAHI, M. J.; SUN, X.; AHMED, S. Exact Augmented Lagrangian Duality for Mixed Integer Linear Programming. *Mathematical Programming*, v. 161, p. 365–287, 01 2017.
- [55] ZHANG, S.; SUN, X. A. Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization. *Math. Program.*, v. 196, n. 1, p. 935–985, 2022. ISSN 1436-4646. Disponível em: <<https://doi.org/10.1007/s10107-022-01875-8>>.
- [56] LAAN, N. van der; ROMEIJNDERS, W. A Converging Benders' Decomposition Algorithm for Two-Stage Mixed-Integer Recourse Models. *Operations Research*, 05 2023.
- [57] FULLNER XU ANDY SUN, S. R. C. A new framework to generate Lagrangian cuts in multistage stochastic mixed-integer programming. Optimization online. 2024.
- [58] DENG, H.; XIE, W. *On the ReLU Lagrangian Cuts for Stochastic Mixed Integer Programming*. 2024. Disponível em: <<https://arxiv.org/abs/2411.01229>>.
- [59] YANG, H.; YANG, H. Globally converging algorithm for multistage stochastic mixed-integer programs via enhanced lagrangian cuts. Optimization online. 2025.
- [60] OWEN, S. M. J. H. On the Value of Binary Expansions for General Mixed-Integer Linear Programs. *Operations Research*, v. 50, n. 5, p. 810–819, 2002. Disponível em: <<https://doi.org/10.1287/opre.50.5.810.370>>.
- [61] YANG, H. et al. Multi-Period Power System Risk Minimization Under Wildfire Disruptions. *IEEE Transactions on Power Systems*, v. 39, n. 5, p. 6305–6318, 2024.
- [62] BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, v. 4, n. 1, p. 238–252, 1962. ISSN 0945-3245. Disponível em: <<http://dx.doi.org/10.1007/BF01386316>>.
- [63] ZOU, J.; AHMED, S.; SUN, X. A. Multistage Stochastic Unit Commitment Using Stochastic Dual Dynamic Integer Programming. *IEEE Transactions on Power Systems*, v. 34, n. 3, p. 1814–1823, 2019.
- [64] QUEZADA, F.; GICQUEL, C.; KEDAD-SIDHOUM, S. Combining Polyhedral Approaches and Stochastic Dual Dynamic Integer Programming for Solving the Uncapacitated Lot-Sizing Problem Under Uncertainty. *INFORMS Journal on Computing*, v. 34, 12 2021.
- [65] HJELMELAND, M. N. et al. Nonconvex Medium-Term Hydropower Scheduling by Stochastic Dual Dynamic Integer Programming. *IEEE Transactions on Sustainable Energy*, v. 10, n. 1, p. 481–490, 2019.

- [66] SUN, X. et al. Multistage Dynamic Planning of Integrated Hydrogen-Electrical Microgrids Under Multiscale Uncertainties. *IEEE Transactions on Smart Grid*, v. 14, n. 5, p. 3482–3498, 2023.
- [67] CAO, B. et al. Multi-stage stochastic long-term planning of grid-connected hydrogen-based energy system based on improved SDDIP. *IET Generation, Transmission & Distribution*, v. 17, p. n/a–n/a, 06 2023.
- [68] SUN, X. et al. Seasonal operation planning of hydrogen-enabled multi-energy microgrids through multistage stochastic programming. *Journal of Energy Storage*, v. 85, p. 111125, 2024. ISSN 2352-152X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2352152X24007096>>.
- [69] SHAHIDEHPOUR, M. et al. Multi-Period Active Distribution Network Planning Using Multi-Stage Stochastic Programming and Nested Decomposition by SDDIP. *IEEE Transactions on Power Systems*, PP, p. 1–1, 10 2020.
- [70] QIAN, T.; WU, Q.; HU, Q. Efficient power system year-round hourly operation simulation based on multi-stage Stochastic Dual Dynamic Integer Programming. *International Journal of Electrical Power and Energy Systems*, v. 161, p. 110158, 2024. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S014206152400379X>>.
- [71] QUEZADA, F.; GICQUEL, C.; KEDAD-SIDHOUM, S. A stochastic dual dynamic integer programming based approach for remanufacturing planning under uncertainty. *International Journal of Production Research*, v. 61, p. 1–21, 09 2022.
- [72] LI, Z. et al. Hydropower preventive maintenance scheduling in a deregulated market: A multi-stage stochastic programming approach. *Electric Power Systems Research*, v. 221, p. 109379, 2023. ISSN 0378-7796. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378779623002687>>.
- [73] YANG, H. et al. Multistage stochastic program for mitigating power system risks under wild-fire disruptions. *Electric Power Systems Research*, v. 234, p. 110773, 2024. ISSN 0378-7796. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037877962400659X>>.
- [74] AHMED, S.; TAWARMALANI, M.; SAHINIDIS, N. A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, v. 100, p. 355–377, 06 2004.
- [75] TRAPP, A. C.; PROKOPYEV, O. A.; SCHAEFER, A. J. On a Level-Set Characterization of the Value Function of an Integer Program and Its Application to Stochastic Programming. *Operations Research*, v. 61, n. 2, p. 498–511, 2013. Disponível em: <<http://dx.doi.org/10.1287/opre.1120.1156>>.
- [76] BALAS, E. Disjunctive Programming. In: HAMMER, P.; JOHNSON, E.; KORTE, B. (Ed.). *Discrete Optimization II*. Elsevier, 1979, (Annals of Discrete Mathematics, v. 5). p. 3–51. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S016750600870342X>>.
- [77] SEN, S.; SHERALI, H. Decomposition with branch-and-cut approaches for two-stage stochastic integer programming. *Mathematical Programming*, v. 106, p. 203–223, 06 2006.
- [78] NTAIMO, L.; SEN, S. A branch-and-cut algorithm for two-stage stochastic mixed-binary programs with continuous first-stage variables. *International Journal of Computational Science and Engineering*, v. 3, p. 232–241, 04 2007.

- [79] LI, C.; GROSSMANN, I. A generalized Benders decomposition-based branch and cut algorithm for two-stage stochastic programs with nonconvex constraints and mixed-binary first and second stage variables. *Journal of Global Optimization*, v. 75, 10 2019.
- [80] GADE, D.; KUCUKYAVUZ, S.; SEN, S. Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, v. 144, p. 39–64, 04 2014.
- [81] ZHANG, M.; KUCUKYAVUZ, S. Finitely Convergent Decomposition Algorithms for Two-Stage Stochastic Pure Integer Programs. *SIAM Journal on Optimization*, v. 24, n. 4, p. 1933–1951, 2014. Disponível em: <<https://doi.org/10.1137/13092678X>>.
- [82] MULVEY, J.; RUSZCZYNSKI, A. A New Scenario Decomposition Method for Large-Scale Stochastic Optimization. *Operations Research*, v. 43, 06 1995.
- [83] CARØE, C. C.; SCHULTZ, R. Dual decomposition in stochastic integer programming. *Operations Research Letters*, v. 24, n. 1, p. 37–45, 1999. ISSN 0167-6377. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167637798000509>>.
- [84] CAROE, C. C.; RUSZCZYNSKI, A.; SCHULTZ, R. Unit Commitment under Uncertainty via Two-Stage Stochastic Programming. In: *Proceedings of NOAS*. [s.n.], 1997. Disponível em: <<https://api.semanticscholar.org/CorpusID:9069155>>.
- [85] LARSEN, E. et al. Fast Continuous and Integer L-Shaped Heuristics Through Supervised Learning. *INFORMS Journal on Computing*, v. 36, 08 2023.
- [86] BIRGE, J. R.; LOUVEAUX, F. *Introduction to Stochastic Programming*. [S.l.]: Springer Science & Business Media, 2011. ISBN 978-1-4614-0237-4.
- [87] HELD, M.; WOLFE, P.; CROWDER, H. P. Validation of subgradient optimization. *Mathematical Programming*, v. 6, p. 62–88, 1974. Disponível em: <<https://api.semanticscholar.org/CorpusID:206797746>>.
- [88] KELLEY, J. E. The Cutting-Plane Method for Solving Convex Programs. *Journal of the Society for Industrial and Applied Mathematics*, Society for Industrial and Applied Mathematics, v. 8, n. 4, p. 703–712, 1960. ISSN 03684245. Disponível em: <<http://www.jstor.org/stable/2099058>>.
- [89] LEMARECHAL, C.; NEMIROVSKI, A.; NESTEROV, Y. New variants of bundle methods. *Mathematical Programming*, v. 69, p. 111–147, 1995. Disponível em: <<https://api.semanticscholar.org/CorpusID:20299105>>.
- [90] LI, Z. et al. Retrofit Planning and Flexible Operation of Coal-Fired Units Using Stochastic Dual Dynamic Integer Programming. *IEEE Transactions on Power Systems*, v. 39, n. 1, p. 2154–2169, 2024.
- [91] BRANDAO, L. C.; DINIZ, A. L.; SIMONETTI, L. Accelerating Dual Dynamic Programming for Stochastic Hydrothermal Coordination Problems. *20th Power Systems Computation Conference (PSCC)*, 2018.

APÊNDICE A – FORMULAÇÃO DO ESTUDO DE CASO “TUTORIAL”

A formulação do caso base utilizado no estudo tutorial, discutido na Seção 5.1, é apresentada a seguir.

$$\min \sum_{i=1}^5 \sum_{t=1}^2 \sum_{j=1}^3 dp_{t,j} \cdot c_i \cdot gt_{i,t,j}$$

s.a.

$$\sum_{i=1}^5 gt_{i,1,1} + gh_{1,1} = 25 \quad (A.1)$$

$$\sum_{i=1}^5 gt_{i,1,2} + gh_{1,2} = 19 \quad (A.2)$$

$$\sum_{i=1}^5 gt_{i,1,3} + gh_{1,3} = 15 \quad (A.3)$$

$$\sum_{i=1}^5 gt_{i,2,1} + gh_{2,1} = 45 \quad (A.4)$$

$$\sum_{i=1}^5 gt_{i,2,2} + gh_{2,2} = 39 \quad (A.5)$$

$$\sum_{i=1}^5 gt_{i,2,3} + gh_{2,3} = 35 \quad (A.6)$$

$$v_1 + \sum_{j=1}^3 \sigma_{1,j}(qt_{1,j} + qs_{1,j}) = v_0 + 310.6944 \quad (A.7)$$

$$v_2 + \sum_{j=1}^3 \sigma_{2,j}(qt_{2,j} + qs_{2,j}) = v_1 + 107.136 \quad (A.8)$$

$$gh_{t,j} \leq AHPF_k(v_{t-1}, v_t, qt_{t,j}), \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\}; k \in \{1, \dots, 6\} \quad (A.9)$$

$$0 \leq gh_{t,j} \leq 46, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.10)$$

$$0 \leq qt_{t,j} \leq 207, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.11)$$

$$0 \leq qs_{t,j} \leq 999, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.12)$$

$$0 \leq v_t \leq 672, \quad \forall t \in \{1, 2\} \quad (A.13)$$

$$0 \leq gt_{1,t,j} \leq 8, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.14)$$

$$0 \leq gt_{2,t,j} \leq 2, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.15)$$

$$0 \leq gt_{3,t,j} \leq 2, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.16)$$

$$0 \leq gt_{4,t,j} \leq 2, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.17)$$

$$0 \leq gt_{5,t,j} \leq 40, \quad \forall t \in \{1, 2\}; j \in \{1, 2, 3\} \quad (A.18)$$

O caso considera dois períodos (t), três patamares de carga (j), cinco usinas térmicas (i)

e uma única usina hidrelétrica, as variáveis de decisão do problema são:

$gt_{i,t,j}$: é a geração, em MW , térmica da usina i , período t e patamar j , cujos custos marginais em $R\$/MWh$ são dados por:

$$c = [31.17, 310.41, 978.10, 1700.00, 6524.05].$$

$gh_{t,j}$: é a geração, em MW , hidráulica no período t e patamar j ;

v_t : é o volume armazenado útil, em hm^3 , no período t , sendo $v_0 = 171.6288$ o volume inicial do estudo;

$qt_{t,j}$: é o turbinamento, em m^3/s , no período t e patamar j ;

$qs_{t,j}$: é o vertimento, em m^3/s , no período t e patamar j ;

As durações dos patamares de carga são:

$$dp_{1,1} = 184h, dp_{1,2} = 208h, dp_{1,3} = 352h$$

$$dp_{2,1} = 168h, dp_{2,2} = 198h, dp_{2,3} = 378h$$

Constantes de conversão de unidades:

$\sigma_{t,j}$: constante para converter vazões vertidas/turbinadas de m^3/s em volumes vertidos/turbinados em hm^3 , dada por:

$$\sigma_{t,j} = \frac{3600 * dp_{t,j}}{1 \times 10^6}$$

Cada variável de decisão possui limites mínimos e máximos conforme definido em (A.10)-(A.18). As restrições deste caso são:

- restrições de atendimento à demanda energética, dadas por (A.1)–(A.6), cujos limites estão dados em MW .
- restrições de balanço hídrico (A.7)-(A.8) para a usina hidrelétrica, cujos limites estão dados em hm^3 ;
- conjuntos de restrições da função de produção hidráulica (AHPF) (A.9), detalhada mais a frente.

Cada função de produção hidráulica é modelada por meio de uma formulação linear por partes, composta por seis restrições lineares, que relacionam a geração hidráulica ao volume inicial e final do reservatório no período e à vazão turbinada. Cada conjunto de restrições é replicado para cada período e patamar e são apresentados nas Equações (A.19)-(A.24).

$$AHPF_1(v_{t-1}, v_t, qt_{t,j}) = 0.0526022810935 \frac{v_{t-1} + v_t}{2} + 0.105158616525 qt_{t,j} \quad (A.19)$$

$$AHPF_2(v_{t-1}, v_t, qt_{t,j}) = 0.0346020062355 \frac{v_{t-1} + v_t}{2} + 0.11978111303 qt_{t,j} \quad (A.20)$$

$$AHPF_3(v_{t-1}, v_t, qt_{t,j}) = 0.0285883368365 \frac{v_{t-1} + v_t}{2} + 0.12955132292 qt_{t,j} \quad (A.21)$$

$$AHPF_4(v_{t-1}, v_t, qt_{t,j}) = 0.0190480483055 \frac{v_{t-1} + v_t}{2} + 0.152801179235 qt_{t,j} \quad (A.22)$$

$$AHPF_5(v_{t-1}, v_t, qt_{t,j}) = 0.0036686838911 \frac{v_{t-1} + v_t}{2} + 0.202773843375 qt_{t,j} \quad (A.23)$$

$$AHPF_6(v_{t-1}, v_t, qt_{t,j}) = 0.0 \frac{v_{t-1} + v_t}{2} + 0.21866853279 qt_{t,j} \quad (A.24)$$


```

&*****
& Descrição colunas:
&*****
& CodUsih: Código da usina hidrelétrica
& NomeUsih: Nome da usina hidrelétrica
& Período: Código do Intervalo de tempo na discretização temporal adotada
& Cenário: Índice do Cenário
& INo: Índice do nó na árvore de cenários
& Pat: Índice do patamar de carga no período
& DuracaoPat: Duração do patamar de carga
& VarmInic: Volume armazenado no início do período operativo da usina
& VarmFinal: Volume armazenado no final do período operativo da usina
& VolInicPer: Volume armazenado no início do período operativo da usina em percentual do volume útil
& VarmFinPer: Volume armazenado no final do período operativo da usina em percentual do volume útil
& Turbinamento: Vazão turbinada no ponto de operação da usina
& Vertimento: Vazão vertida no ponto de operação da usina
& GHidr: Geração hidroelétrica da usina
&*****
&CodUsih;NomeUsih ;Período;Cenário;INo ;Pat ;DuracaoPat;VarmInic ;VarmFinal ;VolInicPer;VarmFinPer;Turbinamento ;Vertimento ;GHidr ;
&- ; - ; - ; - ; - ;h ;hm^3 ;hm^3 ;% ;% ;m^3/s ;m^3/s ;MW ;
&IIIIIIII;SSSSSSSS;IIIIII;IIIIII;IIIIII;IIIIII;FFFFFFF;FFFFFFF;FFFFFFF;FFFFFFF;FFFFFFF;FFFFFFF;FFFFFFF;FFFFFFF;
1;CAMARGOS ; 1; 1; 1; 1; 184.00; 171.63; 365.17; 25.5400; 54.3402; 69.12; 0.00; 15.00
1;CAMARGOS ; 1; 1; 1; 2; 208.00; 171.63; 365.17; 25.5400; 54.3402; 50.30; 0.00; 11.00
1;CAMARGOS ; 1; 1; 1; 3; 352.00; 171.63; 365.17; 25.5400; 54.3402; 26.60; 0.00; 5.82
1;CAMARGOS ; 2; 1; 2; 1; 168.00; 365.17; 0.00; 54.3402; 0.0000; 206.81; 0.00; 31.09
1;CAMARGOS ; 2; 1; 2; 2; 198.00; 365.17; 0.00; 54.3402; 0.0000; 189.36; 0.00; 29.00
1;CAMARGOS ; 2; 1; 2; 3; 378.00; 365.17; 0.00; 54.3402; 0.0000; 155.97; 0.00; 25.00

```

Figura 27 – Resultados da operação das usinas hidráulicas para o problema “tutorial-no-constr” com a representação “binary-storage”, considerando 23 variáveis binárias, resolvido como MILP-global

```

&*****
& Descrição colunas:
&*****
& CodSubm: Código do Submercado
& Período: Código do Intervalo de tempo na discretização temporal adotada
& Cenário: Índice do Cenário
& Pat: Índice do patamar de carga no período
& DuracaoPat: Duração do patamar de carga
& Demanda: Valor da Demanda
& GHidrSub: Geração hidroelétrica no submercado
& GTerSub: Geração térmica total no submercado
&*****
&CodSubm ;Período;Cenário;Pat ;DuracaoPat;Demanda ;GHidrSub ;GTerSub ;
&- ; - ; - ; - ;h ;MW ;MW ;MW ;
&IIIIIIII;IIIIII;IIIIII;IIIIII;FFFFFFF;FFFFFFF;FFFFFFF;FFFFFFF;
1; 1; 1; 1; 184.00; 25.00; 15.00; 10.00
1; 1; 1; 2; 208.00; 19.00; 11.00; 8.00
1; 1; 1; 3; 352.00; 15.00; 5.82; 9.18
1; 2; 1; 1; 168.00; 45.00; 31.09; 13.91
1; 2; 1; 2; 198.00; 39.00; 29.00; 10.00
1; 2; 1; 3; 378.00; 35.00; 25.00; 10.00

```

Figura 28 – Resultados da operação energética para o problema “tutorial-no-constr” com a representação “binary-storage”, considerando 23 variáveis binárias, resolvido como MILP-global

Para o caso “tutorial-no-constr” com a representação “binary-storage” considerando 5 variáveis binárias e resolvido através do MILP-global, os resultados da operação são mostrados nas Figuras 29 e 30.

B.1.2 Cortes de Benders

A Figura 31 mostra os resultados do processo de convergência, para o caso “tutorial-no-constr”, utilizando o algoritmo DDiP com os cortes de Benders e representação “binary-storage”.

```

&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIIIIIIIIIIIII;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;
0;          30.24576200; 37009418.60443800; 1223622.27359500; 0.02; 0.02
1; 146121.18552100; 4043877.69871900; 26.67482100; 0.02; 0.04
2; 489462.01078600; 1951985.76409600; 2.98802300; 0.02; 0.06
3; 1948149.22147100; 2171361.75161100; 0.11457700; 0.02; 0.08
4; 1950121.88038600; 1999113.30747400; 0.02512200; 0.02; 0.11
5; 1950765.13920000; 1950765.13920000; 0.00000000; 0.03; 0.14

```

Figura 31 – Convergência do problema “tutorial-no-constr” resolvido pela DDiP com cortes de Benders com representação “binary-storage” com 23 variáveis de estado.

A Figura 32 mostra os resultados do processo de convergência, para o caso “tutorial-no-constr”, utilizando o algoritmo DDP com os cortes de Benders e representação “continuous-storage”.

```

&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIIIIIIIIIIIII;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFF;
0;          30.24575600; 37009399.18389400; 1223621.89919800; 0.01; 0.01
1; 146121.09542700; 4043882.68811400; 26.67487300; 0.01; 0.02
2; 489461.07964800; 1951985.76831900; 2.98803100; 0.01; 0.03
3; 1948149.20435400; 2171360.98782400; 0.11457600; 0.01; 0.05
4; 1950121.85255900; 1999115.39904400; 0.02512300; 0.01; 0.06
5; 1950765.12849800; 1950765.12849800; 0.00000000; 0.01; 0.07

```

Figura 32 – Convergência do problema “tutorial-no-constr” resolvido pela DDP com cortes de Benders com representação “continuous-storage”.

Verificou-se também que a operação obtida por ambas execuções é idêntica à operação obtida pelo LP-global detalhada na seção B.1.1.

B.1.3 Cortes de Benders Reforçados

A Figura 33 mostra os resultados do processo de convergência utilizando o algoritmo DDIP com os cortes de Benders Reforçados e representação “binary-storage”.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter      ;TempoAcum      ;
&-         ;$         ;$         ;-         ;s         ;s         ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          30.24576200; 37009418.60443900; 1223622.27359500; 0.02; 0.02
1; 146121.18552100; 4043877.69871900; 26.67482100; 0.03; 0.04
2; 489462.01078600; 1951985.76409700; 2.98802300; 0.02; 0.07
3; 1948149.22147100; 2171361.75161100; 0.11457700; 0.03; 0.10
4; 1950121.88038600; 1999113.30747400; 0.02512200; 0.03; 0.12
5; 1950765.13920000; 1950765.13920000; 0.00000000; 0.03; 0.16
```

Figura 33 – Convergência do problema “tutorial-no-constr” resolvido pela DDIP com cortes de Benders Reforçados com representação “binary-storage” com 23 variáveis de estado.

A Figura 34 mostra os resultados do processo de convergência utilizando o algoritmo DDP com os cortes de Benders Reforçados e representação “continuous-storage”.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter      ;TempoAcum      ;
&-         ;$         ;$         ;-         ;s         ;s         ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          30.24575600; 37009399.18389400; 1223621.89919800; 0.01; 0.01
1; 146121.09542700; 4043882.68811400; 26.67487300; 0.01; 0.02
2; 489461.07964800; 1951985.76831900; 2.98803100; 0.01; 0.03
3; 1948149.20435400; 2171360.98782400; 0.11457600; 0.01; 0.04
4; 1950121.85255900; 1999115.39904400; 0.02512300; 0.01; 0.05
5; 1950765.12849800; 1950765.12849800; -0.00000000; 0.01; 0.06
```

Figura 34 – Convergência do problema “tutorial-no-constr” resolvido pela DDP com cortes de Benders Reforçados com representação “continuous-storage”

Verificou-se também que a operação obtida por ambas execuções é idêntica à operação obtida pelo LP-global detalhada na seção B.1.1.

B.1.4 Cortes Inteiros de Otimalidade

O processo de convergência do problema “tutorial-no-constr” resolvido com cortes Inteiros de Otimalidade, com apenas 5 variáveis binárias na representação do volume, está mostrado na Figura 35.

[illegible]

Figura 35 – Convergência do problema “tutorial-no-constr” resolvido pela DDiP com cortes de Inteiros de Otimalidade, com apenas 5 variáveis binárias na representação do volume

Com relação a operação, foi semelhante à operação obtida pelo método MILP-global equivalente (com 5 variáveis binárias). As Figuras 36 e 37 detalham cada uma delas.

```

*****
& Descrição colunas:
*****
& CodUsih: Código da usina hidrelétrica
& NomeUsih: Nome da usina hidrelétrica
& Período: Código do Intervalo de tempo na discretização temporal adotada
& Cenário: Índice do Cenário
& INo: Índice do nó na árvore de cenários
& Pat: Índice do patamar de carga no período
& DuracaoPat: Duração do patamar de carga
& VarmInic: Volume armazenado no início do período operativo da usina
& VarmFinal: Volume armazenado no final do período operativo da usina
& VolInicPer: Volume armazenado no início do período operativo da usina em percentual do volume útil
& VarmFinPer: Volume armazenado no final do período operativo da usina em percentual do volume útil
& Turbinamento: Vazão turbinada no ponto de operação da usina
& Vertimento: Vazão vertida no ponto de operação da usina
& GHidr: Geração hidroelétrica da usina
*****
&CodUsih;NomeUsih;Período;Cenário;INo;Pat;DuracaoPat;VarmInic;VarmFinal;VolInicPer;VarmFinPer;Turbinamento;Vertimento;GHidr
& ; ; ; ; ; ; ; ; ; ; ; ; ; ;
&IIIIIIIIIISSSSSSSSSS;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII;IIIIIIII
&1;CAMARGOS;1;1;1;184.00;171.63;390.19;25.5400;58.0645;59.45;0.00;13.00
&1;CAMARGOS;1;1;2;208.00;171.63;390.19;25.5400;58.0645;32.01;0.00;7.00
&1;CAMARGOS;1;1;3;352.00;171.63;390.19;25.5400;58.0645;22.71;0.00;4.97
&1;CAMARGOS;2;1;2;168.00;390.19;0.00;58.0645;0.0000;206.81;0.00;31.52
&1;CAMARGOS;2;2;2;198.00;390.19;0.00;58.0645;0.0000;199.50;0.00;30.65
&1;CAMARGOS;2;3;378.00;390.19;0.00;58.0645;0.0000;169.05;0.00;27.00

```

Figura 36 – Resultados da operação das usinas hidráulicas para o problema “tutorial-no-constr”, com 5 variáveis binárias na representação do volume, resolvido com cortes Inteiros de Otimalidade


```

&*****
& Descrição colunas:
&*****
& CodSubm: Código do Submercado
& Período: Código do Intervalo de tempo na discretização temporal adotada
& Cenário: Índice do Cenário
& Pat: Índice do patamar de carga no período
& DuracaoPat: Duração do patamar de carga
& Demanda: Valor da Demanda
& GHidrSub: Geração hidroelétrica no submercado
& GTerSub: Geração térmica total no submercado
&*****;*****;*****;*****;*****;*****;*****;*****;*****;*****;
&CodSubm;Período;Cenário;Pat;DuracaoPat;Demanda;GHidrSub;GTerSub;
& ;-;-;-;h;MW;MW;MW;
&IIIIIIII;IIIIII;IIIIII;IIIIII;FFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
1; 1; 1; 1; 184.00; 25.00; 15.00; 10.00
1; 1; 1; 2; 208.00; 19.00; 9.00; 10.00
1; 1; 1; 3; 352.00; 15.00; 6.42; 8.58
1; 2; 1; 1; 168.00; 45.00; 31.15; 13.85
1; 2; 1; 2; 198.00; 39.00; 29.73; 9.27
1; 2; 1; 3; 378.00; 35.00; 25.00; 10.00

```

Figura 37 – Resultados da operação energética para o problema “tutorial-no-constr”, com 5 variáveis binárias na representação do volume, resolvido com cortes Inteiros de Otimalidade

B.1.5 Cortes de Lagrange

O processo de convergência do problema “tutorial-no-constr” resolvido pelo método de cortes de Lagrange está mostrado na Figura 38, com 23 variáveis binárias na representação do volume.

```
&*****
& Nome do Modelo: Libs Customizado Versão: 0.0.1
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;
&Iter;ZInf;ZSup;GAP;TempoIter;TempoAcum;
&-;S;-;S;-;S;-;S;-;
&IIIIIIIIIIIIII;FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF;
0;30.24576200;37009418.60443900;1223622.27359500;0.01;0.01
1;146121.18552100;4043877.69871900;26.67482100;0.02;0.03
2;489462.01078700;1951985.76409700;2.98802300;0.02;0.05
3;1948149.22147100;2171361.75161100;0.11457700;0.01;0.06
4;1950121.88038600;1999113.30747400;0.02512200;0.02;0.08
5;1950765.13920000;1950765.13920000;0.00000000;0.02;0.10
```

Figura 38 – Convergência do problema “tutorial-no-constr” resolvido pela DDiP com cortes de Lagrange

Com relação a operação, foi igual a operação obtida pelo método MILP-global equivalente conforme mostrado na seção B.1.1.

B.2.2 Cortes de Benders

A Figura 41 mostra os resultados do processo de convergência utilizando o algoritmo DDIP com os cortes de Benders e representação “binary-storage”.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          30.24576200; 37009418.60443800; 1223622.27359500; 0.02; 0.02
1; 146121.18552100; 4648797.21770800; 30.81466900; 0.03; 0.05
2; 489462.01078600; 2546653.55353600; 4.20296500; 0.02; 0.08
3; 1948149.22147100; 2931486.20899800; 0.50475400; 0.02; 0.10
4; 1950121.88038600; 2720719.47487500; 0.39515400; 0.02; 0.13
5; 1950765.13920000; 2660646.28603100; 0.36389900; 0.03; 0.16
```

Figura 41 – Convergência do problema “tutorial-cond-constr” resolvido pela DDIP com cortes de Benders com representação “binary-storage”

A Figura 42 mostra os resultados do processo de convergência utilizando o algoritmo DDP com os cortes de Benders e representação “continuous-storage”.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          30.24575600; 37009399.18389400; 1223621.89919800; 0.01; 0.01
1; 146121.09542700; 4648801.71745300; 30.81472000; 0.02; 0.03
2; 489461.07964800; 2546653.15921200; 4.20297400; 0.02; 0.05
3; 1948149.20435400; 2931485.44521100; 0.50475400; 0.02; 0.07
4; 1950121.85255900; 2720722.07363200; 0.39515500; 0.02; 0.08
5; 1950765.12849800; 2660647.28550100; 0.36389900; 0.02; 0.10
```

Figura 42 – Convergência do problema “tutorial-cond-constr” resolvido pela DDP com cortes de Benders com representação “continuous-storage”

A Figura 43 mostra a operação hidráulica do problema resolvido com os cortes de Benders, enquanto a Figura 44 mostra a operação energética.

B.2.3 Cortes de Benders Reforçados

A Figura 45 mostra os resultados do processo de convergência utilizando o algoritmo DDIP com os cortes de Benders Reforçados e representação “binary-storage”.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          30.24576200; 37009418.60443900; 1223622.27359500; 0.02; 0.02
1; 146121.18552100; 4648797.21770800; 30.81466900; 0.03; 0.04
2; 590562.71911300; 1954030.56434800; 2.30876000; 0.03; 0.07
3; 1948149.22147100; 2931486.20899800; 0.50475400; 0.03; 0.10
4; 1952445.92006600; 1952445.92006700; 0.00000000; 0.04; 0.13
```

Figura 45 – Convergência do problema “tutorial-cond-constr” resolvido pela DDP com cortes de Benders Reforçados com representação “binary-storage”

A Figura 46 mostra os resultados do processo de convergência utilizando o algoritmo DDP com os cortes de Benders Reforçados e representação “continuous-storage”.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          30.24575600; 37009399.18389400; 1223621.89919800; 0.01; 0.01
1; 146121.09542700; 4648801.71745300; 30.81472000; 0.02; 0.04
2; 590562.08147400; 1954030.55145200; 2.30876400; 0.02; 0.06
3; 1948149.20435400; 2931485.44521100; 0.50475400; 0.02; 0.09
4; 1952445.91551200; 1952445.91551200; -0.00000000; 0.02; 0.11
```

Figura 46 – Convergência do problema “tutorial-cond-constr” resolvido pela DDP com cortes de Benders Reforçados com representação “continuous-storage”

A Figura 47 mostra a operação hidráulica do problema resolvido com os cortes de Benders Reforçados. A Figura 48 mostra a operação energética.

B.2.4 Cortes Inteiros de Otimalidade (5 variáveis binárias)

A Figura 49 mostra os resultados do processo de convergência utilizando o algoritmo DDiP com os cortes Inteiros de Otimalidade.

```

*****
& Descrição colunas:
*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
*****
&Iter          ;ZInf          ;ZSup          ;GAP          ;TempoIter      ;TempoAcum      ;
&-            ;$            ;$            ;-            ;s              ;s              ;
&IIIIIIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFF;FFFFFFFFFFFFFFF;
0;              31.50396700; 40701771.48322800; 1291956.01358300; 0.01; 0.01
1;              33.74213300; 47227739.03126200; 1399665.68117500; 0.02; 0.04
2;              35.98436900; 53984760.65344300; 1500227.08078400; 0.02; 0.06
3;              38.22660500; 60883291.21348100; 1592693.17979500; 0.02; 0.09
4;              40.46884100; 67781821.77351999; 1674912.83647700; 0.02; 0.11
5;              42.73052600; 74680352.35300700; 1747703.96524700; 0.02; 0.13
6;              44.99730300; 81578882.93758599; 1812971.71690500; 0.02; 0.16
7;              47.26408000; 88477413.52216600; 1871979.00816200; 0.03; 0.18
8;              49.54306000; 95375944.11894800; 1925111.08237900; 0.03; 0.21
9;              51.82699000; 102274474.72068000; 1973381.49376800; 0.03; 0.24
10;             7279.39969100; 34183054.06908300; 4694.86168100; 0.03; 0.27
11;             24623.66645100; 27674433.02597400; 1122.89570700; 0.02; 0.29
12;             43230.24196100; 21167074.29161600; 488.63580400; 0.01; 0.31
13;             65732.36699100; 14663611.10677700; 222.08052800; 0.01; 0.32
14;             99412.27485600; 8202980.11417600; 81.51476100; 0.01; 0.34
15;             136800.56917800; 5114318.04794500; 36.38521000; 0.01; 0.35
16;             176405.40960200; 3507996.84620100; 18.88599400; 0.01; 0.37
17;             491898.84377400; 2542048.70225900; 4.16782800; 0.02; 0.39
18;             1425924.26436500; 2488007.93066200; 0.74483900; 0.02; 0.40
19;             2488007.93066200; 2488007.93066200; 0.00000000; 0.02; 0.42

```

Figura 49 – Convergência do problema “tutorial-cond-constr” com 5 variáveis binárias na representação do volume resolvido pela DDiP com cortes Inteiros de Otimalidade

As Figuras 50 e 51 mostram os valores para a operação hidráulica e energética do problema.

[illegible]

Figura 50 – Resultados da operação das usinas hidráulicas para o problema “tutorial-cond-constr” com 5 variáveis binárias para representar o volume, resolvido com a metodologia de cortes Inteiros de Otimalidade


```

&*****
& Descrição colunas:
&*****
& CodSubm: Código do Submercado
& Período: Código do Intervalo de tempo na discretização temporal adotada
& Cenário: Índice do Cenário
& Pat: Índice do patamar de carga no período
& DuracaoPat: Duração do patamar de carga
& Demanda: Valor da Demanda
& GHidrSub: Geração hidroelétrica no submercado
& GTerSub: Geração térmica total no submercado
&*****;*****;*****;*****;*****;*****;*****;*****;*****;*****;
&CodSubm ;Período;Cenário;Pat ;DuracaoPat;Demanda ;GHidrSub ;GTerSub ;
& ; - ; - ; - ;h ;MW ;MW ;MW ;
&IIIIIIII;IIIIII;IIIIII;IIIIII;FFFFFFFF;FFFFFFFFFFFFFF;FFFFFFFFFFFFFF;FFFFFFFFFFFFFF;
1; 1; 1; 1; 184.00; 25.00; 13.00; 12.00
1; 1; 1; 2; 208.00; 19.00; 7.00; 12.00
1; 1; 1; 3; 352.00; 15.00; 4.97; 10.03
1; 2; 1; 1; 168.00; 45.00; 31.52; 13.48
1; 2; 1; 2; 198.00; 39.00; 30.65; 8.35
1; 2; 1; 3; 378.00; 35.00; 27.00; 8.00

```

Figura 51 – Resultados da operação energética para o problema “tutorial-cond-constr” com 5 variáveis binárias para representar o volume, resolvido com a metodologia de cortes Inteiros de Otimalidade

B.2.5 Cortes de Lagrange

O processo de convergência do problema “tutorial-cond-constr” resolvido pelo método de cortes de Lagrange está mostrado na Figura 52;

```
&*****
& Nome do Modelo: Libs Customizado Versão: 0.0.1
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s             ;s             ;
&IIIIIIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFF;FFFFFFFFFFFFFF;
0;          30.24576200; 37009418.60443900; 1223622.27359500; 0.01; 0.01
1;          146121.18552100; 4648797.21770800; 30.81466900; 0.03; 0.04
2;          166229.46366500; 3832472.04427700; 22.05531100; 0.07; 0.11
3;          777353.00881900; 2041541.68527800; 1.62627400; 0.05; 0.16
4;          1948223.10346600; 2934783.00035600; 0.50639000; 0.02; 0.19
5;          1950714.27729900; 2665396.21953300; 0.36636900; 0.06; 0.25
6;          1951139.66498400; 2625669.76076600; 0.34571100; 0.05; 0.30
7;          1951361.24068100; 2604977.06538300; 0.33495400; 0.04; 0.34
8;          1951505.05080900; 2591546.80564200; 0.32797300; 0.04; 0.39
9;          1951646.49244200; 2578337.73683300; 0.32110900; 0.04; 0.43
10;         1951649.92372300; 2578017.29355800; 0.32094200; 0.04; 0.47
11;         1951712.08152100; 2572212.44945100; 0.31792600; 0.06; 0.53
12;         1951902.50240500; 2554429.26557400; 0.30868700; 0.06; 0.59
13;         1952147.88450900; 2531513.31773400; 0.29678400; 0.05; 0.64
14;         1952240.22542900; 2522889.70711900; 0.29230500; 0.05; 0.70
15;         1952279.27522300; 2519242.89250100; 0.29041100; 0.05; 0.75
16;         1952358.13394300; 2511878.36873700; 0.28658700; 0.06; 0.81
17;         1952397.66958200; 2508186.18162000; 0.28467000; 0.05; 0.87
18;         1952417.98397700; 2506289.04400100; 0.28368500; 0.06; 0.93
19;         1952427.70087800; 2505381.59401800; 0.28321400; 0.07; 1.00
20;         1952441.39563500; 2504102.65669900; 0.28254900; 0.07; 1.07
21;         1952443.33901500; 2503921.16670200; 0.28245500; 0.07; 1.13
22;         1952444.70545400; 2503793.55654900; 0.28238900; 0.07; 1.21
23;         1952444.76618500; 2503787.88498600; 0.28238600; 0.06; 1.27
24;         1952445.70750900; 2503699.97576900; 0.28234000; 0.07; 1.35
25;         1952445.76824000; 2503694.30420700; 0.28233700; 0.06; 1.41
26;         1952445.92006600; 1952445.92006600; -0.00000000; 0.08; 1.49
```

Figura 52 – Convergência do problema “tutorial-cond-constr” resolvido pela DDiP com cortes de Lagrange

Com relação a operação, foi igual a operação obtida pelo método MILP-global equivalente conforme mostrado na seção B.2.1.

C.1.3 Cortes de Lagrange

O processo de convergência para execução utilizando os cortes de Lagrange com até 28 variáveis binárias na representação do volume está exposto na Figura 56.

```
&*****
& Nome do Modelo: Libs Customizado Versão: 0.0.1
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter .....;ZInf .....;ZSup .....;GAP .....;TempoIter .....;TempoAcum .....;
&- .....;$ .....;$ .....;- .....;s .....;s .....;
&IIIIIIIIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
.....0;.....738.60848300;3549384104.46791792;.....4805500.40784700;.....2.02;.....2.02
.....1;3366024748.65212202;3368079122.59030390;.....0.00061000;.....2.59;.....4.61
.....2;3367242325.08868980;3367694531.77953815;.....0.00013400;.....2.23;.....6.84
.....3;3367472096.92611504;3367668401.79615021;.....0.00005800;.....3.13;.....9.97
.....4;3367630577.27827978;3367633185.44441938;.....0.00000100;.....3.70;.....13.67
.....5;3367632222.85431767;3367632224.44902515;.....0.00000000;.....3.40;.....17.07
.....6;3367632223.12103605;3367632223.66353130;.....0.00000000;.....6.60;.....23.68
.....7;3367632221.98130608;3367632222.60409594;.....0.00000000;.....6.71;.....30.39
```

Figura 56 – Convergência do problema “reduced-system-no-constr” resolvido com cortes de Lagrange onde o volume foi representado utilizando até 28 variáveis binárias

C.1.4 Cortes de Benders Reforçado

As Figuras 57 e 58 mostram o processo de convergência para o caso “reduced-system-no-constr” utilizando os cortes de Benders Reforçados na DDP (com representação “continuous-storage”) e na DDiP (com representação “binary-storage”) respectivamente.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          738.60847800; 3549384096.53024340; 4805500.42657200; 0.17; 0.17
1; 3366024753.80215311; 3368079118.66235781; 0.00061000; 0.17; 0.34
2; 3367242326.04762506; 3367694532.16429329; 0.00013400; 0.13; 0.47
3; 3367472097.52492905; 3367668407.67901516; 0.00005800; 0.14; 0.61
4; 3367472097.52493000; 3367660437.28854227; 0.00005600; 0.14; 0.75
5; 3367630578.72027588; 3367633180.78011990; 0.00000100; 0.16; 0.91
6; 3367632219.23738194; 3367632219.23738813; 0.00000000; 0.18; 1.08
```

Figura 57 – Convergência do problema “reduced-system-no-constr” resolvido pela DDP com cortes de Benders Reforçados com representação “continuous-storage”

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter    ;TempoAcum    ;
&-         ;$         ;$         ;-         ;s           ;s           ;
&IIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;          738.60848300; 3549384104.46791792; 4805500.40784700; 3.70; 3.70
1; 3366024771.24167109; 3368079126.31905079; 0.00061000; 7.55; 11.25
2; 3367242335.99331522; 3367694536.99083281; 0.00013400; 7.47; 18.72
3; 3367472113.00405502; 3367668399.44557381; 0.00005800; 8.19; 26.92
4; 3367630590.14827824; 3367633184.32550001; 0.00000100; 7.20; 34.11
5; 3367632233.47331667; 3367632222.32540989; -0.00000000; 10.34; 44.45
6; 3367632233.47331667; 3367632222.23133898; -0.00000000; 9.20; 53.65
```

Figura 58 – Convergência do problema “reduced-system-no-constr” resolvido pela DDiP com cortes de Benders Reforçados com representação “binary-storage”

C.2 Resultados do problema “reduced-system-cond-constr”

C.2.1 Cortes de Benders

As Figuras 59 e 60 mostram o processo de convergência para o caso “reduced-system-cond-constr” utilizando os cortes de Benders na DDP (com representação “continuous-storage”) e na DDiP (com representação “binary-storage”) respectivamente.

```

&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter      ;TempoAcum      ;
&-         ;$         ;$         ;-         ;s         ;s         ;
&IIIIIIIIIIIIIIIIIIII;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;
0;          600918.64909400; 3550320168.93402815;          5907.15441400;          0.24;          0.24
1; 3366024753.80218220; 3371833109.72514677;          0.00172600;          0.38;          0.61
2; 3367237702.06526518; 3371516269.47965908;          0.00127100;          0.35;          0.97
3; 3367470970.72112703; 3371392330.72817898;          0.00116400;          0.33;          1.30
4; 3367627848.05328178; 3371415773.33743382;          0.00112500;          0.32;          1.61
5; 3367628219.61763000; 3371415773.33743477;          0.00112500;          0.24;          1.85

```

Figura 59 – Convergência do problema “reduced-system-cond-constr” resolvido pela DDP com cortes de Benders com representação “continuous-storage”

```

&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter      ;TempoAcum      ;
&-         ;$         ;$         ;-         ;s         ;s         ;
&IIIIIIIIIIIIIIIIIIII;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFFFFFFFFFF;
0;          600941.18631500; 3550320268.29124594;          5906.93300400;          2.98;          2.98
1; 3366024755.67368078; 3371833117.44946003;          0.00172600;          2.15;          5.13
2; 3367237703.55961323; 3371516273.88462782;          0.00127100;          2.22;          7.35
3; 3367470972.70765781; 3371392312.48101807;          0.00116400;          2.39;          9.74
4; 3367627850.87939501; 3371415777.38327885;          0.00112500;          1.66;          11.40
5; 3367628220.92064714; 3371415775.33504200;          0.00112500;          1.86;          13.26
6; 3367628221.41403818; 3371415782.60816002;          0.00112500;          2.55;          15.81

```

Figura 60 – Convergência do problema “reduced-system-cond-constr” resolvido pela DDiP com cortes de Benders com representação “binary-storage”

C.2.2 Cortes de Lagrange

O processo de convergência para execução utilizando os cortes de Lagrange com até 28 variáveis binárias na representação do volume está exposto na Figura 61.

```
&*****
& Nome do Modelo: Libs Customizado Versão: 0.0.1
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****;*****;*****;*****;*****;*****;*****;
&Iter .....;ZInf .....;ZSup .....;GAP .....;TempoIter .....;TempoAcum .....;
&- .....;$ .....;$ .....;$ .....;$ .....;$ .....;$ .....;$ .....;$ .....;
&IIIIIIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
.....0;.....600941.18631500; 3550320268.29131079;.....5906.93300400;.....1.94;.....1.94
.....1; 2971957426.33068180; 3378929781.05970621;.....0.13693700;.....3.68;.....5.62
.....2; 3106945439.23869038; 3377567441.84605122;.....0.08710200;.....210.63;.....216.25
.....3; 3366231934.93644524; 3371833130.95484114;.....0.00166400;.....3.80;.....220.05
.....4; 3366231934.63328123; 3371833200.48612690;.....0.00166400;.....18.74;.....238.79
...
.....39; 3366231948.54302788; 3371916909.02845907;.....0.00168900;.....268.87;.....3456.07
.....40; 3366231948.18222904; 3372048165.46979523;.....0.00172800;.....121.96;.....3578.03
.....41; 3366231948.29134083; 3371815714.16363001;.....0.00165900;.....135.77;.....3713.81
...
.....58; 3366231952.56506729; 3371987836.58529806;.....0.00171000;.....228.30;.....6838.78
.....59; 3366231951.03895092; 3372101617.48398018;.....0.00174400;.....396.62;.....7235.40
.....60; 3366231952.30790186; 3372068491.20375824;.....0.00173400;.....241.24;.....7476.65
...
.....95; 3366231954.85112429; 3371953228.64077902;.....0.00170000;.....702.77;.....26296.38
.....96; 3366231954.79881287; 3371989711.21537924;.....0.00171000;.....769.11;.....27065.49
.....97; 3366231955.69485188; 3371963611.78955078;.....0.00170300;.....759.50;.....27824.99
```

Figura 61 – Convergência do problema “reduced-system-cond-constr” resolvido com cortes de Lagrange onde o volume foi representado utilizando até 28 variáveis binárias

C.2.3 Cortes de Benders Reforçado

As Figuras 62 e 63 mostram o processo de convergência para o caso “reduced-system-cond-constr” utilizando os cortes de Benders Reforçados na DDP (com representação “continuous-storage”) e na DDiP (com representação “binary-storage”) respectivamente.

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter      ;TempoAcum      ;
&-         ;$         ;$         ;-         ;s         ;s         ;
&IIIIIIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;      600918.64909400; 3550320168.93402815;      5907.15441400;      0.17;      0.17
1; 3366024753.80217838; 3371833109.72514677;      0.00172600;      0.59;      0.76
2; 3367238227.80955505; 3371516269.47966003;      0.00127000;      0.75;      1.51
3; 3367470977.02505398; 3371392122.24630022;      0.00116400;      0.74;      2.25
4; 3367628019.44534254; 3371415773.33743382;      0.00112500;      1.00;      3.26
5; 3367628368.49712706; 3371415773.33743382;      0.00112500;      0.60;      3.85
```

Figura 62 – Convergência do problema “reduced-system-cond-constr” resolvido pela DDP com cortes de Benders Reforçados com representação “continuous-storage”

```
&*****
& Descrição colunas:
&*****
& Iter: Número da iteração
& ZInf: Limite inferior no processo iterativo
& ZSup: Limite superior no processo iterativo
& GAP: Gap de convergência: (zsup - zinf) / zinf
& TempoIter: Tempo da iteração
& TempoAcum: Tempo acumulado até a iteração
&*****
&Iter      ;ZInf      ;ZSup      ;GAP      ;TempoIter      ;TempoAcum      ;
&-         ;$         ;$         ;-         ;s         ;s         ;
&IIIIIIIIIIIIII;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;FFFFFFFFFFFFFFFF;
0;      600941.18631500; 3550320268.29131079;      5906.93300400;      3.67;      3.67
1; 3366024766.24590921; 3371833116.56705999;      0.00172600;      6.45;      10.12
2; 3367238236.98535824; 3371516297.18408823;      0.00127000;      6.80;      16.91
3; 3367470988.55532885; 3371392129.76153517;      0.00116400;      8.15;      25.07
4; 3367628026.90600300; 3371415777.62676620;      0.00112500;      7.51;      32.57
5; 3367628380.41071987; 3371415779.43598700;      0.00112500;      8.07;      40.64
6; 3367628381.61497402; 3371415794.71581268;      0.00112500;      7.93;      48.57
```

Figura 63 – Convergência do problema “reduced-system-cond-constr” resolvido pela DDiP com cortes de Benders Reforçados com representação “binary-storage”

APÊNDICE D – EXEMPLO UNIDIMENSIONAL FUNÇÃO DE RECURSO

O problema exemplo retirado de [63], analisado na seção 6.1, foi formulado e resolvido com o seguinte código *python*.

```
import numpy as np
from scipy.optimize import milp, LinearConstraint, Bounds
# escolha do mu
mu = 4
#custo
c = np.array([1,1,-mu])
#integralidade
integrality = ([1,0,0])
# restricoes
A0 = [2, 1, -3]
lb0 = 0
ub0 = 100000000000
# caixa
# y1
A1 = [1, 0, 0]
lb1=0
ub1=2
# y2
A2 = [0, 1, 0]
lb2=0
ub2=3
# z
A3 = [0, 0, 1]
lb3=0
ub3=1
#
A = np.array([A0,A1,A2,A3])
lb = np.array([lb0,lb1,lb2,lb3])
ub = np.array([ub0,ub1,ub2,ub3])
res=milp(c,integrality=integrality,constraints=LinearConstraint(A,lb,ub))
print ('Valor otimo: ',res.fun)
print('y1 = ',res.x[0])
print('y2 = ',res.x[1])
print('z = ',res.x[2])
```

APÊNDICE E – SENSIBILIDADE PARA DIFERENTES REGRAS DE ESCOLHA DAS SOLUÇÕES INICIAIS PARA A DECOMPOSIÇÃO EM DUAS ETAPAS

Os valores candidatos de armazenamento utilizados na segunda etapa da abordagem de decomposição em duas etapas são obtidos a partir de uma combinação das soluções das aproximações “no-constr”, “with-constr” e “cond-constr” geradas na primeira etapa, conforme descrito nas Equações (10.2)–(10.10).

Nesta seção, avalia-se a sensibilidade do método considerando combinações alternativas para os volumes $V_{1i}^{t,s}$ e $V_{2i}^{t,s}$, conforme descrito a seguir:

- **Alternativa 1:** considera apenas as soluções das variantes “no-constr” e “cond-constr”:

$$V_{1i}^{t,s} = \min\{V_{no\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (E.1)$$

$$V_{2i}^{t,s} = \max\{V_{no\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (E.2)$$

- **Alternativa 2:** considera apenas as soluções das variantes “with-constr” e “cond-constr”:

$$V_{1i}^{t,s} = \min\{V_{with\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (E.3)$$

$$V_{2i}^{t,s} = \max\{V_{with\ i}^{t,s}, V_{cond\ i}^{t,s}\} \quad (E.4)$$

- **Alternativa 3:** considera apenas as soluções das variantes “no-constr” e “with-constr”:

$$V_{1i}^{t,s} = \min\{V_{no\ i}^{t,s}, V_{with\ i}^{t,s}\} \quad (E.5)$$

$$V_{2i}^{t,s} = \max\{V_{no\ i}^{t,s}, V_{with\ i}^{t,s}\} \quad (E.6)$$

A Figura 64 compara os resultados obtidos por cada alternativa com aqueles produzidos pela abordagem proposta (em amarelo), considerando as mesmas 21 instâncias do problema de grande porte analisadas na Seção 10.3.2. Essa análise complementar confirma a importância de priorizar a solução da variante “cond-constr”, conforme estabelecido nas Equações (10.2)–(10.3). Observa-se que desconsiderar essa solução (Alternativa 3) resulta nos piores desempenhos dentre as alternativas avaliadas. Além disso, verifica-se que, entre as quatro estratégias investigadas, a abordagem proposta apresenta os menores *gaps* relativos de otimalidade.

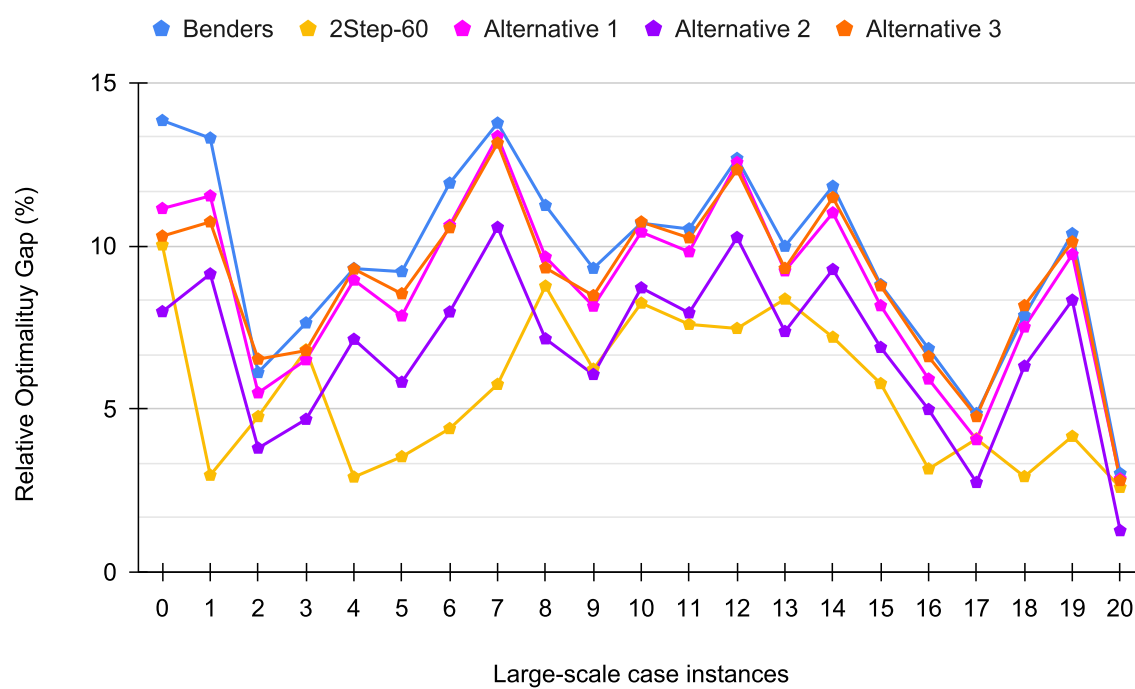


Figura 64 – Gap relativo de otimalidade para diferentes escolhas dos valores iniciais utilizados na segunda etapa