

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

Lucas Reis Conceição

Paralelização Assíncrona da Programação Dinâmica Dual com Agrupamento Ótimo de Nós Para o Planejamento da Operação de Sistemas Hidrotérmicos

Juiz de Fora

2019

Lucas Reis Conceição

Paralelização Assíncrona da Programação Dinâmica Dual com Agrupamento Ótimo de Nós Para o Planejamento da Operação de Sistemas Hidrotérmicos

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

Orientador: André Luís Marques Marcato, D.Sc

Coorientador: André Luiz Diniz Souto Lima, D.Sc

Juiz de Fora

2019

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

Reis, Lucas.

Paralelização Assíncrona da Programação Dinâmica Dual com Agrupamento Ótimo de Nós Para o Planejamento da Operação de Sistemas Hidrotérmicos / Lucas Reis Conceição. – 2019.

81 f. : il.

Orientador: André Luís Marques Marcato, D.Sc

Coorientador: André Luiz Diniz Souto Lima, D.Sc

Dissertação (mestrado acadêmico) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica , 2019.

1. Planejamento Sistemas Hidrotérmicos. 2. Programação Dinâmica Dual. 3. Paralelização assíncrona. 4. Algoritmo Genético. I. Marcato, André Luís Marques, orient. II. Diniz, André Luiz Souto Lima, coorient. III. Título.

Lucas Reis Conceição

Paralelização Assíncrona da Programação Dinâmica Dual com Agrupamento Ótimo de Nós Para o Planejamento da Operação de Sistemas Hidrotérmicos

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

Aprovada em: 24 de outubro de 2019

BANCA EXAMINADORA

Professor André Luís Marques Marcato, D.Sc -
Orientador
Universidade Federal de Juiz de Fora

Professor André Luiz Diniz Souto Lima, D.Sc -
Coorientador
Centro de Pesquisas de Energia Elétrica
Universidade Estadual do Rio de Janeiro

Professora Carmen Lucia Tancredo Borges, Dra.Sc
Universidade Federal do Rio de Janeiro

Professor João Alberto Passos Filho, D.Sc
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Agradeço a toda minha família, principalmente aos meus pais Júlio e Rogéria e meus irmãos Rafael e Felipe, por todo o suporte fundamental para minhas conquistas, com seus ensinamentos e companheirismo são responsáveis pela formação do meu caráter. À minha namorada Roberta, por sempre estar ao meu lado me apoiando com toda sua cumplicidade e companheirismo.

Ao professor e orientador André Marcato, por me apresentar a área de planejamento energético e me dar todo o apoio, incentivo e atenção no desenvolvimento da minha dissertação, além de ser responsável, em grande parte, pelo meu crescimento acadêmico nesta universidade, e pela confiança no meu trabalho para realizar os projetos de P&D. Ao professor e coorientador André Luiz Diniz, pela confiança, apoio e, principalmente, pelas revisões e sugestões nos trabalhos desenvolvidos durante meu mestrado. Ao professor João Alberto e à professora Carmen Lucia, membros da banca, pela disponibilidade em fazer parte da minha defesa, enriquecendo o trabalho com os seus conhecimentos na área.

Ao PPEE-UFJF e à UFJF pelo suporte financeiro e por prover a estrutura e as ferramentas necessárias para o desenvolvimento deste trabalho. Ao suporte financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), do Centro de Pesquisas de Energia Elétrica (CEPEL), da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), do Instituto Nacional de Energia Elétrica (INERGE) e da Agência Nacional de Energia Elétrica (ANEEL).

Por fim, agradeço à todos que, de alguma forma, contribuíram para a realização deste trabalho.

RESUMO

O problema de planejamento da operação de sistemas de energia hidrotérmicos de grande porte é complexo devido à necessidade de se considerar as incertezas inerentes às afluências às usinas hidroelétricas. Para a solução deste problema várias metodologias podem ser aplicadas, entre elas a Programação Dinâmica Dual (PDD), que é uma estratégia de decomposição capaz de resolver problemas de otimização estocástica multi estágio. Para problemas reais como o do Sistema Interligado Nacional (SIN) do Brasil, esta metodologia pode demandar um esforço computacional elevado devido ao número de reservatórios e horizonte de estudo. Este trabalho apresenta uma nova metodologia que visa a redução de tempo computacional da PDD. A metodologia implementada propõe um aprimoramento no algoritmo de processamento paralelo assíncrono para a PDD através de um agrupamento ótimo dos nós da árvore de cenários em subproblemas, por meio de algoritmo genético. Estas estratégias fazem melhor uso dos recursos disponíveis ao contornar algumas restrições de sincronismo da PDD que podem ser muito prejudiciais ao paralelismo. A eficiência da metodologia proposta é mostrada para um conjunto de usinas hidroelétricas do SIN.

Palavras-chave: Planejamento Sistemas Hidrotérmicos. Programação Dinâmica Dual. Paralelização assíncrona. Algoritmo Genético.

ABSTRACT

The hydrothermal coordination problem of real systems is complex due to the need to consider the uncertainties inherent in the inflow to hydroelectric plants. To solve this problem, several methodologies can be applied, including the Dual Dynamic Programming (DDP), which is a decomposition strategy capable of solving multi-stage stochastic optimization problems. For real problems such as Brazil's National Interconnected System (SIN), this methodology may require a high computational effort due to the number of reservoirs and study horizon. This masters dissertation presents a new methodology that aims to reduce the computational time of DDP. The implemented methodology proposes an enhancement in asynchronous parallel processing algorithm for DDP, including an optimal grouping of scenario tree nodes into subproblems by genetic algorithm. These strategies make the best use of available resources bypassing some DDP synchronization restrictions that can be very detrimental to parallelism. The efficiency of the proposed methodology is shown for a set of SIN hydroelectric plants.

Key-words: hydrothermal Coordination Problem. Dual Dynamic Programming. Asynchronous Parallelization. Genetic Algorithm.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação dos subsistemas do SIN	17
Figura 2 – Complementariedade anual entre as diversas fontes de geração	18
Figura 3 – Dilema do Operador	19
Figura 4 – Função de custo imediato, custo futuro e custo total	19
Figura 5 – Modelagem do Sistema Hidrotérmico no Planejamento da Operação . .	20
Figura 6 – Reservatórios Equivalentes de Energia	21
Figura 7 – Tipos de agrupamento de nós (1)	30
Figura 8 – Tipos de agrupamento de nós (2)	30
Figura 9 – Pseudocódigo algoritmo evolucionista	31
Figura 10 – Exemplo árvore de cenários	39
Figura 11 – Fluxograma Decomposição de Benders	43
Figura 12 – Fluxograma PDD tradicional	45
Figura 13 – Fluxograma PDD recursiva	47
Figura 14 – Processo de solução PDD recursiva (1)	49
Figura 15 – Processo de solução PDD recursiva (2)	49
Figura 16 – Exemplo de agrupação de nós (1)	52
Figura 17 – Exemplo de agrupação de nós (2)	52
Figura 18 – Funcionamento PDD paralelizada de forma assíncrona (1)	53
Figura 19 – Funcionamento PDD paralelizada de forma assíncrona (2)	54
Figura 20 – Funcionamento PDD paralelizada de forma assíncrona (3)	55
Figura 21 – Fluxograma processador mestre	56
Figura 22 – Fluxograma processador escravo	57
Figura 23 – Distribuição de subproblemas para um mesmo processador	57
Figura 24 – Fluxograma algoritmo genético	58
Figura 25 – Hardware CLUSTER-PPEE	61
Figura 26 – Sistema Tutorial	62
Figura 27 – Afluências sorteadas para uma árvore de cenários de 4 estágios e 2 aberturas	62
Figura 28 – Pesos para uma árvore de cenários de 3 estágios e 2 aberturas com nós equiprováveis	63
Figura 29 – Convergência PDD recursiva serial sistema tutorial	65
Figura 30 – Convergência PDD recursiva paralelizada de forma assíncrona sistema tutorial	66
Figura 31 – Comparação dos tempos computacionais para sistema tutorial(1) . . .	66
Figura 32 – Convergência das metodologias para sistema tutorial	67
Figura 33 – Comparação dos tempos computacionais para sistema tutorial(2) . . .	67
Figura 34 – Comparação dos tempos computacionais para diferentes agrupamentos	68
Figura 35 – Avaliação Algoritmo Genético	69

Figura 36 – Usinas Hidrelétricas Subsistema Norte	70
Figura 37 – Usinas Térmicas Subsistema Norte	70
Figura 38 – Convergência PDD recursiva serial subsistema Norte	71
Figura 39 – Convergência PDD recursiva paralelizada de forma assíncrona susistema Norte	71
Figura 40 – Comparação dos tempos computacionais para subsistema Norte(1) . . .	72
Figura 41 – Geração Hidrelétrica e Termelétrica Total do Subsistema	72
Figura 42 – Armazenamento de água no primeiro estágio	73
Figura 43 – Convergência das metodologias para subsistema Norte	74
Figura 44 – Comparação dos tempos computacionais para subsistema Norte(2) . . .	74
Figura 45 – Speed up Test Subsistema Norte(1)	75
Figura 46 – Speed up Test Subsistema Norte(2)	75

LISTA DE ABREVIATURAS E SIGLAS

AE	Algoritmo Evolucionista
AG	Algoritmo Genético
ANEEL	Agência Nacional de Energia Elétrica
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CEPEL	Centro de Pesquisas de Energia Elétrica
CLUSTER	Plataforma de Computação Paralela
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CPU	Unidade Central de Processamento
FAPEMIG	Fundação de Amparo à Pesquisa do Estado de Minas Gerais
FCF	Função de Custo Futuro
FPH	Função de Produção Hidráulica
FPHA	Função de Produção Hidráulica Aproximada
INERGE	Instituto Nacional de Energia Elétrica
MPI	<i>Message Passing Interface</i>
ONS	Operador Nacional do Sistema Elétrico
PDD	Programação Dinâmica Dual
PDDD	Programação Dinâmica Dual Determinística
PL	Problema Linear
PLPE	Problemas Lineares de Programação Estocástica
PMO	Programa Mensal de Operação Energética
PPE	Problema de Programação Estocástica
PPEE	Programa de Pós Graduação de Engenharia Elétrica da Universidade Federal de Juiz de Fora
SBAI	Simpósio Brasileiro de Automação Inteligente
SIN	Sistema Interligado Nacional

UERJ	Universidade Estadual do Rio de Janeiro
UFJF	Universidade Federal de Juiz de Fora
UFOP	Universidade Federal de Ouro Preto
UFRJ	Universidade Federal do Rio de Janeiro

LISTA DE SÍMBOLOS

NT	Número de térmicas do sistema em estudo
NH	Número de hidrelétricas do sistema em estudo
t	Período de tempo do nó
s	Cenário de entrada do nó
ct_i	Custo de geração de energia da usina térmica i
$gt_i^{t,s}$	Geração Termelétrica da usina i no período de tempo t para o cenário de entrada s
c_{def}	Custo do déficit de energia
$def^{t,s}$	Valor de déficit de energia no período t para o cenário de entrada s
K_{pen}	Constante de penalização do vertimento
$vv_j^{t,s}$	Vazão Vertida da usina j no período de tempo t para o cenário de entrada s
$\alpha^{t+1,s}$	Estimativa do custo futuro para o período de tempo $t+1$ para o cenário de entrada s
NCB	Número de cortes de Benders
$\lambda_{j,k}^{t+1,s}$	Coefficiente do corte k da FCF que multiplica o volume armazenado da hidrelétrica j , dado pelo multiplicador de Lagrange da equação de balanço hídrico da usina j no período de tempo $t + 1$ para o cenário s
$va_j^{t+1,s}$	Volume armazenado da usina j ao final do período de tempo t para o cenário de entrada s
$K_{indep_k}^{t+1,s}$	Coefficiente independente do corte k da FCF no estágio $t + 1$ para o cenário de entrada s
DEM_t^s	Valor da demanda de energia do sistema em estudo no período t para o cenário de entrada s
J	Conjunto de todas as usinas hidrelétricas do sistema
K_{conv}	Constante de conversão de unidades, de m^3/s para hm^3 , de valor igual a 2,592

$va_j^{t,s}$	Volume armazenado da usina j no início do período de tempo t para o cenário de entrada s
$Vinc_j^{t,s}$	Vazão incremental afluyente da usina j no período de tempo t para o cenário de entrada s
M	Conjunto de todas as usinas hidrelétricas à montante da usina j
$vt_m^{t,s}$	Vazão turbinada pela usina m a montante da usina j no período de tempo t para o cenário de entrada entrada s
$vv_m^{t,s}$	Vazão vertida pela usina m a montante da usina j no período de tempo t para o cenário de entrada entrada s
$vt_j^{t,s}$	Vazão turbinada pela usina j no período de tempo t para o cenário de entrada entrada s
$vv_j^{t,s}$	Vazão vertida da usina j no período de tempo t para o cenário de entrada entrada s
GH_n	Geração hidráulica do ponto n
ρ_{esp}	Produtibilidade específica da usina
Q_y	Vazão turbinada do ponto n
V_x	Volume armazenado do ponto n
S_z	Vazão vertida do ponto n
h_{mon}	Altura de queda líquida montante
h_{jus}	Altura líquida de jusante
h_{perdas}	Altura líquida de perdas
P_j	Conjunto de todos hiperplanos obtidos para a usina j
$\gamma_{0_j}^p$	Parâmetro constante do hiperplano p para a usina j
$\gamma_{va_j}^p$	Parâmetro do hiperplano p associado ao volume armazenado da usina j
$\gamma_{vt_j}^p$	Parâmetro do hiperplano p associado à vazão turbinada da usina j
$\gamma_{vv_j}^p$	Parâmetro do hiperplano p associado à vazão vertida da usina j
VA_j^{min}	Volume de armazenamento mínimo da usina j
VA_j^{max}	Volume de armazenamento máximo da usina j

VT_j^{max}	Vazão máxima que pode ser turbinada pela usina j
VV_j^{max}	Vazão máxima que pode ser vertida pela usina j
GT_i^{max}	Capacidade máxima de geração da termelétrica j
$SubP_t^s$	Subproblema para o cenário s do estágio t
D_{t-1}^u	Conjunto de cenários descendentes do subproblema $SubP_t^u$
P_t^s	Probabilidade condicional do cenário s do estágio t dado que o cenário u tenha ocorrido e $s \in D_{t-1}^u$
x_1, x_2	Variáveis que podem assumir valores distintos nos subproblemas mestre e escravo
y_1	Variáveis de decisão, calculadas durante o subproblema mestre
y_1^*	Valor encontrado para y_1 no subproblema mestre e mantido como parâmetro no subproblema escravo
$f_1(x_1, y_1)$	Parcela da função objetivo do problema original que irá compor a função objetivo do subproblema mestre
α	Parcela que compõe a função objetivo do subproblema mestre e representa a estimativa do custo do subproblema escravo
$f_2(x_2)$	Parcela da função objetivo do problema original que será a função objetivo do subproblema escravo
$g_1(x_1, y_1) \geq B_1$	Restrições do subproblema mestre
$g_2(x_1, y_1^*) \geq B_2$	Restrições do subproblema escravo
λ_2	Valor do coeficiente de Lagrange associado à restrição do subproblema escravo
$Zinf$	Limite inferior do problema
$Zsup$	Limite superior do problema
ϵ	Tolerância para convergência pré-determinada
E_{y_1}	Derivadas parciais das restrições $g_2(x_1, y_1^*)$ em relação a y_1^*
w_2	Valor da função objetivo obtido do subproblema escravo
V_{y_1}	Conjunto das variáveis de decisão y_1 presentes no subproblema mestre

$Rest_{y_1^*}$	Conjunto de restrições presentes no subproblema escravo que incluem o parâmetro y_1^*
λ_2^r	Valor do coeficiente de Lagrange obtido do subproblema escravo associado à restrição r
$E_{y_1}^r$	Derivada parcial da restrição r em relação a y_1
f_1^*	Valor encontrado para a função objetivo do subproblema do primeiro estágio
α_1^*	Valor estimado do custo futuro ao final da resolução do subproblema do primeiro estágio
T	Número de estágios do problema
NA	Número de aberturas da árvore de cenários do problema
$Zsup_{med}$	Limite superior médio do problema
f_t^a	Valor encontrado para a função objetivo do subproblema do estágio t e abertura a
α_t^a	Valor estimado do custo futuro ao final da resolução do subproblema do estágio t e abertura a
c_{gT1}	Custo da geração da usina térmica 1 do sistema tutorial
c_{gT2}	Custo da geração da usina térmica 2 do sistema tutorial
c_{defST}	Custo do déficit do sistema tutorial
$gT1_a^e$	Variáveis de decisão da geração da usina térmica 1 do sistema tutorial para a abertura a do estágio e
$gT2_a^e$	Variáveis de decisão da geração da usina térmica 2 do sistema tutorial para a abertura a do estágio e
gH_a^e	Variáveis de decisão da geração da usina hidrelétrica do sistema tutorial para a abertura a do estágio e
def_{STa}^e	Variáveis de decisão do déficit do sistema tutorial para a abertura a do estágio e
DEM_{STa}^e	Valor da demanda do sistema tutorial para a abertura a do estágio e
vaH_a^{e+1}	Variáveis de decisão do volume armazenado da hidrelétrica do sistema tutorial para a abertura a ao final do estágio e

vaH_a^e	Variáveis de decisão do volume armazenado da hidrelétrica do sistema tutorial para a abertura a do estágio e
K_{conv}	Constante de conversão de unidades, de m^3/s para hm^3/s , de valor igual a 2,592
$VAF L_a^e$	Valor da vazão afluyente à usina hidrelétrica do sistema tutorial para a abertura a do estágio e
vtH_a^e	Variáveis de decisão do volume turbinado da hidrelétrica do sistema tutorial para a abertura a do estágio e
	Variáveis de decisão do volume vertido da hidrelétrica do sistema tutorial para a abertura a do estágio e
$GT1_{max}$	Capacidade máxima de geração da usina térmica 1 do sistema tutorial
$GT2_{max}$	Capacidade máxima de geração da usina térmica 2 do sistema tutorial
VAH_{min}	Volume de armazenamento mínimo da usina hidrelétrica do sistema tutorial
VAH_{max}	Volume de armazenamento máximo da usina hidrelétrica do sistema tutorial
VTH_{max}	Vazão máxima que pode ser turbinada pela usina hidrelétrica do sistema tutorial
VVH_{max}	Vazão máxima que pode ser vertida pela usina hidrelétrica do sistema tutorial

SUMÁRIO

1	INTRODUÇÃO E MOTIVAÇÃO	17
1.1	CONSIDERAÇÕES INICIAIS	17
1.2	MOTIVAÇÃO	22
1.3	OBJETIVOS	23
1.4	PUBLICAÇÕES	24
1.5	ESTRUTURA DA DISSERTAÇÃO	24
2	REVISÃO BIBLIOGRÁFICA	26
2.1	PROGRAMAÇÃO DINÂMICA DUAL	26
2.1.1	Considerações iniciais	26
2.1.2	Varredura da árvore de cenários	26
2.1.3	Paralelização	27
2.1.4	Agrupamento de nós	29
2.2	ALGORITMOS GENÉTICOS	31
3	METODOLOGIA PROPOSTA	33
3.1	FORMULAÇÃO PLANEJAMENTO ENERGÉTICO	33
3.1.1	Função objetivo	33
3.1.2	Função de custo futuro	34
3.1.3	Atendimento à demanda	35
3.1.4	Balanco hídrico	35
3.1.5	Função de produção hidráulica	36
3.1.6	Limites operativos das variáveis	38
3.2	PROGRAMAÇÃO DINÂMICA DUAL	39
3.2.1	Decomposição de Benders	39
3.2.2	Programação dinâmica dual tradicional	44
3.2.3	Programação dinâmica dual recursiva	46
3.2.4	Programação dinâmica dual recursiva paralelizada de forma assíncrona	51
3.2.5	Agrupamento ótimo de nós	58
4	RESULTADOS	60
4.1	SISTEMA TUTORIAL	61
4.1.1	Resultados sistema tutorial	65
4.2	SUBSISTEMA NORTE	69
4.2.1	Resultados subsistema norte	71

5	CONCLUSÃO E TRABALHOS FUTUROS	76
5.1	Conclusões	76
5.2	Trabalhos futuros	76
	REFERÊNCIAS	78

1 INTRODUÇÃO E MOTIVAÇÃO

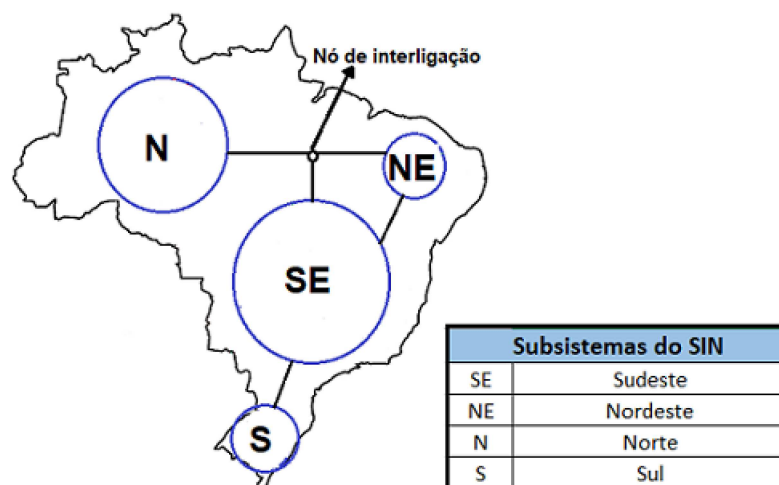
1.1 CONSIDERAÇÕES INICIAIS

O território brasileiro possui características particulares que fazem com que seu sistema elétrico seja diferenciado em relação aos demais países do mundo. De acordo com [1] o sistema de produção e transmissão de energia elétrica do Brasil é um sistema hidro-termo-eólico de grande porte, com predominância de usinas hidrelétricas e com múltiplos proprietários. O Sistema Interligado Nacional (SIN) é constituído por quatro subsistemas: Sul, Sudeste/Centro-Oeste, Nordeste e a maior parte da região Norte. A Figura 1 ilustra a divisão do SIN em subsistemas.

Ainda segundo [1] a capacidade instalada de geração do SIN é composta, principalmente, por usinas hidrelétricas distribuídas em dezesseis bacias hidrográficas nas diferentes regiões do país. As usinas térmicas, em geral localizadas nas proximidades dos principais centros de carga, desempenham papel estratégico relevante, pois contribuem para a segurança do SIN. Essas usinas são despachadas em função das condições hidrológicas vigentes, permitindo a gestão dos estoques de água armazenada nos reservatórios das usinas hidrelétricas, para assegurar o atendimento futuro. A interconexão entre os subsistemas permite obter vantagens da diversidade hidrológica entre as diversas bacias hidrográficas, uma vez que estas apresentam regimes hidrológicos complementares, como apresentado na Figura 2.

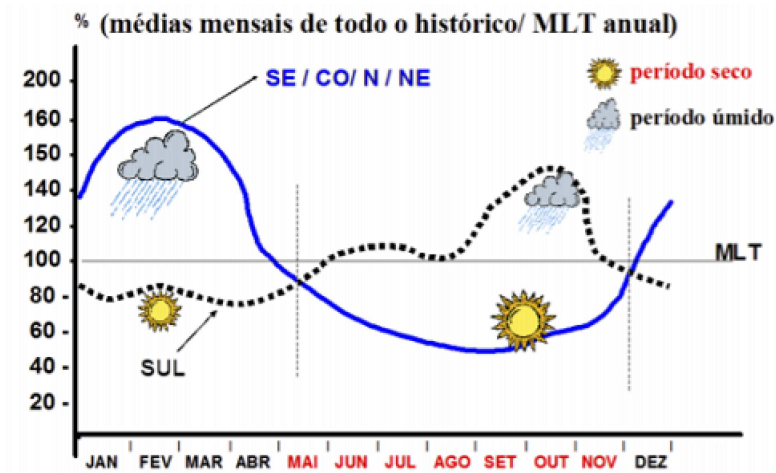
O SIN por ser um sistema predominantemente hídrico com complementariedade térmica possui características particulares, destacam-se a estocasticidade, acoplamento espacial da operação e o acoplamento temporal da operação [3][6]. A estocasticidade

Figura 1 – Representação dos subsistemas do SIN



Fonte: Elaborada pelo autor

Figura 2 – Complementariedade anual entre as diversas fontes de geração



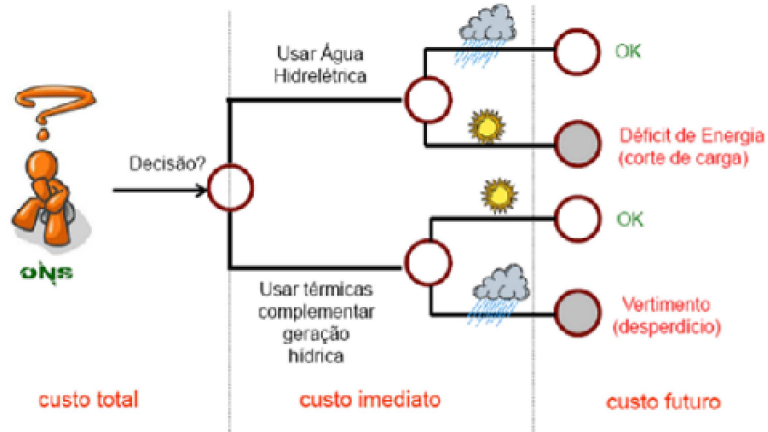
Fonte: [2]

está relacionada com a incerteza em relação às afluências futuras, que se tornam mais significativas quanto maior for o horizonte do planejamento. O acoplamento espacial da operação existe devido a construção de usinas hidráulicas em cascata, que faz com que a operação de uma usina à montante interfira na operação das usinas à jusante [7], diferentemente das usinas térmicas que apresentam a operação independente. E o acoplamento temporal ocorre pois, a decisão da utilização dos recursos hídricos em um mês afeta os meses subsequentes, com, por exemplo, o não suprimento da carga (déficit) ou mesmo o vertimento de água dos reservatórios, que representa um desperdício de energia.

Realizar o planejamento da operação do SIN está ligado, portanto, à questão de quanto e quando produzir a energia necessária ao atendimento da demanda pelo menor custo total de operação possível. O quanto produzir está associado à quantidade de energia a ser gerada pelas usinas hidrelétricas, termelétricas (nuclear, óleo, diesel, gás natural e outras), eólicas, de biomassa, solares e a importação de países vizinhos que pode ser considerada como uma termelética. A determinação de quando produzir refere-se à decisão de qual o melhor momento de se gerar pelas usinas hidrelétricas, dada a ocorrência de chuva em cada período de tempo analisado. Esse processo de decisão, também conhecido como o “dilema do operador” está ilustrado na Figura 3.

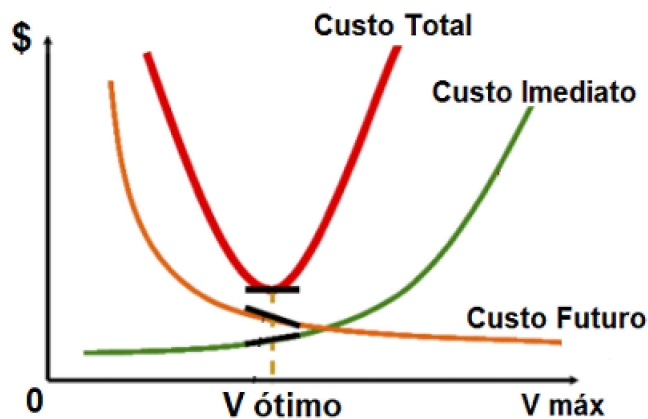
O processo decisório do “dilema do operador” faz com que o custo da operação do sistema hidrotérmico seja composto por parcelas referentes ao custo imediato e ao custo futuro associada à decisão tomada, pois, apesar do custo da água ser considerado nulo, a utilização ou não das reservas hídricas resultam em um custo futuro associada a essa decisão. A Figura 4 apresenta as curvas que representam o custo imediato, futuro e total, em função do volume de água armazenado em um determinado período de tempo.

Figura 3 – Dilema do Operador



Fonte: [2]

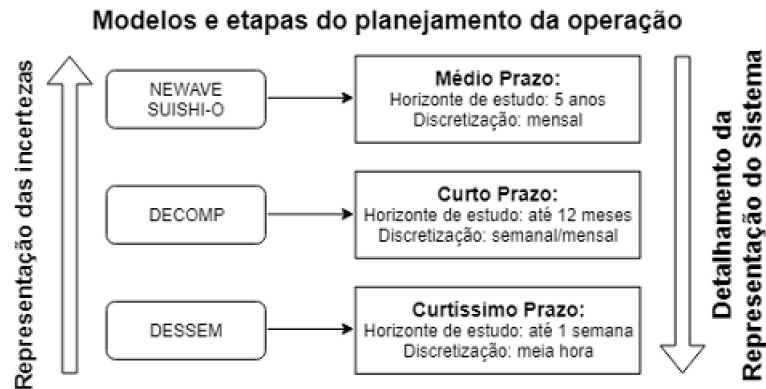
Figura 4 – Função de custo imediato, custo futuro e custo total



Fonte: [2]

Observa-se que os custos dependem do volume armazenado. O custo imediato é maior quando o volume armazenado é maior, ao armazenar água no sistema a demanda é atendida pelas usinas térmicas o que encarece a operação. Em contrapartida, quando o volume armazenado é maior existirá água no sistema para a demanda futura ser atendida por hidrelétricas, barateando a operação futura e consequentemente resultando em um custo futuro menor. Considerando um volume armazenado menor, o custo imediato é menor, ao utilizar a água do sistema a demanda será atendida pelas hidrelétricas barateando a operação. Em compensação, a demanda do futuro deverá ser atendida pelas usinas termelétricas o que elevará o custo da operação futura e consequentemente resultará em um custo futuro maior. Desta maneira, para encontrar o volume armazenado que minimiza o custo da operação deve-se levar em consideração seu impacto nos custos imediatos e futuros. Busca-se então, minimizar o custo total caracterizado como a soma desses custos.

Figura 5 – Modelagem do Sistema Hidrotérmico no Planejamento da Operação



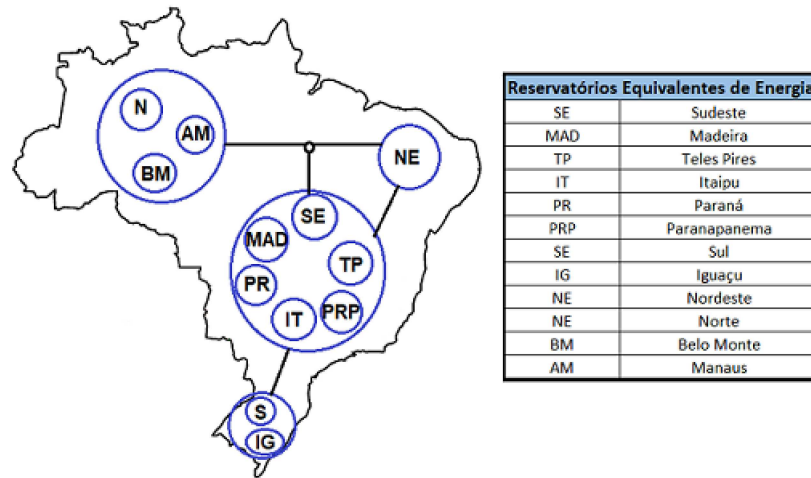
Fonte: Elaborada pelo autor

Devido às características particulares já descritas do SIN, a operação do mesmo deve ser realizada de forma coordenada a fim de minimizar o custo total de operação levando em conta a interdependência operativa entre as usinas, a interconexão dos sistemas elétricos e a estocasticidade das afluições. De acordo com [3] e [8], a interconexão entre os sistemas elétricos e a integração dos recursos de geração e transmissão são de suma importância para o planejamento e redução de custos. Em períodos hidrológicos desfavoráveis em determinadas regiões, outras podem estar em situação hidrológica favorável, reduzindo a necessidade da utilização de energia térmica. Por outro lado, em períodos de déficit hidrológicos as térmicas contribuem para o atendimento da carga.

O problema ainda possui restrições de utilização das usinas térmicas, não linearidades na função de produção das usinas hidrelétricas, interdependência operativa entre as usinas hidrelétricas e incerteza relacionadas às afluições, o que o torna complexo e de difícil solução. Essa complexidade levou à necessidade de construção de modelos específicos para gerenciar a operação elétrica e energética do sistema. O problema é resolvido em diferentes modelos de acordo com o horizonte de estudo, sendo eles de longo, médio e curto prazo e todos esses modelos computacionais são operados de forma coordenada visando minimizar os custos globais do sistema como um todo. As diversas etapas do planejamento da operação estão representadas na Figura 5. Os modelos foram desenvolvidos pelo Centro de Pesquisas de Energia Elétrica (CEPEL) [8] e são homologados pela Agência Nacional de Energia Elétrica (ANEEL) e Operador Nacional do Sistema Elétrico (ONS) com a participação dos agentes do setor elétrico.

A representação do sistema, portanto, varia de acordo com o horizonte a ser analisado. No planejamento da operação de médio prazo, caracterizado pelo horizonte de estudo de até cinco anos com discretização mensal, a estocasticidade das afluições apresenta uma representação detalhada e as usinas hidrelétricas que compõem cada sistema são representadas de forma agregada, por bacias hidrográficas. Desta forma, são

Figura 6 – Reservatórios Equivalentes de Energia



Fonte: Elaborada pelo autor

criados sistemas equivalentes com reservatórios equivalentes de energia, Figura 6, e as não linearidades são representadas de forma indireta através de parábolas de correção [8].

Além disso, o sistema é dividido em quatro submercados que concentram regiões do país (Sul, Sudeste/Centro-Oeste, Nordeste e Norte), Figura 1. Os subsistemas podem trocar energia entre si até um limite máximo de intercâmbio, que representam as restrições elétricas no fluxo de energia entre as diversas regiões do país. Desta etapa resulta uma função multivariada que define o valor econômico da energia armazenada em função dos níveis de armazenamento e afluências aos meses passados, chamada Função de Custo Futuro (FCF) [3].

O planejamento da operação de curto prazo é caracterizado pelo horizonte de estudo de alguns meses e discretização mensal, sendo possível discretizar o mês inicial em estágios semanais. Nesta etapa a incerteza relacionada às afluências aos reservatórios é representada através de uma árvore de vazões, porém no primeiro mês as vazões são determinísticas, e as usinas são representadas de forma individualizada. O objetivo é, a partir da função de custo futuro gerada pelo modelo de médio prazo em um estágio que coincida com o final do horizonte do modelo de curto prazo, gerar uma função que retrate o valor econômico da água armazenada nos reservatórios em função dos níveis de armazenamento dos reservatórios [3] [4] [9].

Por fim a programação diária da operação, é caracterizada pelo horizonte de apenas alguns dias, discretizados em etapas horárias ou de meia em meia hora. Nesta etapa, a incerteza das vazões não é representada. Em contrapartida, o parque hidrotérmico é representado de forma detalhada, levando-se em conta as restrições relativas às máquinas e turbinas, tais como: tomada e alívio de carga, faixas operativas das turbinas, etc. A

rede de transmissão é representada com precisão. A FCF gerada pelo modelo de curto prazo no estágio que coincide com o último estágio no modelo de programação diária é utilizada para definir a meta de geração de cada unidade geradora [3] [5].

A complexidade do problema de planejamento energético de sistemas hidrotérmicos, tais como o SIN, e a constante busca do aprimoramento dos modelos utilizados em sua solução, está sempre fomentando pesquisas no setor elétrico e na comunidade acadêmica.

1.2 MOTIVAÇÃO

O problema de planejamento energético de sistemas hidrotérmicos, como o SIN, é um problema que envolve decisões considerando-se incertezas, como descrito na seção anterior. Por conta desta característica, ele pode ser modelado como um Problema de Programação Estocástica (PPE), o qual busca encontrar uma decisão ótima capaz de minimizar o valor esperado de uma determinada função objetivo baseada em incertezas [12] [11]. Dentro do PPE's encontra-se uma classe especial de problemas referenciada na literatura como Problemas Lineares de Programação Estocástica (PLPE), os quais consideram uma função objetivo e restrições lineares. Para manter o problema nesta classe a Função de Produção Hidráulica, que apresenta não linearidades, é representada por uma aproximação linear por partes.

Além disso, em PLPE's as incertezas são representadas por meio de uma árvore de cenários. Os níveis da árvore representam os estágios de decisão e os ramos os valores discretos para as variáveis aleatórias de cada nó da árvore, assim cada nó possui um possível cenário de vazões afluentes com o objetivo de modelar as incertezas hidrológicas inerentes ao problema. Mesmo com essas considerações, podem ser difíceis de serem resolvidos, principalmente devido à sua grande dimensão.

Quando se considera o problema de planejamento energético de sistemas hidrotérmicos de sistemas reais, como o SIN, tem-se um número muito grande de estágios e possíveis cenários de afluições, o que resulta em uma árvore de cenários com muitos nós. Por conta disso, a formulação equivalente determinística deste problema é alcançada por um Problema Linear (PL) de grandes dimensões, uma vez que o mesmo incluiria as restrições e contribuições para a função objetivo relacionada a todos os cenários da árvore. Assim, dependendo do problema, o tamanho deste PL pode se tornar proibitivo. Para solucionar esta questão, algoritmos de decomposição têm sido propostos na literatura como uma alternativa para resolver esses problemas de grandes dimensões nos quais a abordagem equivalente e determinística provou ser ineficiente ou intratável.

Dentro dos vários algoritmos de decomposição que podem ser utilizados para solucionar o problema destaca-se a decomposição por cortes de Benders [13], técnica na qual o conjunto de variáveis de decisão é explicitamente dividido de acordo com os estágios

do problema e as chamadas variáveis de estado conectam estes estágios, que são resolvidos iterativamente até a convergência do problema. Inicialmente a decomposição de Benders foi aplicada para resolver problemas de dois estágios de PPE's, levando à Programação Linear *L-shaped* [14]. Posteriormente, foi proposta a decomposição de Benders para vários estágios, também conhecida como Programação Dinâmica Dual (PDD) [15], que é amplamente utilizada para resolução de problemas lineares ou convexos com uma representação em árvore de cenários.

No entanto, a resolução de problemas estocásticos através da PDD pode exigir um número considerável de iterações até que a convergência seja alcançada. Como é desejável resolver os problemas de grande proporção sem comprometer o tempo necessário para a obtenção da convergência do mesmo, várias técnicas para melhorar a eficiência dos algoritmos originais foram propostos na literatura com o objetivo de reduzir esse tempo necessário para a solução do problema [16], seja por meio da redução do número de iterações, do tempo de cada iteração, do número de operações necessárias, entre outras estratégias.

Dentro das estratégias para acelerar o tempo de solução do problema, este trabalho dará destaque à paralelização da metodologia PDD, abordagem já considerada em [18], [19], [20] e [16]. A paralelização leva a uma melhoria no tempo de resolução em problemas que compreendem tarefas independentes, e sua implementação só é possível através da utilização de computadores que compreendem muitos processadores, que operam simultaneamente, ou de clusters computacionais. O trabalho propõe um aprimoramento da metodologia proposta em [20] e [16], com o objetivo de quebrar a dependência temporal do algoritmo PDD tradicional e, assim, aumentar sua adequação para ambientes paralelos.

A paralelização aplicada à PDD requer sincronização entre o processador mestre e os demais processadores alocados para os subproblemas, o que pode tornar a implementação para problemas em múltiplos estágios, inviável. Para solucionar este problema, técnicas de agregação de nós podem ser aplicadas. Como abordado em [19], existem diversas técnicas de agrupamentos diferentes, como o agrupamento por decomposição por nó, decomposição por cenário, por subárvore, e por cenário completo. Este trabalho propõe utilizar a técnica de algoritmos genéticos para definir o agrupamento ótimo para os nós da árvore de cenários do problema a ser estudado.

1.3 OBJETIVOS

O principal objetivo desta dissertação é o de propor novas técnicas capazes de melhorar do ponto de vista de eficiência computacional a Programação Dinâmica Dual utilizando processamento paralelo assíncrono. Para isso, alguns tópicos analisados são fundamentais para nortear o estudo em busca do objetivo geral deste trabalho. Compreende os objetivos específicos:

- Fazer uma revisão da literatura sobre o tema, verificando o estado da arte com relação à paralelização da PDD, agrupamentos ótimos de nós da árvore de cenários e algoritmos genéticos.
- Propor uma estratégia de varredura de árvore baseada em uma implementação recursiva do algoritmo da PDD a fim de facilitar a paralelização assíncrona do mesmo.
- Propor um aprimoramento em um algoritmo de paralelização assíncrona da metodologia da PDD proposto em [16], através de um algoritmo genético capaz de encontrar o agrupamento para os nós da árvore de cenários.
- Analisar e comparar as metodologias implementadas com relação ao esforço computacional.
- Testar as metodologias propostas para o SIN considerando subsistemas isolados.

1.4 PUBLICAÇÕES

O desenvolvimento da dissertação levou à publicação do artigo Paralelização Assíncrona da Programação Dinâmica Dual com Agrupamento Ótimo de Nós, Lucas R. Conceição, André L. Diniz, André L. M. Marcato, Amanda Pavila. Aprovado no 14º Simpósio Brasileiro de Automação Inteligente (SBAI) realizado em outubro de 2019, sob organização da Universidade Federal de Ouro Preto (UFOP) com o apoio do Instituto Tecnológico Vale.

1.5 ESTRUTURA DA DISSERTAÇÃO

Este documento está organizado em 5 capítulos, iniciado por esta introdução que aborda as considerações iniciais e motivação do trabalho para contextualizá-lo. Assim como os objetivos que são esperados serem alcançados pelo mesmo e as publicações decorrentes dele.

O Capítulo 2 apresenta a revisão bibliográfica realizada para este trabalho, dando destaque aos conceitos necessários para a realização do mesmo. Estão nele conceitos relacionados à Programação Dinâmica Dual, apresentando o que se encontra na literatura sobre estratégias de varreduras da árvore de cenários, estratégias para o agrupamento dos nós da árvore de cenário para serem resolvidos como subproblemas e estratégias de paralelização e paralelização assíncrona da PDD. Além disso, uma breve revisão sobre algoritmos genéticos e seu uso em aplicações na área de planejamento energético.

Em seguida, o Capítulo 3 apresenta as considerações adotadas e as metodologias implementadas durante a execução do trabalho. Sendo dividido em duas seções. A primeira

aborda a formulação adotada para o problema de planejamento energético de sistemas hidrotérmicos e a segunda a metodologia proposta, mostrando as diferentes implementações necessárias para o seu funcionamento.

No Capítulo 4 estão apresentados os resultados encontrados pelo trabalho, também divididos em seções. Cada seção corresponde a um sistema de estudo diferente e trás os resultados encontrados pela simulação da metodologia implementada. Por fim, o Capítulo 5 apresenta as conclusões e os trabalhos futuros sugeridos. Além disso ao final do trabalho as referências bibliográficas utilizadas são apresentadas.

2 REVISÃO BIBLIOGRÁFICA

2.1 PROGRAMAÇÃO DINÂMICA DUAL

2.1.1 Considerações iniciais

A metodologia Programação Dinâmica Dual (PDD) é um método de decomposição que divide um problema principal em diversos subproblemas que são resolvidos de forma iterativa até que o ótimo global seja encontrado. Essa característica o faz ser muito utilizado para resolver problemas de programação estocástica, como o descrito no primeiro capítulo. Porém, sua característica iterativa pode demandar um número grande de iterações para se alcançar a convergência e, com a divisão do problema, cria-se um elevado número de subproblemas a serem resolvidos a cada iteração. Com isso, a metodologia pode apresentar um elevado tempo de execução dependendo das dimensões do problema principal a ser resolvido.

Para sanar esse problema, são encontradas na literatura diversas estratégias que buscam melhorar a taxa de convergência da PDD, diminuindo o número necessário de iterações, e/ou que buscam reduzir o tempo necessário para cada iteração. O trabalho [16] faz uma revisão bibliográfica detalhada dessas estratégias. Nesta seção as estratégias necessárias e utilizadas no desenvolvimento da metodologia proposta neste trabalho serão revisadas. As seções 2.1.2 e 2.1.3 trazem a revisão das diferentes estratégias de varredura de árvore de cenários e da utilização de técnicas de paralelização na PDD. Destaca-se que a revisão dessas subseções foi baseada na revisão de [16], sendo a seção 2.1.2 atualizada com os resultados encontrados pela metodologia proposta em [16]. Já a seção 2.1.4 traz a revisão das estratégias de agrupamento de nós da árvore de cenários, esta estratégia é fundamental para a implementação do aprimoramento da metodologia de [16], proposto neste trabalho. Essa estratégia também foi revista em [16] porém os diferentes modelos de agrupamentos estão mais detalhados na revisão deste trabalho.

2.1.2 Varredura da árvore de cenários

Uma estratégia que possui um impacto importante na convergência do algoritmo, como mostrado em [22], é a maneira como a árvore de cenários é percorrida, conhecida como estratégia de travessia de árvore. Algumas das estratégias clássicas são a *fast-forward-fast-backward*, abordagem *backward* e abordagem *forward*.

Proposta em [23], a travessia de árvore *fast-forward-fast-backward* consiste em avançar pela árvore até o último estágio, resolvendo todos os subproblemas de todos os estágios da árvore e transferindo os valores das variáveis de estado encontrados. Ao chegar no último estágio a árvore é então percorrida de forma inversa até o primeiro estágio transmitindo informações do custo futuro através dos cortes de Benders. Nesta estratégia,

as informações das variáveis de estado e dos cortes são rapidamente espalhadas por toda a árvore mesmo que em iterações anteriores essa informação não corresponda a pontos na vizinhança dos valores ótimos, ainda desconhecidos, das variáveis de estado.

Outra estratégia clássica é a abordagem *backward*, nessa estratégia a varredura da árvore ocorre sempre no sentido *backward*, ou seja, retrocedendo os estágios exceto quando o nó raiz é alcançado ou quando novas informações não são mais produzidas com relação às iterações anteriores. A construção de cortes que representem a função de custo futuro nos estágios iniciais pode exigir um grande esforço uma vez que estão em regiões distantes para a solução ótima.

Por fim, a abordagem *forward* realiza a varredura da árvore sempre no sentido *forward*, ou seja, avançando na árvore de cenários exceto quando um nó de folha é alcançado ou quando nenhuma nova informação é produzida em relação às iterações anteriores [15]. Nesse caso, a principal preocupação é gastar muito tempo tentando alcançar a convergência nos estágios finais de valores ruins para as variáveis de estado dos estágios anteriores.

Novas estratégias vêm sendo propostas na literatura, uma delas [24] introduz um protocolo chamado de ϵ - *strategy*, no qual a direção com que o algoritmo avança sobre a árvore é alterado caso o gap de convergência local no estágio atual for menor que uma tolerância ϵ . Outra [25], introduz uma técnica de saltos, na qual os autores primeiro definem um estágio $t < T$, onde T é o número total de estágios do problema, e dois tipos de iterações. Uma das iterações é menor, ela avança e retrocede ao longo dos estágios 1 até t . A outra iteração é a principal, que consiste em um avanço completo do estágio 1 para T , depois um passo para trás do estágio T para t , seguido por outro passo para frente do estágio t para T , e finalmente um passo para trás completo de T para 1. O algoritmo procede alternando entre iterações principais, que atravessam a árvore inteira, e iterações menores, tentando evitar estágios finais, que podem ter um grande número de cenários. Mais recentemente, o trabalho [21] utiliza o método *bouncing*, descrito acima e proposto em [25], com uma maneira dinâmica de escolher o estágio t , combinado com estratégias de ϵ - *strategy* [24]. Porém, para a ϵ - *strategy* ele aplica um parâmetro dinâmico calculando uma medida de discrepância que depende do tamanho do gap de convergência.

2.1.3 Paralelização

A estratégia de paralelização leva a uma melhoria no tempo de resolução em problemas que compreendem tarefas independentes, e sua implementação só é possível através da utilização de computadores que compreendem muitos processadores, que operam simultaneamente, ou de clusters computacionais. Além de reduzir drasticamente o tempo de execução de um algoritmo ela ainda permite a introdução de novas técnicas para aumentar a eficiência do algoritmo. Porém, para aplicá-la se faz necessário a adequação do algoritmo para a paralelização e como consequência, a estratégia de paralelização e a

eficiência de um algoritmo estão fortemente relacionadas à sua estrutura, uma vez que as dependências ao longo dos algoritmos criam gargalos no processo paralelo e podem limitar a eficiência e escalabilidade de um esquema de paralelização, reduzindo seu desempenho [16].

Em muitas áreas a computação paralela vem sendo aplicada e são encontrados na literatura muitas abordagens para paralelizar os métodos PDD. A dificuldade para realizar esta paralelização se dá devido a estrutura em árvore dos problemas estocásticos de múltiplos estágios. Neste tipo de problema existe uma alta dependência entre os diferentes níveis da árvore. Considerando a estratégia de travessia da árvore tradicional, por exemplo, cada nó da árvore deve esperar novos valores de variáveis de estado de seu nó ascendente nos passos *forwards*, bem como novos cortes de Benders de seus nós descendentes nos passos *backwards*.

Algumas abordagens de paralelização para problemas de PDD exploram a independência entre os nós do mesmo nível, porém a dependência entre níveis restringe essas propostas. O trabalho [29] resolve subárvores independentes em paralelo e o [27] explora a independência entre níveis resolvendo os mesmos estágios em paralelo. Os autores encontraram uma queda considerável no tempo de execução com um pequeno número de processadores, mas saturação prematura de aceleração à medida que mais processadores foram empregados. E os resultados desses trabalhos evidenciaram que a aceleração e a eficiência dos esquemas de paralelização são muito sensíveis ao tamanho e a forma da árvore de cenários.

Com o intuito de quebrar a dependência entre os níveis da árvore de cenários dos algoritmos PDD, podem ser encontrados na literatura algumas técnicas. Em [30], os autores propuseram um assincronismo parcial para resolver programas estocásticos de dois estágios, método *L-shaped* assíncrono, onde o subproblema do primeiro estágio é resolvido novamente sem esperar que todos os nós do segundo estágio sejam resolvidos, evitando assim processadores inativos. O trabalho mostrou que o esquema proposto não prejudica a convergência do método e pode ser muito adequado para ambientes paralelos heterogêneos.

Em [31] uma abordagem PDD assíncrona foi proposta para um programa estocástico multiestágio, onde o subproblema de um determinado nó é calculado sempre que uma nova variável de estado ou corte estiver disponível. Apesar deste método reduzir a ociosidade dos processadores, o critério de convergência do algoritmo é de alguma forma afetado e os resultados não são muito conclusivos.

Mais recentemente, [20] propôs uma versão assíncrona para Programação Dinâmica Dual Determinística (PDDD). Neste tipo de problema são resolvidos problemas determinísticos e a dependência entre estágios do algoritmo PDD original não permite nenhum nível de paralelização. A proposta do trabalho é a resolução assíncrona do problema, na qual todos os nós são resolvidos em paralelo e depois as informações, variáveis de estado e cortes

de Benders, são trocadas. Desta forma, o problema da dependência entre os problemas é superada e os resultados do trabalho mostraram uma redução de tempo significativa em comparação com o algoritmo sequencial para árvores de qualquer tamanho, embora a eficiência e a aceleração do algoritmo não tenham sido satisfatórias.

Ainda explorando a paralelização assíncrona com o objetivo de quebrar a dependência temporal do algoritmo PDD tradicional e, assim, aumentar sua adequação para ambientes paralelos, encontra-se o trabalho [16]. Ele propõe duas versões assíncronas do algoritmo PDD, que são muito mais adequados para ambientes paralelos em comparação com o algoritmo tradicional, sem perder suas propriedades de convergência. A ideia proposta é executar etapas totalmente assíncronas, com relação à resolução da programação linear dos subproblemas de cada nó, em vez de usar as iterações clássicas com passos *forwards* e *backwards*. O primeiro algoritmo PDD assíncrono permite granularidade homogênea e paralelização completa por nó dentro de cada etapa. A segunda difere-se devido à introdução de um grau de sincronismo, a fim de obter uma melhor taxa de convergência, sendo uma versão parcial do primeiro algoritmo. Os resultados do trabalho mostraram que ambas as abordagens apresentaram melhor aceleração e eficiência em comparação com o algoritmo PDD tradicional.

2.1.4 Agrupamento de nós

Ainda com o intuito de melhorar o desempenho da PDD foi encontrado na literatura estratégias de agrupamento de nós. Elas agrupam os nós da árvore de cenários e os nós que pertencem ao mesmo grupo são resolvidos juntos como subproblemas equivalentes.

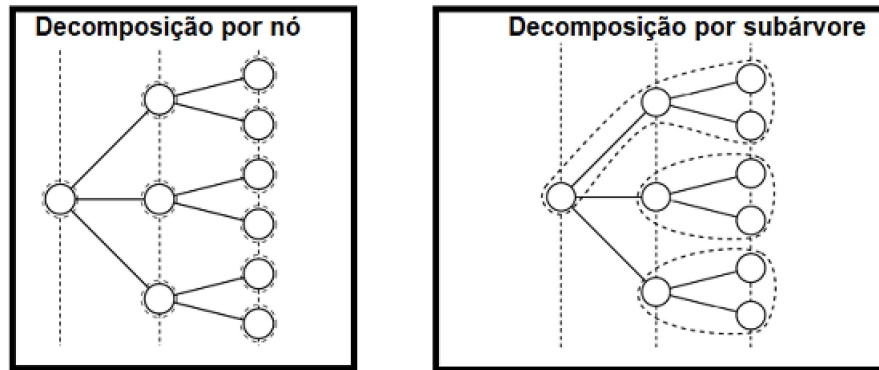
Na literatura são encontrados diferentes modelos de agregação de nós, sendo rotulados como decomposição completa por nó, decomposição por cenário, decomposição por subárvore e decomposição por cenário completo. Estes modelos são propostos e estudados em [10], [26], [27], [19] e [28].

O método de decomposição por nó considera cada nó sendo um subproblema, Figura 7, e tem como vantagem reduzir os subproblemas para o menor tamanho possível. Porém possui a desvantagem de requerer um número maior de iterações por cada subproblema estar atento a uma fração pequena do problema total [19].

Já a decomposição por subárvore, agrupa os nós tentando replicar a estrutura da árvore em uma escala menor. Um dos agrupamentos o nó do primeiro estágio juntamente com um cenário completo, Figura 7. [19].

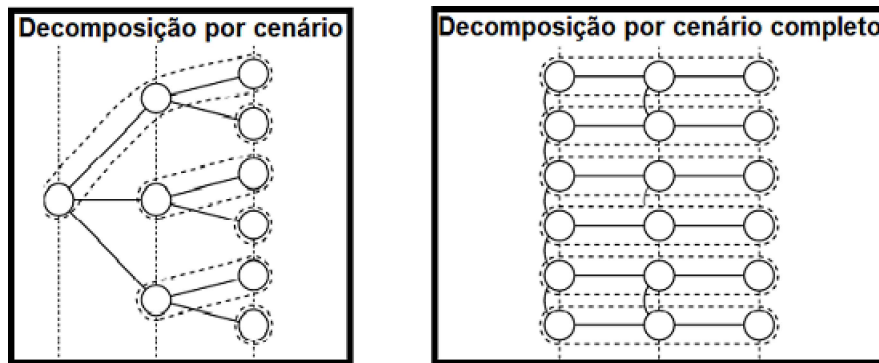
Já o método de decomposição por cenário agrupa os nós que pertencem ao mesmo cenário que não tenham sido já agrupados, Figura 8. Dessa forma, ao menos um cenário completo é incluído nos subproblemas, o que pode acelerar a solução já que evita os casos divergentes do esquema anterior [19].

Figura 7 – Tipos de agrupamento de nós (1)



Fonte: [19]

Figura 8 – Tipos de agrupamento de nós (2)



Fonte: [19]

Por fim tem-se a decomposição completa por cenário, apresentada em [19] e que visa resolver o problema da assimetria dos métodos anteriores que causam ociosidade dos recursos, pois na medida que o processo de solução caminha para os estágios finais mais recursos são necessários. Neste método, como todos cenários são completos, todo subproblema tem solução caso exista solução para a árvore completa, Figura 8.

Esses métodos são muito sensíveis ao tamanho e à forma da árvore de cenários, o que dificulta a avaliação e a comparação dessas estratégias para problemas gerais. Quando muitos nós são agregados em um mesmo grupo, os subproblemas correspondentes tornam-se mais complexos e exigem mais memória e tempo da Unidade Central de Processamento (CPU) para serem resolvidos. Por outro lado, quando os nós são desagregados, os subproblemas são mais simples e mais rápidos de serem resolvidos, mas a granularidade é maior, o que pode levar a uma taxa de convergência mais lenta [19].

Figura 9 – Pseudocódigo algoritmo evolucionista

```

Algoritmo evolucionista
Início
  t=0
  Inicializa população(t)
  Avalia população(t)
  Enquanto não termina o processo faça
    t = t + 1
    Selecione a população(t) a partir da população(t-1)
    Altere a população(t)
  Fim Enquanto
Fim

```

Fonte: Elaborada pelo autor

2.2 ALGORITMOS GENÉTICOS

Ao longo da história, os mecanismos heurísticos mais poderosos foram inspirados pela natureza. Um exemplo são os Algoritmos Evolucionistas (AE) [32] que imitam a teoria da evolução das espécies de Charles Darwin [33] onde, indivíduos com maior capacidade de adaptação ao meio ambiente, tem maiores chances de sobrevivência. Ainda pela teoria, as características genéticas são passadas de geração para geração, com isso, os indivíduos gerados possuem material genético dos indivíduos anteriores, geralmente escolhidos entre os mais aptos [33].

Embora atualmente existem muitas variantes diferentes de AE's, a ideia comum por trás de todas essas técnicas permanece a mesma. Dentro da população de indivíduos (as soluções) a pressão ambiente provoca a seleção natural, (sobrevivência do indivíduo) causando um aumento na tensão da população [34]. Portanto, primeiro, ocorre a inicialização da população, geralmente de forma randômica, devendo cada indivíduo ser avaliado e associado a uma medida de aptidão. Entra-se, então, em um ciclo evolutivo, até que um critério de convergência seja alcançado onde, indivíduos são selecionados para a reprodução, gerando filhos por recombinação/mutação para uma nova população e, por fim, os novos indivíduos (filhos) serão avaliados novamente, reiniciando o ciclo. Cada AE executa estes passos de uma forma específica, mas a Figura 9 apresenta o pseudocódigo de um algoritmo evolucionista.

Segundo [35], as três diferenças mais significativas entre os AE's e os métodos de busca clássicos são a busca por meio de uma população de soluções em cada iteração, ao invés de um único candidato, a falta de necessidade de muita informação do problema e a utilização de regras de transição probabilística, ao invés de determinísticas. Estas características tornam os AE's flexíveis, permitindo que sejam aplicados e/ou adaptados para diversos tipos de problemas.

Em 1975, John Holland [36] propõe os chamados Algoritmos genéticos (AG's) para

simular sistemas adaptativos que foram popularizados por Goldberg [37] [38]. Os AGs são métodos heurísticos de otimização e busca inspirados nos mecanismos de evolução de populações de seres vivos e, portanto, um AE.

O AG possui funcionamento similar ao dos AEs descrito acima, fazendo uso de uma nomenclatura utilizada em genética das populações para definir seus componentes e operadores. Portanto, seu funcionamento se inicia com a geração de uma população inicial de cromossomos, que representam possíveis soluções de um problema. A população é avaliada e cada cromossomo recebe uma avaliação, ou seja, a sua aptidão. Tendo como base a lei de Darwin, são selecionados os cromossomos para a próxima geração, adotando maior probabilidade para os mais aptos. Os cromossomos selecionados transferem suas características para seus descendentes através da reprodução, simulada no AG por meio dos operadores genéticos de recombinação e mutação. E por fim, todo o processo é repetido até que seja encontrada uma solução satisfatória.

Na área de planejamento energético a literatura apresenta diversos trabalhos que utilizam AG's. Em [39] é apresentada uma revisão de técnicas evolutivas aplicadas a sistemas de potência e em [40] é investigada a aplicação de algoritmos genéticos na otimização da operação de sistemas hidrotérmicos de geração. No trabalho [41], é apresentada a aplicação de Algoritmos Genéticos na otimização da operação de sistemas hidrotérmicos de potência e em [42] é proposto uma modelagem utilizando AG's, de modo a considerar múltiplos objetivos no processo de otimização sem que a representação física do sistema energético e suas incertezas sejam comprometidas.

3 METODOLOGIA PROPOSTA

3.1 FORMULAÇÃO PLANEJAMENTO ENERGÉTICO

O problema de planejamento energético de sistemas hidrotérmicos tem um papel importante nos estudos do planejamento da operação e expansão de sistemas de geração de energia. Como já citado no primeiro capítulo, existem diferentes maneiras de se modelar o sistema de acordo com o detalhe da representação das incertezas e o horizonte de tempo que se deseja para o estudo. Este trabalho modela o problema através de uma função objetivo linear com restrições lineares, destaca-se que foram consideradas as não linearidades da Função de Produção Hidráulica (FPH) das usinas hidroelétricas através de uma aproximação linear por partes. A modelagem considera o problema como de curto prazo, ou seja, representação individual de usinas, térmicas e hidrelétricas, com hidrelétricas em cascata e discretização temporal mensal.

A seguir, a formulação matemática da função objetivo e das restrições de cada subproblema linear dos nós da árvore de cenários do problema serão apresentadas. Cada nó (t, s) está relacionado a um período de tempo t e um cenário de entrada s , com um Problema Linear (PL) correspondente. Os parâmetros de entrada deste subproblema são as chamadas variáveis de estado, que são os armazenamentos nos reservatórios no início do cenário. A formulação matemática da função objetivo e restrições do PL de um dado nó (t, s) é apresentada abaixo, para este trabalho não considerou-se a representação da rede elétrica.

3.1.1 Função objetivo

A função objetivo de cada PL individual é composta pelo somatório dos custos de geração térmica, o custo de um possível déficit e uma estimativa de custos futuros. Os custos de geração térmica juntamente com o custo de déficit formam o custo atual. E a estimativa de custos futuros é retirada das Funções de Custo Futuro (FCF's) construídas com os Cortes de Benders através da solução do problema PDD. Além dessas parcelas, um custo artificial para o vertimento de água também é incluído para evitar o desperdício da mesma, a função objetivo completa pode ser vista na equação (3.1).

$$\min\left[\left(\sum_{i=1}^{NT} ct_i gt_i^{t,s}\right) + c_{def} def^{t,s} + \left(\sum_{j=i}^{NH} K_{pen} vv_j^{t,s}\right) + \alpha^{t+1,s}\right] \quad (3.1)$$

Sendo:

NT = Número de térmicas do sistema em estudo

NH = Número de hidrelétricas do sistema em estudo

t = Período de tempo do nó

s = Cenário de entrada do nó
 ct_i = Custo de geração de energia da usina térmica i
 $gt_i^{t,s}$ = Geração Termelétrica da usina i no período de tempo t para o cenário de entrada s
 c_{def} = Custo do déficit de energia
 $def^{t,s}$ = Valor de déficit de energia no período t para o cenário de entrada s
 K_{pen} = Constante de penalização do vertimento
 $vv_j^{t,s}$ = Vazão Vertida da usina j no período de tempo t para o cenário de entrada s
 $\alpha^{t+1,s}$ = Estimativa do custo futuro para o período de tempo $t+1$ para o cenário de entrada s

3.1.2 Função de custo futuro

A aproximação linear por partes da função de custo futuro (FCF) ou corte de Benders do estágio t , dado pelo conjunto de (hiper)planos lineares que fornece uma região convexa de solução é apresentada na equação (3.2).

$$\alpha^{t+1,s} \geq \sum_{j=1}^{NH} (\lambda_{j,k}^{t+1,s} va_j^{t,s}) + K_{indep_k}^{t+1,s} \quad \forall k \in NCB \quad (3.2)$$

Sendo:

NH = Número de hidrelétricas do sistema em estudo
 t = Período de tempo do nó
 s = Cenário de entrada do nó
 NCB = Número de cortes de Benders
 $\alpha^{t+1,s}$ = Estimativa do custo futuro para o período de tempo $t+1$ para o cenário de entrada s
 $\lambda_{j,k}^{t+1,s}$ = Coeficiente do corte k da FCF que multiplica o volume armazenado da hidrelétrica j , dado pelo multiplicador de Lagrange da equação de balanço hídrico da usina j no período de tempo $t + 1$ para o cenário s
 $va_j^{t+1,s}$ = Volume armazenado da usina j ao final do período de tempo t para o cenário de entrada s
 $K_{indep_k}^{t+1,s}$ = Coeficiente independente do corte k da FCF no estágio $t + 1$ para o cenário de entrada s

3.1.3 Atendimento à demanda

A restrição de atendimento à demanda ou balanço energético do sistema, pode ser vista na equação (3.3). Por ela, a soma das energias hidrelétrica e termelétrica geradas mais um possível déficit de energia deve ser igual à demanda do sistema,

$$\sum_{i=1}^{NT} gt_i^{t,s} + \sum_{j=1}^{NH} gh_j^{t,s} + def^{t,s} = DEM_t \quad (3.3)$$

Sendo:

NT = Número de térmicas do sistema em estudo

NH = Número de hidrelétricas do sistema em estudo

t = Período de tempo do nó

s = Cenário de entrada do nó

$gt_i^{t,s}$ = Geração termelétrica da usina i no período de tempo t para o cenário de entrada s

$gh_j^{t,s}$ = Geração hidrelétrica da usina j no período de tempo t para o cenário de entrada s

$def^{t,s}$ = Valor de déficit de energia no período t para o cenário de entrada s

DEM_t = Valor da demanda de energia do sistema em estudo no período t para o cenário de entrada s

3.1.4 Balanço hídrico

Para cada usina do sistema, o volume da usina armazenado ao final de um estágio t deve ser igual à soma de seu volume inicial com o volume incremental afluyente e o volume defluyente das usinas à montante, menos o volume defluyente da usina em questão. O volume incremental afluyente é proveniente do volume lateral entre a usina e suas usinas à montante, e o volume defluyente é dado pela soma entre os volumes turbinado e vertido. A formulação deste balanço hídrico é apresentada na equação (3.4).

$$va_j^{t+1,s} = va_j^{t,s} + K_{conv} (Vinc_j^{t,s} + \sum_{m \in M} (vt_m^{t,s} + vv_m^{t,s}) - vt_j^{t,s} - vv_j^{t,s}) \quad \forall j \in J \quad (3.4)$$

Sendo:

J = Conjunto de todas as usinas hidrelétricas do sistema

M = Conjunto de todas as usinas hidrelétricas à montante da usina j

t = Período de tempo do nó

s = Cenário de entrada do nó

K_{conv} = Constante de conversão de unidades, de m^3/s para hm^3 , de valor igual a 2,592

$va_j^{t+1,s}$ = Volume armazenado da usina j ao final do período de tempo t para o cenário de entrada s

$va_j^{t,s}$ = Volume armazenado da usina j no início do período de tempo t para o cenário de entrada s

$Vinc_j^{t,s}$ = Vazão incremental afluyente da usina j no período de tempo t para o cenário de entrada s

$vt_m^{t,s}$ = Vazão turbinada pela usina m a montante da usina j no período de tempo t para o cenário de entrada s

$vv_m^{t,s}$ = Vazão vertida pela usina m a montante da usina j no período de tempo t para o cenário de entrada s

$vt_j^{t,s}$ = Vazão turbinada pela usina j no período de tempo t para o cenário de entrada s

$vv_j^{t,s}$ = Vazão vertida da usina j no período de tempo t para o cenário de entrada s

3.1.5 Função de produção hidráulica

A geração de uma usina hidrelétrica é determinada através da altura de queda líquida da usina, a qual pode ser modelada por polinômios de quarto grau em função do armazenamento, turbinamento e vertimento da usina. Porém, com essa representação o problema de otimização se tornaria não linear devido à introdução do polinômio de maior grau no problema. Como problemas não lineares apresentam uma maior complexidade para serem resolvidos, Diniz e Maceira [43] propuseram uma aproximação da geração hidrelétrica denominada Função de Produção Hidráulica Aproximada (FPHA). O objetivo é realizar uma aproximação linear por partes da função de geração hidrelétrica não linear garantindo a convexidade da região viável definida pela função, preservando a variação da produtividade com a altura de queda mas mantendo a modelagem do problema como linear.

Para o trabalho, a aproximação linear por partes da geração hidrelétrica adotada foi a estratégia de discretização do espaço de estados apresentada em [44]. Os valores discretos do volume armazenado no reservatório, da vazão turbinada e da vazão vertida são determinados e realiza-se a discretização do volume armazenado em N_v valores entre o volume armazenado máximo e mínimo, discretização da vazão turbinada em N_q valores entre a vazão turbinada máxima e mínima e discretização da vazão vertida em N_s valores

entre zero e a vazão vertida máxima. São obtidos $N = N_v N_q N_s$ pontos discretizados e cada n ponto (V_x, Q_y, S_z) determina um valor de geração hidrelétrica (GH_n), dada pela equação (3.5).

$$GH_n = \rho_{esp} Q_y [h_{mon}(V_x) - h_{jus}(Q_y, S_z) - h_{perdas}] \quad (3.5)$$

Sendo:

GH_n = Geração hidráulica do ponto n

ρ_{esp} = Produtibilidade específica da usina

Q_y = Vazão turbinada do ponto n

V_x = Volume armazenado do ponto n

S_z = Vazão vertida do ponto n

h_{mon} = Altura de queda líquida montante

h_{jus} = Altura líquida de jusante

h_{perdas} = Altura líquida de perdas

Com isso, N pontos (V_x, Q_y, S_z, GH_n) são determinados e desses pontos através de um algoritmo *Convex Hull* [45] encontra-se a superfície convexa delimitada pelos pontos. No trabalho [44] a biblioteca Qhull [46], que realiza este procedimento de forma otimizada, foi utilizada obtendo um conjunto de P hiperplanos que representam os pontos discretizados como uma região convexa. A equação (3.6) apresenta a restrição da geração hidrelétrica, para este trabalho esta função por partes é calculada a priori e adicionada ao modelo.

$$gh_j^p \leq \gamma_{0_j}^p + \gamma_{va_j}^p va_j^{t,s} + \gamma_{vt_j}^p vt_j^{t,s} + \gamma_{vv_j}^p vv_j^{t,s} \quad \forall p \in P_j \text{ e } \forall j \in J \quad (3.6)$$

Sendo:

J = Conjunto de todas as usinas hidrelétricas do sistema

P_j = Conjunto de todos hiperplanos obtidos para a usina j

$va_j^{t+1,s}$ = Volume armazenado da usina j ao final do período de tempo t para o cenário de entrada s

$vt_j^{t,s}$ = Vazão turbinada pela usina j no período de tempo t para o cenário de entrada s

$vv_j^{t,s}$ = Vazão vertida da usina j no período de tempo t para o cenário de entrada s

γ_{0j}^p = Parâmetro constante do hiperplano p para a usina j

$\gamma_{va_j}^p$ = Parâmetro do hiperplano p associado ao volume armazenado da usina j

$\gamma_{vt_j}^p$ = Parâmetro do hiperplano p associado à vazão turbinada da usina j

$\gamma_{vv_j}^p$ = Parâmetro do hiperplano p associado à vazão vertida da usina j

3.1.6 Limites operativos das variáveis

Por fim, as equações (3.7) a (3.15) apresentam os limites operativos das variáveis do problema.

$$VA_j^{min} \leq va_j^{t+1,s} \leq VA_j^{max} \quad \forall j \in J \quad (3.7)$$

$$0 \leq vt_j^{t,s} \leq VT_j^{max} \quad \forall j \in J \quad (3.8)$$

$$0 \leq vv_j^{t,s} \leq VV_j^{max} \quad \forall j \in J \quad (3.9)$$

$$0 \leq gt_i^{t,s} \leq GT_i^{max} \quad \forall i \in I \quad (3.10)$$

$$def^{t,s} \geq 0 \quad (3.11)$$

$$\alpha^{t+1,s} \geq 0 \quad (3.12)$$

Sendo:

$va_j^{t+1,s}$ = Volume armazenado da usina j ao final do período de tempo t para o cenário de entrada s

$vt_j^{t,s}$ = Vazão turbinada pela usina j no período de tempo t para o cenário de entrada s

$vv_j^{t,s}$ = Vazão vertida da usina j no período de tempo t para o cenário de entrada s

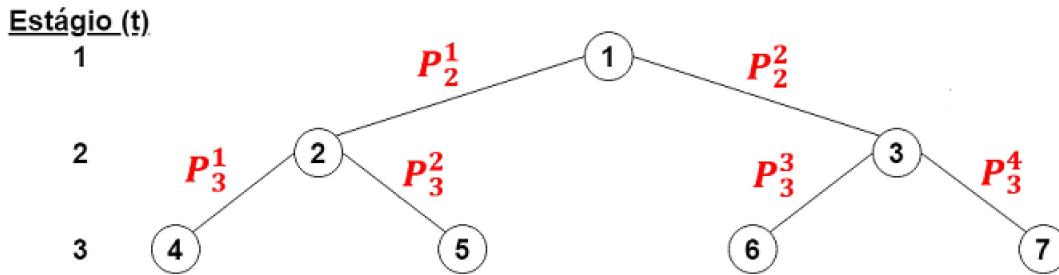
$gt_i^{t,s}$ = Geração Termelétrica da usina i no período de tempo t para o cenário de entrada s

VA_j^{min} = Volume de armazenamento mínimo da usina j

VA_j^{max} = Volume de armazenamento máximo da usina j

VT_j^{max} = Vazão máxima que pode ser turbinada pela usina j

Figura 10 – Exemplo árvore de cenários



Fonte: Elaborada pelo autor

VV_j^{max} = Vazão máxima que pode ser vertida pela usina j

GT_i^{max} = Capacidade máxima de geração da termelétrica j

3.2 PROGRAMAÇÃO DINÂMICA DUAL

3.2.1 Decomposição de Benders

O problema a ser resolvido, o planejamento energético de um sistema hidrotérmico modelado como descrito na seção 3.1, é um problema de otimização de multiestágio com cenários discretizados. Caracterizado por discretizações de tempo $T = 1, \dots, T$ onde cada estágio $t \in T$ é composto por subproblemas, ou nós, denotados como $SubP_t^s$, onde s é um cenário do estágio t .

A Figura 10 ilustra um exemplo simples de uma árvore de cenários com 3 estágios ($T = 3$) e duas aberturas. Cada nó é associado a dois possíveis cenários descendentes, com um total de 7 nós ou subproblemas.

Define-se D_{t-1}^u como o conjunto de cenários descendentes do subproblema $SubP_t^u$, então $s \in D_{t-1}^u$ se e somente se $SubP_t^s$ for descendente dos subproblemas $SubP_{t-1}^u$. A probabilidade do cenário s do estágio t dado que o cenário u tenha ocorrido e $s \in D_{t-1}^u$ é dada por P_t^s .

Considerando o exemplo de árvore de cenários da Figura 10, a maneira mais intuitiva para resolver um problema de minimização como este seria agregar todos os estágios o transformando em um problema único que contenha todas as restrições e contribuições de todos os cenários com suas respectivas probabilidades para a função objetivo. Este tipo de metodologia é chamada de Problema Determinístico Equivalente e sua formulação para o exemplo da Figura 10 está na equação (3.13).

$$\min f_1(x_1) + \sum_{u \in D_1} P_2^u (f_2(x_2^u) + \sum_{s \in D_2^u} P_3^s [(f_3(x_3^s))])$$

(3.13)

Sendo todas as funções e restrições lineares, o problema da equação (3.13) pode ser resolvido com métodos de programação linear. No entanto, à medida em que a árvore de cenários se torna mais complexa, incrementando o número de estágios e aberturas, o problema torna-se proibitivo de ser resolvido por esses métodos. A alternativa encontrada para contornar o aumento da dimensionalidade são os algoritmos de decomposição. Neste trabalho a estratégia utilizada foi a Programação Dinâmica Dual (PDD).

A PDD faz uso da Decomposição de Benders [13], que consiste em dividir o problema em diversos subproblemas os quais são dependentes entre si. A fim de exemplificar a aplicação da decomposição de Benders considera-se um problema de minimização genérico como o da equação (3.14).

$$\min f_1(x_1, y_1) + f_2(x_2) \quad (3.14)$$

Sujeito a:

$$g_1(x_1, y_1) \geq B_1$$

$$g_2(x_2, y_1) \geq B_2$$

Sendo:

$x_1, x_2 =$ Variáveis que podem assumir valores distintos no problema original

$f_1(x_1, y_1) + f_2(x_2) =$ função objetivo do problema original

$g_1(x_1, y_1) \geq B_1$ e $g_2(x_2, y_1) \geq B_2 =$ Restrições do problema original

O problema original é então dividido em dois subproblemas um mestre e outro escravo, equações (3.15) e (3.16) respectivamente, que são resolvidos separadamente e de forma iterativa até que ocorra a convergência do problema.

$$\min f_1(x_1, y_1) + \alpha \quad (3.15)$$

Sujeito a:

$$g_1(x_1, y_1) \geq B_1$$

$$\min f_2(x_2) \quad (3.16)$$

Sujeito a:

$$g_2(x_2, y_1^*) \geq B_2 \quad (\lambda_2)$$

Sendo:

x_1, x_2 = Variáveis que podem assumir valores distintos nos subproblemas mestre e escravo

y_1 = Variáveis de decisão, calculadas durante o subproblema mestre

y_1^* = Valor encontrado para y_1 no subproblema mestre e mantido como parâmetro no subproblema escravo

$f_1(x_1, y_1)$ = Parcela da função objetivo do problema original que irá compor a função objetivo do subproblema mestre

α = Parcela que compõe a função objetivo do subproblema mestre e representa a estimativa do custo do subproblema escravo

$f_2(x_2)$ = Parcela da função objetivo do problema original que será a função objetivo do subproblema escravo

$g_1(x_1, y_1) \geq B_1$ = Restrições do subproblema mestre

$g_2(x_1, y_1^*) \geq B_2$ = Restrições do subproblema escravo

λ_2 = Valor do coeficiente de Lagrange associado à restrição do subproblema escravo

Nota-se que a restrição do subproblema escravo, equação (3.16), possui um coeficiente de Lagrange (λ_2) associado. Esse coeficiente representa o quanto o custo da função objetivo alteraria com o acréscimo de uma unidade no valor do termo independente (B_2) da restrição do subproblema escravo. Evidenciando o impacto da restrição sobre a função objetivo do subproblema escravo.

Para verificar se ocorreu convergência durante o processo iterativo, são calculados os limites superiores e inferiores do problema, equações (3.17) e (3.18). A convergência ocorre quando a diferença absoluta entre esses limites for inferior a uma tolerância (ϵ) pré-determinada, equação (3.19).

Limites:

$$Z_{inf} = f_1(x_1, y_1) + \alpha \quad (3.17)$$

$$Z_{sup} = f_1(x_1, y_1) + f_2(x_2) \quad (3.18)$$

Convergência:

$$\frac{Zsup - Zinf}{Zsup} \leq \epsilon \quad (3.19)$$

Sendo:

x_1, x_2 = Variáveis que podem assumir valores distintos nos subproblemas mestre e escravo

y_1 = Variáveis de decisão, calculadas durante o subproblema mestre

α = Parcela que compõe a função objetivo do subproblema mestre e representa a estimativa do custo do subproblema escravo

$f_1(x_1, y_1)$ = Parcela da função objetivo do problema original que irá compor a função objetivo do subproblema mestre

$f_2(x_2)$ = Parcela da função objetivo do problema original que será a função objetivo do subproblema escravo

$Zinf$ = Limite inferior do problema

$Zsup$ = Limite superior do problema

ϵ = Tolerância para convergência pré-determinada

Caso não tenha ocorrido convergência o corte de Benders, equação (3.20), deve ser inserido no subproblema mestre para dar continuidade ao processo iterativo. Podemos expandir essa equação para o caso de múltiplas variáveis de decisão e múltiplas restrições associadas a elas, equação (3.21).

Corte de Benders:

$$\alpha + \lambda_2 E_{y_1} y_1 \geq w_2 + \lambda_2 E_{y_1} y_1^* \quad (3.20)$$

$$\alpha + \sum_{y_1 \in V_{y_1}} \left(\sum_{r \in Rest_{y_1}^*} (\lambda_2^r E_{y_1}^r) y_1 \right) \geq w_2 + \sum_{y_1 \in V_{y_1}} \left(\sum_{r \in Rest_{y_1}^*} (\lambda_2^r E_{y_1}^r) y_1^* \right) \quad (3.21)$$

Sendo:

α = Parcela que compõe a função objetivo do subproblema mestre e representa a estimativa do custo do subproblema escravo

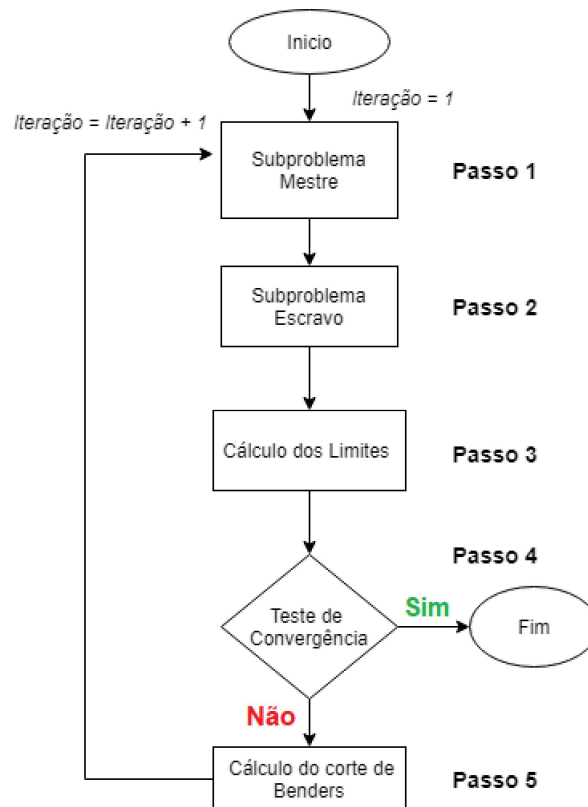
λ_2 = Valor do coeficiente de Lagrange associado à restrição do subproblema escravo

E_{y_1} = Derivadas parciais das restrições $g_2(x_1, y_1^*)$ em relação a y_1^*

y_1 = Variáveis de decisão, calculadas durante o subproblema mestre

w_2 = Valor da função objetivo obtido do subproblema escravo

Figura 11 – Fluxograma Decomposição de Benders



Fonte: Elaborada pelo autor

y_1^* = Valor encontrado para y_1 no subproblema mestre e mantido como parâmetro no subproblema escravo

V_{y_1} = Conjunto das variáveis de decisão y_1 presentes no subproblema mestre

$Rest_{y_1^*}$ = Conjunto de restrições presentes no subproblema escravo que incluem o parâmetro y_1^*

λ_r^2 = Valor do coeficiente de Lagrange obtido do subproblema escravo associado à restrição r

$E_{y_1^r}$ = Derivada parcial da restrição r em relação a y_1

A Figura 11 apresenta o fluxograma que resume a técnica de decomposição de Benders. Após a inicialização do problema, a técnica otimiza o subproblema mestre buscando os valores de decisão y_1^* considerando uma estimativa para o valor ótimo do subproblema escravo, α . Este é o passo 1 da técnica, vale ressaltar que para a primeira iteração o valor de α é considerado 0, pois não há restrições para ele.

Dando continuidade à solução do problema, no passo 2 o problema escravo é otimizado considerando o valor de y_1^* encontrado pela solução do problema mestre. Esse

valor serve de parâmetro para o problema escravo que é solucionado para a determinação dos valores do multiplicador de Lagrange λ_2 e do custo w_2 necessários para o cálculo do corte de Benders, passo 5.

Porém, antes de calcular o corte de Benders são realizados os cálculos dos limites superiores e inferiores do problema. Com esses valores será realizado o teste de convergência da técnica, passos 3 e 4. Se houver convergência o processo iterativo é finalizado, caso contrário o corte de Benders é calculado, passo 5. O corte é então incluído ao subproblema mestre que será resolvido novamente, iniciando uma nova iteração.

Para resolver problemas com árvores de cenários, como o exemplo da Figura 10, existem diferentes técnicas dentro da PDD. As próximas subseções irão apresentar a técnica clássica e em seguida as técnicas propostas por este trabalho.

3.2.2 Programação dinâmica dual tradicional

A solução clássica da PDD realiza uma varredura de árvore tradicional. Essa varredura é composta por duas etapas, uma etapa *forward* e uma etapa *backward*. A Figura 12 apresenta o fluxograma desta técnica.

Após inicializar o problema, a etapa *forward* tem início com a resolução do subproblema do primeiro estágio ($t=1$) e termina com a resolução do subproblema do último estágio ($t=T$). Durante as soluções dos subproblemas os valores encontrados para as variáveis de decisão do estágio t , y_t^* , são armazenadas para servirem de parâmetro para a etapa *backward*.

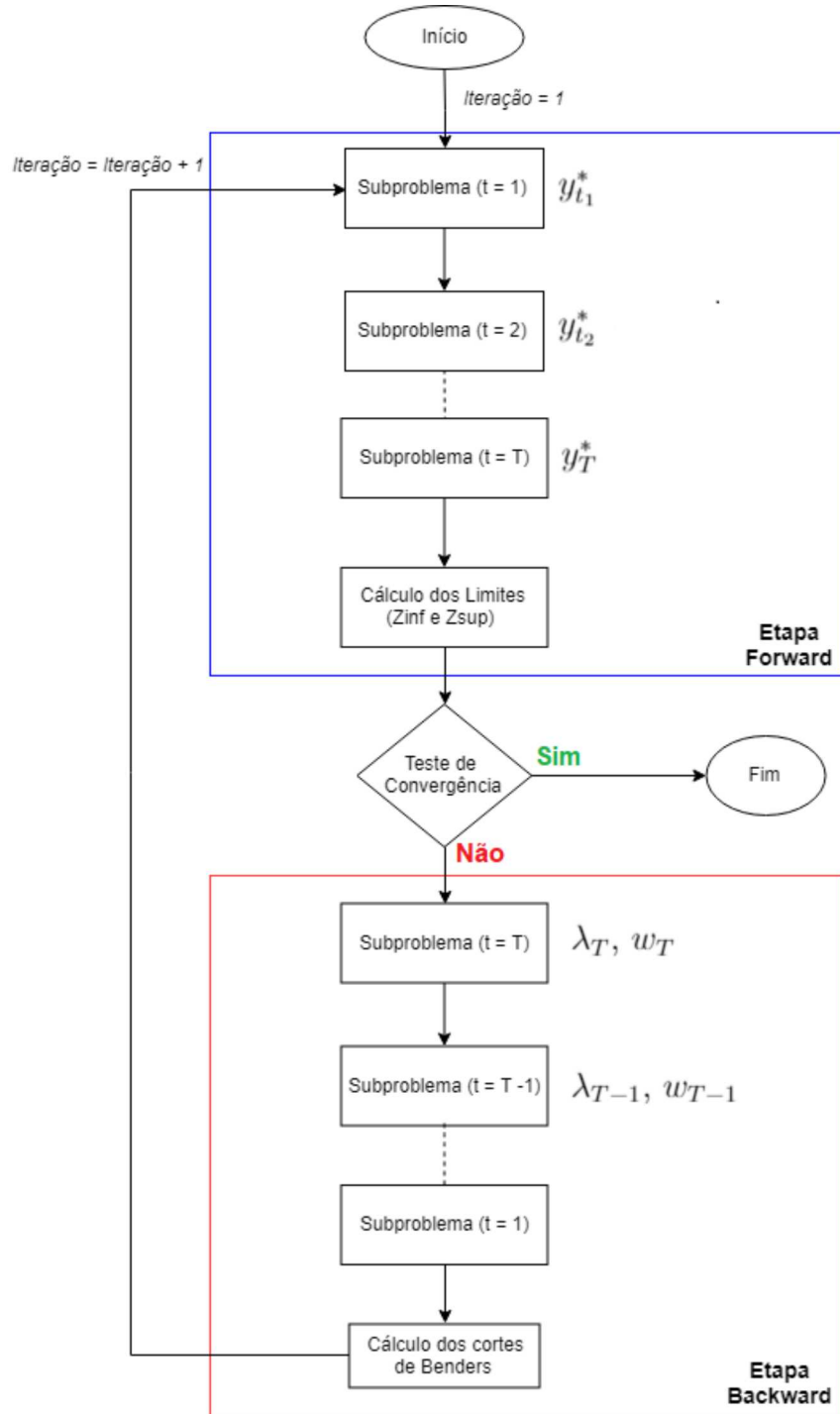
Ao final da etapa *forward*, é realizado o teste de convergência com os valores calculados para os limites superiores e inferiores da iteração corrente através das equações (3.17), (3.18) e (3.19). Caso não tenha ocorrido convergência a etapa *backward* se inicia.

No processo *backward* segue-se caminho contrário do anterior, iniciando na resolução do subproblema do último estágio ($t=T$) e finalizando na resolução do subproblema do primeiro estágio ($t=1$). Como dito anteriormente os valores calculados na etapa *forward* são utilizados como parâmetros para os subproblemas nesta etapa.

Durante as soluções dos subproblemas os valores da função objetivo obtido e o valor do coeficiente de Lagrange (λ_t e w_t) necessários para a construção dos cortes de Benders de cada estágio são armazenados. Assim ao final da etapa *backward*, são contruídos cortes de Benders, equação (3.20) ou (3.21), para cada subproblema. Estas restrições serão consideradas durante a nova etapa *forward* iniciando assim uma nova iteração.

Novas soluções ótimas são encontradas para as variáveis de decisão y_t e novos cortes são construídos à medida que o processo iterativo prossegue. Ele terá fim quando a diferença absoluta entre o limite inferior (Z_{inf}) e o limite superior (Z_{sup}) for igual à tolerância ϵ adotada, ponto este que corresponde à convergência do problema.

Figura 12 – Fluxograma PDD tradicional



Fonte: Elaborada pelo autor

3.2.3 Programação dinâmica dual recursiva

Neste trabalho adotou-se uma implementação do algoritmo PDD de forma recursiva, fazendo com que a estratégia de varredura da árvore fosse diferente da apresentada na subseção anterior. A modelagem do problema, assim como o mecanismo de solução são análogos aos apresentados nas subseções anteriores.

A diferença na implementação do algoritmo de forma recursiva está na ordem de solução das etapas *forward* e *backward*. Elas se misturam durante a solução do problema, criando uma nova estratégia de varredura da árvore, seu fluxograma é apresentado na Figura 13.

A introdução da recursividade no processo de solução da metodologia ocorre através da função em destaque azul Resolve Nó. Ela possui como parâmetro de entrada o número do estágio do subproblema que irá resolver.

O processo se inicia no primeiro estágio ($t=1$), o subproblema equivalente à etapa *forward* deste estágio é resolvido e o valor encontrado para a variável de decisão, y_t^* , é armazenado. Em seguida, como o primeiro estágio ($t=1$) não é igual a T , a metodologia chama a função Resolve Nó para o estágio seguinte ($t=t+1$). O subproblema deste estágio equivalente à etapa *forward* é resolvido e o valor encontrado para a variável de decisão y_t^* armazenado.

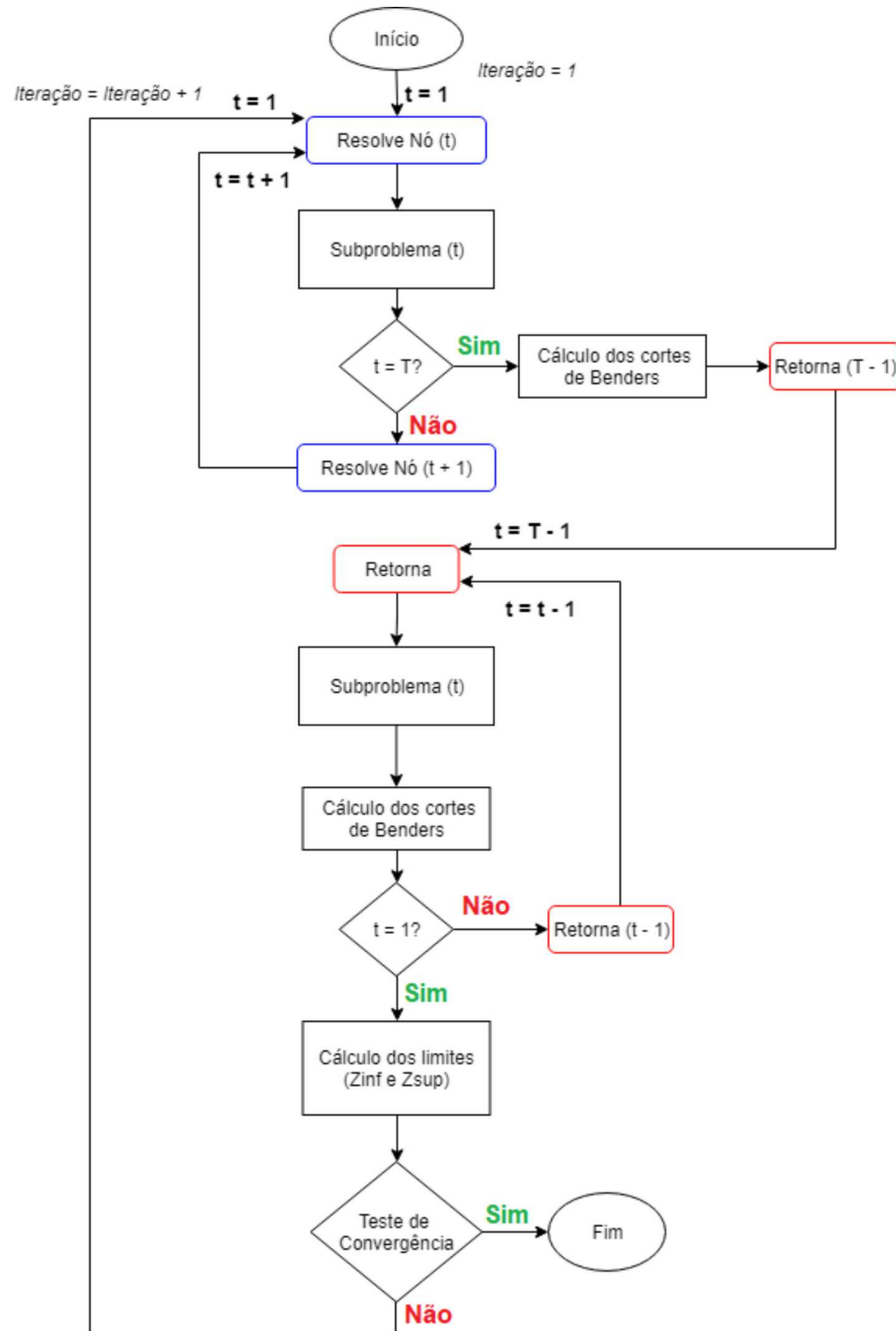
Este processo se repete até que o estágio passado para a função Resolve Nó seja igual a T . Para este caso, a resolução do subproblema será como a etapa *backward* e ao final os valores da função objetivo obtido e o valor do coeficiente de Lagrange (λ_t e w_t) necessários para a construção dos cortes de Benders de cada estágio serão armazenados. Após a construção do corte de Benders ser realizada, a função Resolve Nó irá retornar para o estágio anterior ($T-1$). No fluxograma da Figura 13, esse retorno está representado pelo bloco Retorna destacado em vermelho.

Após o retorno o subproblema para o estágio $T-1$ equivalente a sua etapa *backward* será resolvido, seus valores de λ_t e w_t serão armazenados e o cálculo do corte de Benders será realizado. Caso $T-1$ não seja o primeiro estágio, a função Resolve Nó retorna para o estágio anterior ($T-2$) e este procedimento se repete até que o estágio corrente ($T-n$) seja igual ao primeiro estágio.

Para o primeiro estágio, após a resolução de seu subproblema equivalente à etapa *backward*. Os limites (Z_{inf} e Z_{sup}) são calculados e o teste de convergência é efetuado. Caso haja convergência a metodologia termina, caso contrário uma nova iteração é iniciada. Nesta nova iteração os cortes construídos pelas etapas *backwards* da iteração anterior serão utilizados nas etapas *forwards*. Este processo iterativo irá se repetir até que ocorra convergência.

A fim de exemplificar o funcionamento da PDD recursiva, as etapas da solução de

Figura 13 – Fluxograma PDD recursiva



Fonte: Elaborada pelo autor

um problema de planejamento de um sistema hidrotérmico, como apresentado na seção 3.1, para uma árvore de cenários simples de 3 estágios e duas aberturas são apresentadas nas Figuras 14 e 15.

A Figura 14 apresenta os primeiros cinco subproblemas resolvidos pela PDD recursiva. Como pode ser observado, no 1º passo, o subproblema equivalente à etapa *forward* para o nó do primeiro estágio (nó 1) é resolvido. O valor encontrado para o volume armazenado pela usina hidrelétrica ao final deste estágio (V_{f1}) é passado para os nós do estágio seguinte (nó 2 e nó 3) para serem utilizados como volumes iniciais, neste exemplo os volumes armazenados são variáveis de decisão.

Em seguida, como pode ser visto no 2º passo, a árvore é percorrida até o segundo estágio, no qual o problema equivalente à etapa *forward* do nó 2 é resolvido e o valor encontrado para o volume armazenado pela usina hidrelétrica ao final da solução é passado para seus nós descendentes (V_{f2}). Este valor será utilizado como o volume armazenado inicial pelos subproblemas descendentes do nó 2.

A progressão continua até um nó de último estágio ser atingido (nó 4), como pode ser visto no 3º passo. Ao chegar em um problema de fim de árvore o subproblema equivalente à etapa *backward* para este nó é resolvido e a construção do corte de Benders é realizada. Este corte, Corte 4, é transmitido para seu nó ascendente, nó 2.

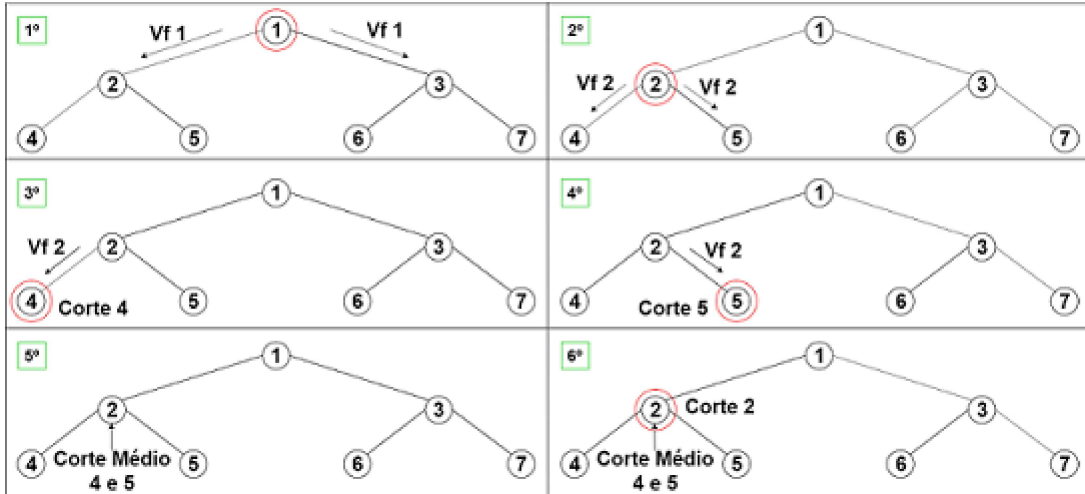
Para o caso de um problema com mais de um cenário para cada estágio, o algoritmo percorre todas as aberturas do ramo da árvore em que se encontra, 4º passo, antes de retornar para o estágio anterior. Assim, o subproblema equivalente à etapa *backward* para nó 5 é resolvido e seu corte de Benders calculado, Corte 5.

Neste trabalho, os cenários são considerados equiprováveis. Por isso, o corte médio entre os cortes construídos pelos nós 4 e 5 é calculado, 5º passo. Este corte médio (Corte Médio 4 e 5) será utilizado pelo nó do estágio anterior, nó 2, para efetuar a solução do seu subproblema equivalente à etapa *backward*, 6º passo. Após a solução deste subproblema, o corte do nó 2 (Corte 2) é construído.

A Figura 15 mostra que durante o 7º, 8º, 9º e 10º passo, o processo descrito na Figura 14 é repetido para o segundo ramo da árvore, iniciado pelo nó 3. No 11º passo, o subproblema equivalente à etapa *backward* para o primeiro estágio é resolvido considerando o corte médio recebido pelo seus nós descendentes (Corte Médio 2 e 3).

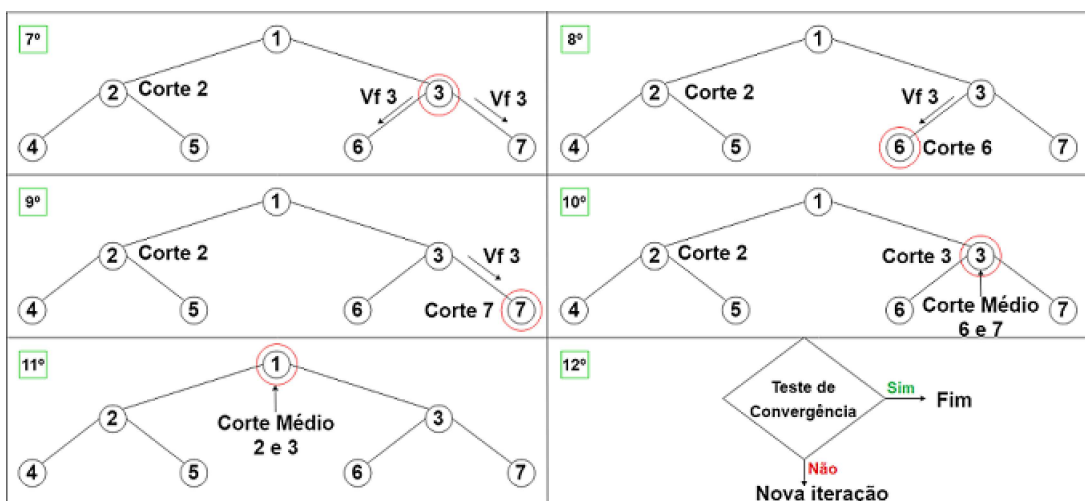
Ao final de todos esses passos que representam uma iteração do problema, o teste de convergência é realizado, 12º passo. Como existem mais de um cenário por estágio e os mesmos são considerados equiprováveis, compara-se o valor de Z_{inf} com o valor de Z_{sup} médio encontrado ao longo de toda a iteração. O valor de Z_{inf} é calculado como o custo imediato mais o custo futuro da solução do subproblema do primeiro estágio, nó 1, como apresentado em (3.22). Já o valor de Z_{sup} médio é encontrado realizando a média

Figura 14 – Processo de solução PDD recursiva (1)



Fonte: Elaborada pelo autor

Figura 15 – Processo de solução PDD recursiva (2)



Fonte: Elaborada pelo autor

entre os valores dos custos imediatos, valor da função objetivo menos o valor associado ao custo futuro, encontrados ao longo dos diferentes cenários da árvore, equação (3.23). O teste de convergência, portanto, será a diferença absoluta entre $Zinf$ e $Zsup_{med}$ calculado como mostrado na equação (3.24). Por fim, caso não haja convergência todo o processo é repetido considerando agora as restrições para os custos futuros construídas durante a primeira iteração.

$$Zinf = f_1^* + \alpha_1^* \quad (3.22)$$

$$Zsup_{med} = \sum_t^T \sum_a^{NA} \frac{f_t^{a*} - \alpha_t^{a*}}{(NA)^{t-1}} \quad (3.23)$$

$$\frac{Zsup_{med} - Zinf}{Zsup_{med}} \leq \epsilon \quad (3.24)$$

Sendo:

$Zinf$ = Limite inferior do problema

f_1^* = Valor encontrado para a função objetivo do subproblema do primeiro estágio

α_1^* = Valor estimado do custo futuro ao final da resolução do subproblema do primeiro estágio

T = Número de estágios do problema

NA = Número de aberturas da árvore de cenários do problema

$Zsup_{med}$ = Limite superior médio do problema

f_t^a = Valor encontrado para a função objetivo do subproblema do estágio t e abertura a

α_t^a = Valor estimado do custo futuro ao final da resolução do subproblema do estágio t e abertura a

ϵ = Tolerância para convergência pré-determinada

Para problemas mais complexos, nos quais o número de estágios e aberturas da árvore de cenários são maiores, o tempo computacional necessário para encontrar a solução pela PDD recursiva aumenta consideravelmente. Com o intuito de melhorar do ponto de vista computacional esta metodologia, realizou-se sua paralelização assíncrona, a qual será apresentada na subseção seguinte.

3.2.4 Programação dinâmica dual recursiva paralelizada de forma assíncrona

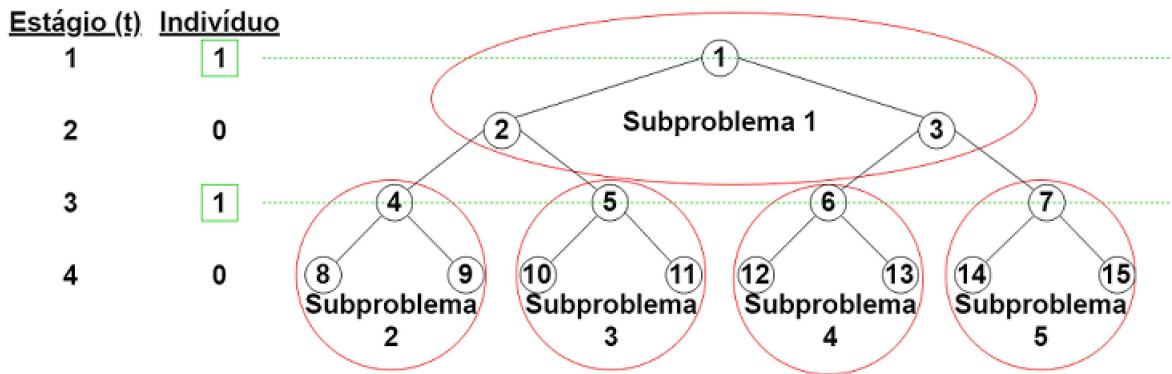
Encontra-se em [27] um modo clássico de paralelizar a PDD, o qual explora a independência dos cenários no mesmo estágio da árvore. O algoritmo PDD tradicional é aplicado mas a resolução dos problemas dos nós pertencentes ao mesmo estágio é realizada em paralelo. Como o subproblema de um nó requer, durante a etapa *forward* da solução, o estado das variáveis de estado de seu nó ascendente ou, durante a etapa *backward* da solução os cortes de Benders de seus nós descendentes, existe uma dependência entre estágios que cria um ponto de sincronização em cada estágio, esses pontos de sincronização são uma grande desvantagem deste método [16]. Além de criar uma dependência entre os estágios durante o processo *backward* e o processo *forward*, o método clássico também limita a granularidade do algoritmo, uma vez que a granularidade máxima de um estágio será dada pelo número de cenários nesse estágio. Portanto, como cada estágio tem uma granularidade diferente, espera-se a ociosidade ao longo do procedimento tradicional de solução da PDD, especialmente nos primeiros cenários da árvore de cenários.

Por esse motivo a abordagem escolhida por este trabalho foi a de paralelizar a PDD de forma assíncrona, como proposto em [20] e [16]. No algoritmo da PDD paralelizado de forma assíncrona as etapas *forward* e *backward* se misturam, e por conta disso sua iteração ocorre de forma diferente da iteração do modelo tradicional. Suas iterações são caracterizadas pela resolução independente de todos os subproblemas da árvore de cenário. A troca de informações ocorre simultaneamente ao longo da árvore entre as iterações, as variáveis de decisão são passadas para os estágios descendentes e os cortes de Benders são passados para os estágios ascendentes de cada subproblema. Desta forma, os nós são totalmente independentes entre si e podem ser resolvidos simultaneamente dentro de um passo que definirá a resolução de uma iteração, porém entre os passos há um ponto de sincronização. Como apresentado na subseção anterior, a implementação recursiva da PDD também se comporta desta maneira e, por isso, a paralelização foi implementada a partir dela.

Para realizar a divisão do problema original em subproblemas, que serão resolvidos por diferentes processadores, realizou-se o agrupamento dos nós da árvore de cenários. Como previamente abordado no capítulo 2, existem diferentes maneiras de agrupar os nós da árvore de cenários [19]. Neste trabalho é proposto a utilização de um algoritmo genético para definir qual seria o agrupamento ótimo para determinada árvore de cenários.

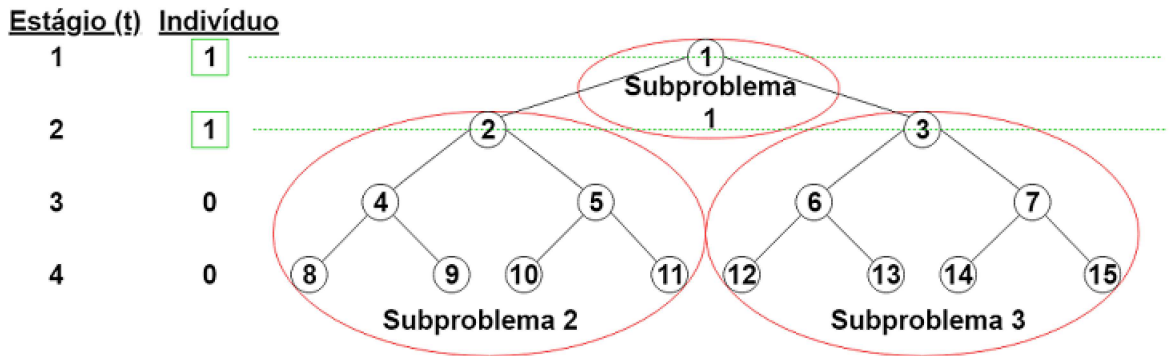
O indivíduo (variável binária com o número de bits igual ao número de estágios) é responsável por definir em quais estágios da árvore a mesma será desmembrada e ocorrerá o agrupamento dos nós para serem resolvidas como subproblemas, cada indivíduo representa um tipo de agrupamento. As Figuras 16 e 17 mostram exemplos de agrupamento para um problema de quatro estágios e duas aberturas, a variável indivíduo determina as posições de início de agrupamento adotando a informação 1 para tal. Dessa forma, o primeiro

Figura 16 – Exemplo de agrupação de nós (1)



Fonte: Elaborada pelo autor

Figura 17 – Exemplo de agrupação de nós (2)



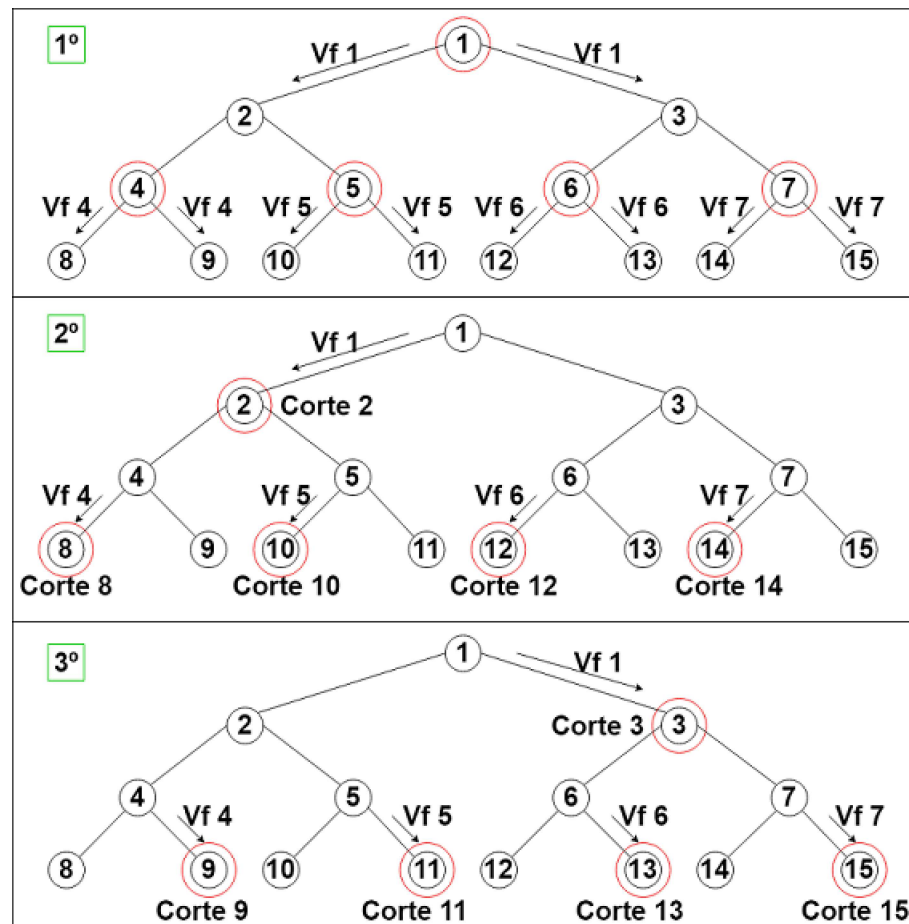
Fonte: Elaborada pelo autor

estágio sempre será igual a 1 enquanto os demais estágios podem ser 0 ou 1.

No exemplo da Figura 16, a divisão da árvore de cenários foi realizada no estágio 1 e 3, criando assim, um subproblema entre o primeiro e segundo estágio e quatro subproblemas entre o terceiro e quarto estágio. A Figura 17 apresenta um agrupamento diferente para o mesmo problema, neste caso a divisão foi realizada no estágio 1 e 2, criando assim, um subproblema no estágio 1 e dois subproblemas entre os estágios 2 e 4.

Cada subproblema será enviado a processadores diferentes (processadores escravos) para serem resolvidos simultaneamente. Além disso, um dos processadores (processador mestre) ficará por conta de realizar as sincronizações entre os subproblemas ao final de cada iteração. As Figuras 18, 19 e 20 ilustram o funcionamento do algoritmo da PDD recursiva paralelizada de forma assíncrona para o agrupamento da Figura 16. O problema é considerado como um problema de planejamento energético de sistemas hidrotérmicos, apresentado na seção 3.1, e todos os cenários são considerados equiprováveis. O funcionamento é similar ao do algoritmo PDD recursiva mostrado na subseção anterior.

Figura 18 – Funcionamento PDD paralelizada de forma assíncrona (1)

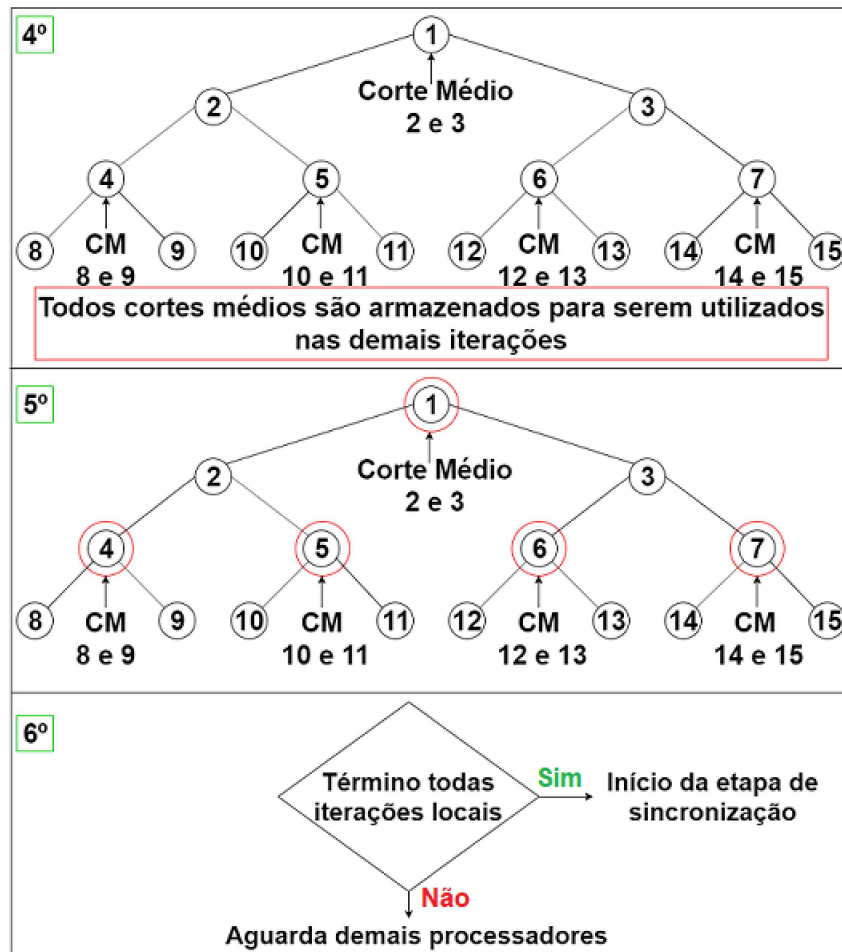


Fonte: Elaborada pelo autor

Os volumes de armazenamento iniciais das usinas hidrelétricas do sistema de estudo só é conhecido para o primeiro estágio do problema. Por isso, adota-se para a primeira iteração que todos os nós de início de agrupamento iniciarão com esses valores de volumes armazenados. Com os valores de volumes armazenados iniciais definidos, o processo de solução da metodologia é iniciado. No 1º passo, os nós 1, 4, 5, 6 e 7 serão resolvidos de forma simultânea, Figura 18. Os resultados encontrados para a variável de estado que representa o volume armazenado pela usina hidrelétrica ao final do estágio corrente (V_{f_t}) serão passados para os estágios descendentes pertencentes aos respectivos agrupamentos, destacados pela Figura 16. O nó 1 transmite o valor de V_{f_1} para os nós 2 e 3, o nó 4 transmite o valor de V_{f_4} para os nós 8 e 9, o nó 5 transmite o valor de V_{f_5} para os nós 10 e 11, o nó 6 transmite o valor de V_{f_6} para os nós 12 e 13, e o nó 7 transmite o valor de V_{f_7} para os nós 14 e 15.

Ainda na Figura 18, no 2º e 3º passo ocorre a resolução dos subproblemas dos nós descendentes de cada agrupamento (nós 2, 3, 8, 9, 10, 11, 12, 13, 14 e 15), para a resolução destes subproblemas os valores de V_{f_t} recebidos por estes nós serão utilizados

Figura 19 – Funcionamento PDD paralelizada de forma assíncrona (2)



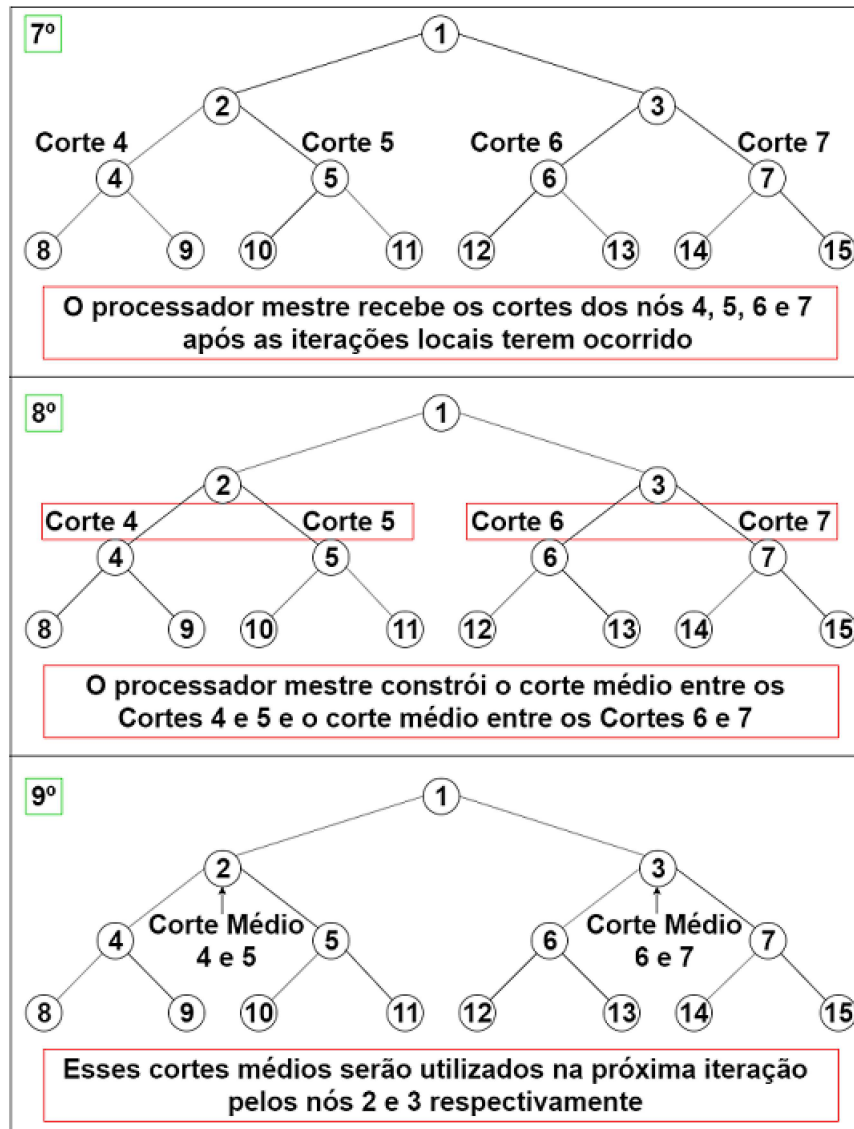
Fonte: Elaborada pelo autor

como parâmetro para o volume de armazenamento inicial. Como esses nós são todos de fim de árvore para seus respectivos agrupamentos, Figura 16, ao final da solução de seus subproblemas são construídos cortes de Benders (Corte t). Esses cortes serão transmitidos para os nós ascendentes, os Cortes 2 e 3 serão transmitidos para o nó 1, os cortes 8 e 9 para o nó 4, os cortes 10 e 11 para o nó 5, os cortes 12 e 13 para o nó 6 e os cortes 14 e 15 para o nó 7.

A Figura 19 apresenta a continuação da solução, no 4º passo os cortes médios de Benders são calculados a partir dos cortes (Cortes t) recebidos pelos nós descendentes. Com a informação adicional do corte médio calculado, o problema dos nós 1, 4, 5, 6 e 7 é resolvido mais uma vez, 5º passo. Por fim, no 6º passo, os processadores aguardam o término da solução de todos os demais. Quando todos finalizam, a etapa de sincronização se inicia.

A sincronização dos resultados obtidos é realizada pelo processador mestre. Os processadores escravos transmitem para ele as soluções encontradas nos subproblemas de

Figura 20 – Funcionamento PDD paralelizada de forma assíncrona (3)



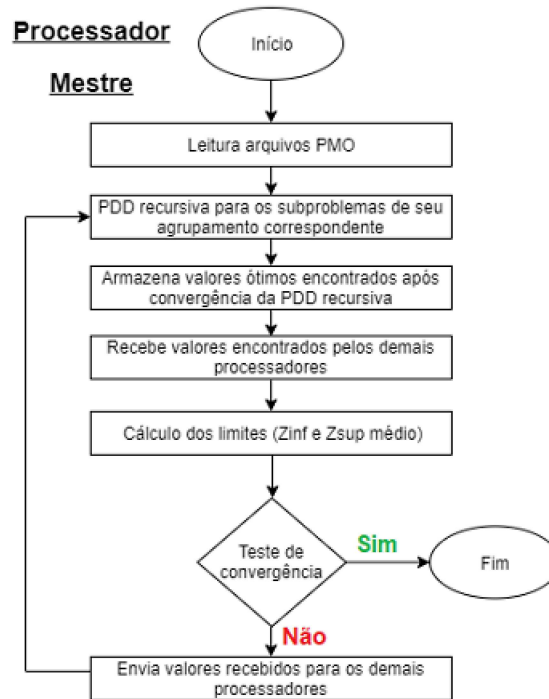
Fonte: Elaborada pelo autor

seu agrupamento, os valores das variáveis de estado, os cortes de Benders construídos e o limite superior encontrado após a iteração local.

Para os agrupamentos em que o primeiro nó não é o nó raiz da árvore de cenários, ao final da convergência, também são transmitidos os valores necessários (λ_t , w_t) para a construção de seus cortes de Benders. De posse desses valores, o nó mestre calcula os cortes médios para serem considerados nas demais iterações. Neste exemplo isto ocorre para os nós 4, 5, 6 e 7, como mostra a Figura 20.

Com todas informações coletadas, o processador mestre verifica se existe convergência através dos valores do limite inferior do problema original (Zinf) e do limite superior médio (Zsup médio, calculado com os dados coletados dos demais processadores). Caso não

Figura 21 – Fluxograma processador mestre

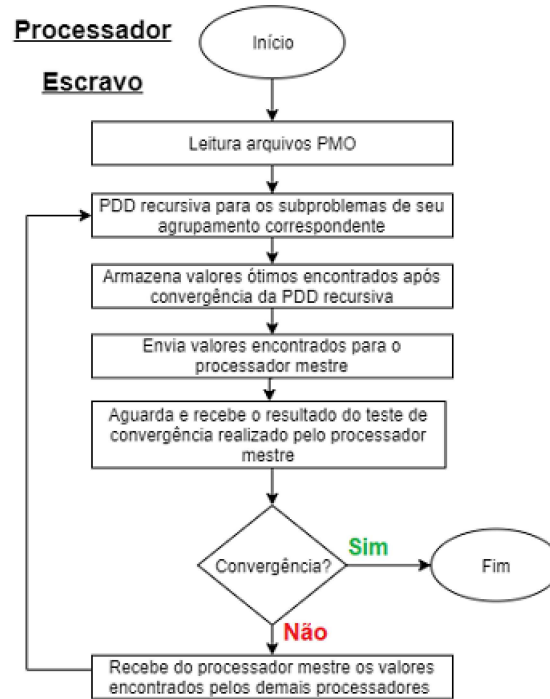


Fonte: Elaborada pelo autor

tenha ocorrido convergência, ele distribui os dados coletados e calculados para os demais processadores e uma nova iteração do problema é inicializada. Desta vez, os subproblemas são resolvidos considerando os cortes de Benders e os novos valores de volume armazenado inicial, calculados na iteração anterior. O processo é repetido até que ocorra convergência. O fluxograma do processador mestre e dos processadores escravos estão nas Figuras 21 e 22.

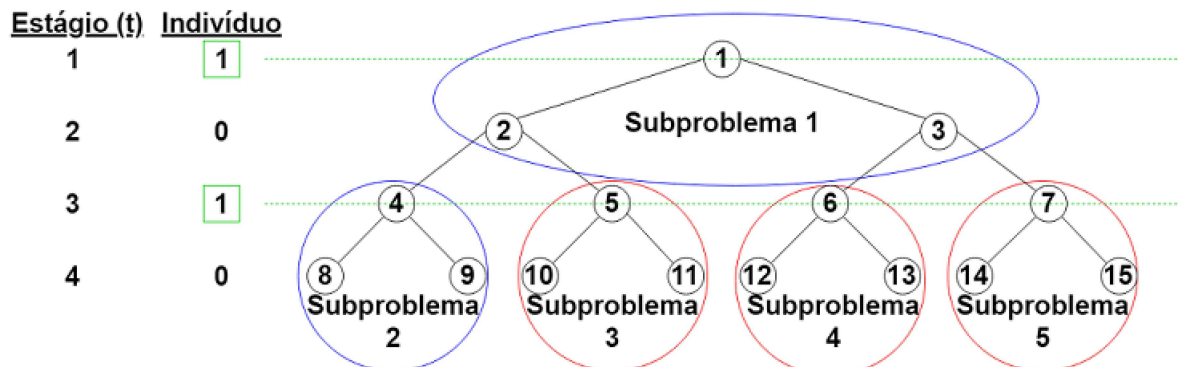
Para os casos em que o número de agrupamentos de nós definido for maior do que o número de processadores disponíveis, os agrupamentos são igualmente divididos pelos processadores. Com o intuito de diminuir a perda de tempo computacional para estes casos, a distribuição é realizada de forma hierárquica, fazendo com que os subproblemas acumulados por cada processador sejam pertencentes ao mesmo ramo da árvore e com isso possam já trocar informações entre eles durante o processo de solução local. Para o exemplo da Figura 16, caso houvessem apenas 4 processadores disponíveis para resolvê-lo, um processador ficaria com 2 subproblemas que seriam resolvidos em série e os demais ficariam com 1 problema cada. Porém, para evitar maiores perdas de tempo computacional, a distribuição dos subproblemas ocorre como mostra a Figura 23. Os subproblemas 1 e 2 destacados em azul seriam resolvidos por um mesmo processador, desta maneira durante as iterações locais os nós 4 e 2 já trocariam informações. Os demais subproblemas seriam resolvidos cada um por um processador diferente.

Figura 22 – Fluxograma processador escravo



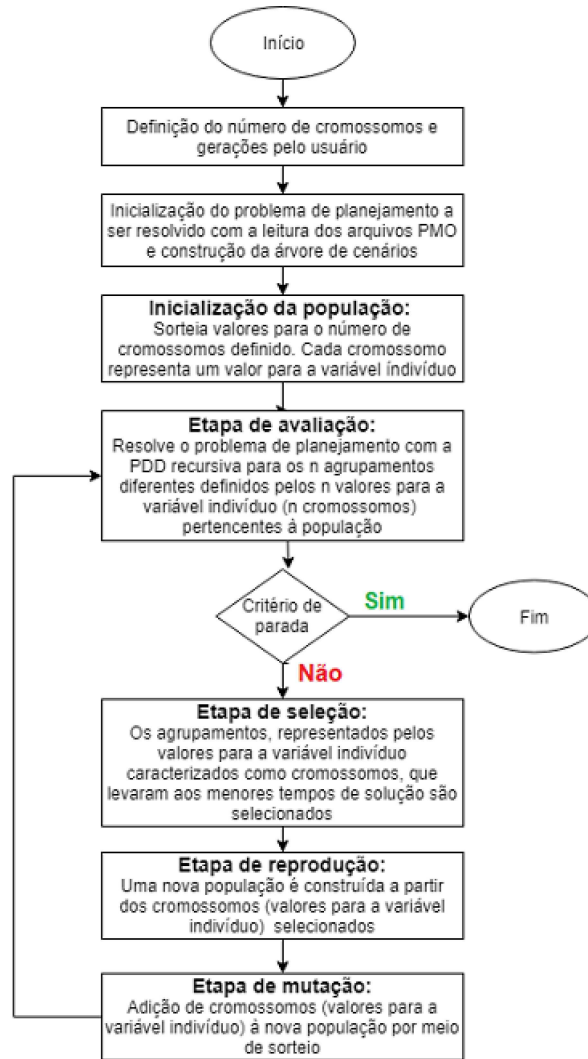
Fonte: Elaborada pelo autor

Figura 23 – Distribuição de subproblemas para um mesmo processador



Fonte: Elaborada pelo autor

Figura 24 – Fluxograma algoritmo genético



Fonte: Elaborada pelo autor

3.2.5 Agrupamento ótimo de nós

Para problemas com árvores de cenários mais complexas, maior número de estágios e aberturas, o número de maneiras de se agrupar os nós em subproblemas tende a crescer consideravelmente e buscar qual seria o melhor agrupamento de forma exaustiva, analisando todas as possibilidades, se torna algo inviável. Propõe-se, portanto, utilizar a técnica de algoritmos genéticos (AG's) para definir o melhor agrupamento dos nós da árvore de cenários para serem resolvidos como subproblemas.

A escolha pelo AG se deve à natureza binária da variável a ser otimizada, a variável indivíduo, apresentada na subseção anterior. O algoritmo otimiza a variável buscando o menor tempo de solução do problema original. A Figura 24 apresenta o fluxograma do algoritmo genético implementado.

Define-se o tamanho da população, número de cromossomos que ela terá e a quantidade de gerações que serão analisadas [38]. Após essa definição o problema de planejamento energético de sistema hidrotérmico a ser resolvido, apresentado na seção 3.1, é inicializado e sua árvore de cenários definida. Com estas informações, o algoritmo constrói populações de cromossomos por meio de sorteio, esses cromossomos representam possíveis valores para a variável indivíduo, apresentada na subseção anterior, e conseqüentemente, um respectivo agrupamento para os nós da árvore de cenários.

A avaliação destes cromossomos, ocorre através da solução do problema de planejamento energético de sistema hidrotérmico inicializado pela metodologia PDD recursiva paralelizada de forma assíncrona, apresentada na subseção anterior, para o seu respectivo agrupamento de nós. Os cromossomos (valores para a variável indivíduo) que resultem em um menor tempo de solução para o problema, são os melhores avaliados e terão maior probabilidade de serem selecionados na etapa da seleção. Essa etapa compara os cromossomos da população e seleciona os de melhor desempenho para passarem pela etapa de reprodução, responsável pela criação dos novos cromossomos da nova população.

Para evitar que a solução do problema fique presa em ótimos locais, a etapa de mutação adiciona cromossomos de forma aleatória à nova população. Todas essas etapas são repetidas com a nova população criada até que o critério de parada seja atingido. Para o caso deste trabalho, o critério de parada escolhido foi o número total de gerações ou a repetição durante cinco gerações consecutivas do valor ótimo encontrado.

Como existe a necessidade de se resolver o problema de planejamento de sistemas hidrotérmicos para um mesmo sistema várias vezes, espera-se que após a otimização realizada pelo algoritmo genético o agrupamento ótimo encontrado poderá ser repetido para as demais resoluções dele, garantindo assim a melhor distribuição de subproblemas entre os processadores e, conseqüentemente, um melhor tempo computacional.

4 RESULTADOS

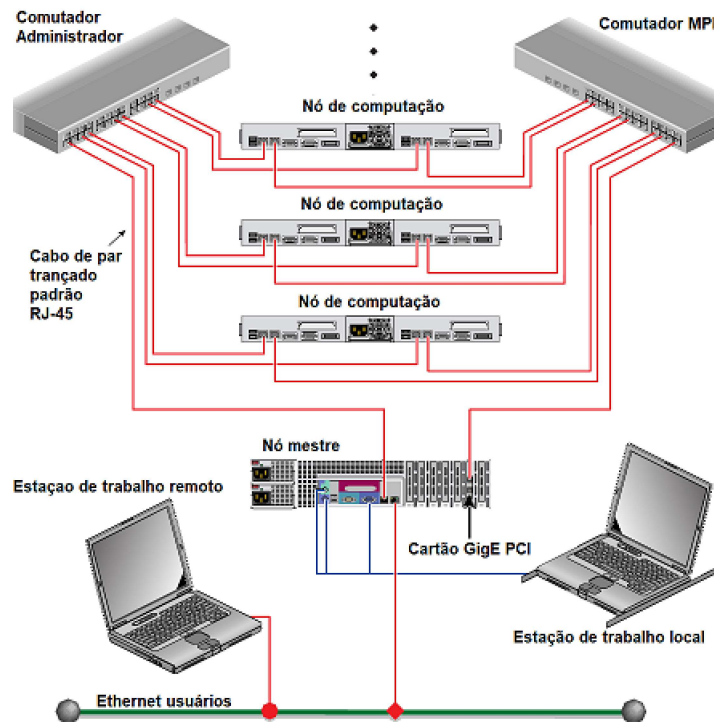
A fim de validar a metodologia proposta e avaliar os seus ganhos computacionais, este capítulo irá mostrar os resultados obtidos para diferentes sistemas simulados. Primeiramente, definiu-se um problema teste de instância mínima com uma usina hidrelétrica e duas termelétricas, no qual o usuário determina o número de estágios e de aberturas a serem consideradas para a montagem da árvore de cenários e qual usina hidrelétrica do sistema brasileiro deseja utilizar. O intuito de resolver um problema simples é o de validar o valor encontrado pela metodologia comparando-o com o encontrado pela solução do problema como um Problema Linear (PL) único equivalente. Validada a convergência da metodologia, a simulação do problema de planejamento energético para o subsistema norte do SIN é realizada, a fim de verificar o ganho computacional da metodologia em sistemas reais mais complexos.

As simulações foram todas realizadas pela Plataforma de Computação Paralela do Programa de Pós Graduação de Engenharia Elétrica da Universidade Federal de Juiz de Fora (CLUSTER-PPEE). O Cluster-PPEE possui um nó mestre com 2 Intel Quad-Core Xeon E5620 (2,40GHz, cache MB, 1066 FSB) com 8 núcleos, que permite 16 processos simultâneos, e 24 GB de memória RAM DDR3 1066MHz. Os nós de computação são de dois tipos, 10 nós com 2 Intel Quad-Core Xeon E5420 (2,50 GHz, cache 12MB, 1333 FSB) com 8 núcleos, que permite 8 processos simultâneos, e 8 GB de memória RAM DDR2 1333 MHz. E 22 nós com 2 Intel Quad-Core Xeon E5620 (2,40 GHz, cache 12 MB, 1066 FSB) com 8 núcleos, que permite 16 processos simultâneos, e 12 GB memória RAM DDR3 1066 MHz. Além disso, possui 3 computadores de Rede e 1 console para gerenciamento, a estrutura dos componentes de *hardware* do CLUSTER-PPEE podem ser vistas na Figura 25. Para as simulações foram dedicados 10 nós Intel Quad-Core Xeon E5620 (2,40 GHz, cache 12 MB, 1066FSB) com 16 núcleos, permitindo assim, até 160 processos simultâneos.

O *software* de Gerenciamento é o Perceus 1.9 que provisiona e gerencia os grupos de servidores. O TORQUE PBS 5.1 é o responsável por alocar os nós computacionais para acesso exclusivo e/ou compartilhado aos usuários por um determinado período de tempo, assim como, gerenciar a fila de submissão de trabalhos arbitrando conflitos entre os pedidos de recursos computacionais.

A implementação das metodologias foram todas realizadas em python 3 e para a paralelização fez-se uso do protocolo *Message Passing Interface* (MPI). Os resultados encontrados estão expostos nas seções seguintes, separados pelos sistemas modelados para a simulação.

Figura 25 – Hardware CLUSTER-PPEE



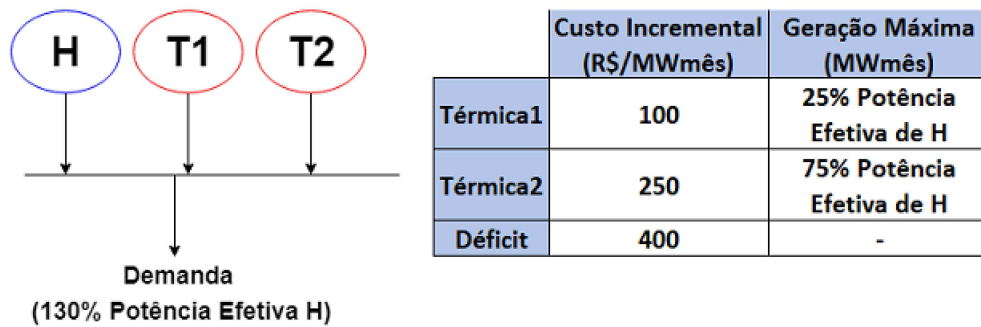
Fonte: Elaborada pelo autor

4.1 SISTEMA TUTORIAL

A partir dos dados da usina hidrelétrica escolhida obtidos nos arquivos do Programa Mensal de Operação Energética (PMO) disponível em [47], são definidos os dados das usinas termelétricas e de demanda do sistema tutorial. Para os testes realizados, as potências máximas das térmicas foram estabelecidas como sendo setenta e cinco por cento e vinte e cinco por cento da potência efetiva da hidrelétrica escolhida. Além disso a demanda do sistema é definida como sendo cento e trinta por cento a potência efetiva da usina hidrelétrica. A Figura 26 representa o sistema teste juntamente com os dados de custos incrementais das térmicas e custo de déficit considerados para o mesmo. As restrições consideradas para o problema foram as restrições de atendimento a demanda, de balanço hídrico e de canalização das variáveis de decisão do problema, mostradas na seção 3.1.

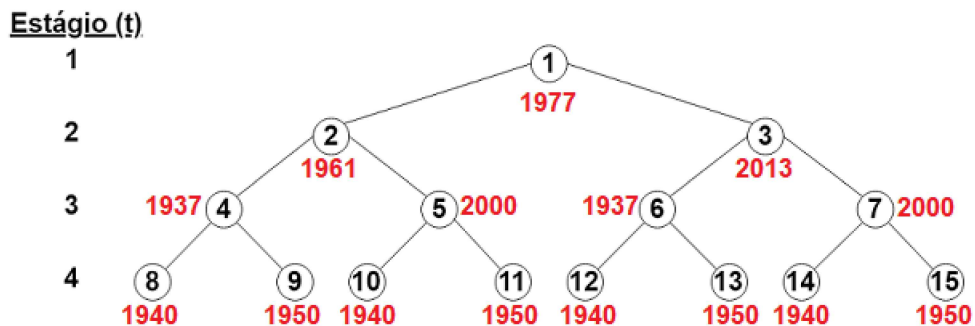
Para construir a árvore de cenários a ser resolvida utilizou-se o histórico de vazões encontrados nos arquivos do PMO. Um sorteio entre os dados históricos foi realizado para definir as afluências à usina hidrelétrica em cada nó da árvore, a Figura 27 mostra um exemplo para uma árvore de cenários de 4 estágios e 2 aberturas. Para o trabalho optou-se por definir que as afluências das aberturas de cada estágio são iguais. Para o caso da Figura 27, como são apenas duas aberturas, são sorteados apenas dois valores de afluência

Figura 26 – Sistema Tutorial



Fonte: Elaborada pelo autor

Figura 27 – Afluências sorteadas para uma árvore de cenários de 4 estágios e 2 aberturas



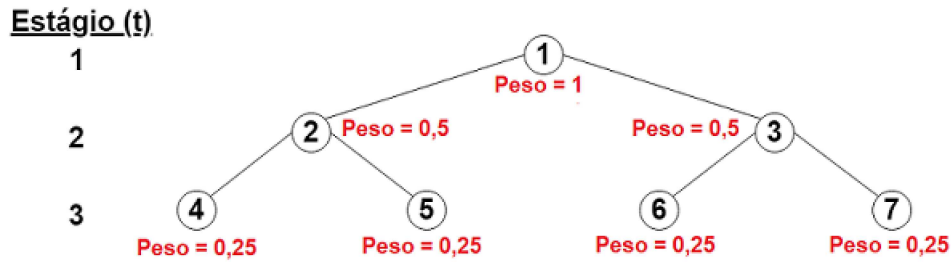
Fonte: Elaborada pelo autor

para cada estágio, vide os nós 4-6 e 5-7 bem como os nós 8-10, 9-11, 12-14 e 13-15.

Para garantir que os algoritmos da PDD recursiva e PDD recursiva paralelizada de forma assíncrona implementados convergem para o resultado real do problema, o problema foi também modelado como um PL único. Para resolver o problema de árvore completa como um PL único, todos os estágios são incluídos em um mesmo subproblema e suas restrições de balanço hídrico, atendimento a demanda e de canalização são adotadas, considerando pesos referentes às probabilidades de ocorrência de cada possível cenário da árvore.

Para exemplificar a formulação equivalente ao PL único do problema definido pelo sistema tutorial, considera-se uma árvore simples com três estágios e duas aberturas com nós equiprováveis, Figura 28. O equacionamento 4.1, apresenta a função objetivo e restrições resultantes do PL único para este exemplo. Destaca-se que quanto maior a dimensão da árvore de cenários, a resolução do problema por este tipo de formulação se torna computacionalmente ineficiente ou até mesmo inviável, o que reforça a necessidade da utilização de metodologias de decomposição para a solução de problemas de maior

Figura 28 – Pesos para uma árvore de cenários de 3 estágios e 2 aberturas com nós equiprováveis



Fonte: Elaborada pelo autor

dimensão.

$$\begin{aligned} \min \quad & c_{gT1} \times gT1_1 + c_{gT2} \times gT2_1 + c_{defST} \times def_1 + 0,5[(c_{gT1} \times gT1_2^1 + c_{gT2} \times gT2_2^1 + \\ & c_{defST} \times def_2^1) + (c_{gT1} \times gT1_2^2 + c_{gT2} \times gT2_2^2 + c_{defST} \times def_2^2)] + 0,25[(c_{gT1} \times gT1_3^1 + c_{gT2} \times \\ & gT2_3^1 + c_{defST} \times def_3^1) + (c_{gT1} \times gT1_3^2 + c_{gT2} \times gT2_3^2 + c_{defST} \times def_3^2) + (c_{gT1} \times gT1_3^3 + c_{gT2} \times \\ & gT2_3^3 + c_{defST} \times def_3^3) + (c_{gT1} \times gT1_3^4 + c_{gT2} \times gT2_3^4 + c_{defST} \times def_3^4)] \end{aligned}$$

(4.1)

Sujeito a:

$$gT1_a^e + gT2_a^e + gH_a^e + def_a^e = DEM_{STa}^e$$

$$vaH_a^{e+1} = vaH_a^e + K_{conv}(VAF L_a^e + vtH_a^e + vvH_a^e)$$

$$0 \leq gT1_a^e \leq GT1_{max}$$

$$0 \leq gT2_a^e \leq GT2_{max}$$

$$def_a^e \geq 0$$

$$VAH_{min} \leq vaH_a^{e+1} \leq VAH_{max}$$

$$0 \leq vtH_a^e \leq VTH_{max}$$

$$0 \leq vvH_a^e \leq VVH_{max}$$

Sendo:

e = Número do estágio

a = Número da abertura

c_{gT1} = Custo da geração da usina térmica 1 do sistema tutorial

c_{gT2} = Custo da geração da usina térmica 2 do sistema tutorial

c_{defST} = Custo do déficit do sistema tutorial

$gT1_a^e$ = Variáveis de decisão da geração da usina térmica 1 do sistema tutorial para a abertura a do estágio e

$gT2_a^e$ = Variáveis de decisão da geração da usina térmica 2 do sistema tutorial para a abertura a do estágio e

gH_a^e = Variáveis de decisão da geração da usina hidrelétrica do sistema tutorial para a abertura a do estágio e

def_{STa}^e = Variáveis de decisão do déficit do sistema tutorial para a abertura a do estágio e

DEM_{STa}^e = Valor da demanda do sistema tutorial para a abertura a do estágio e

vaH_a^{e+1} = Variáveis de decisão do volume armazenado da hidrelétrica do sistema tutorial para a abertura a ao final do estágio e

vaH_a^e = Variáveis de decisão do volume armazenado da hidrelétrica do sistema tutorial para a abertura a do estágio e

K_{conv} = Constante de conversão de unidades, de m^3/s para hm^3/s , de valor igual a 2,592

$VAF L_a^e$ = Valor da vazão afluyente à usina hidrelétrica do sistema tutorial para a abertura a do estágio e

vtH_a^e = Variáveis de decisão do volume turbinado da hidrelétrica do sistema tutorial para a abertura a do estágio e

vvH_a^e = Variáveis de decisão do volume vertido da hidrelétrica do sistema tutorial para a abertura a do estágio e

$GT1_{max}$ = Capacidade máxima de geração da usina térmica 1 do sistema tutorial

$GT2_{max}$ = Capacidade máxima de geração da usina térmica 2 do sistema tutorial

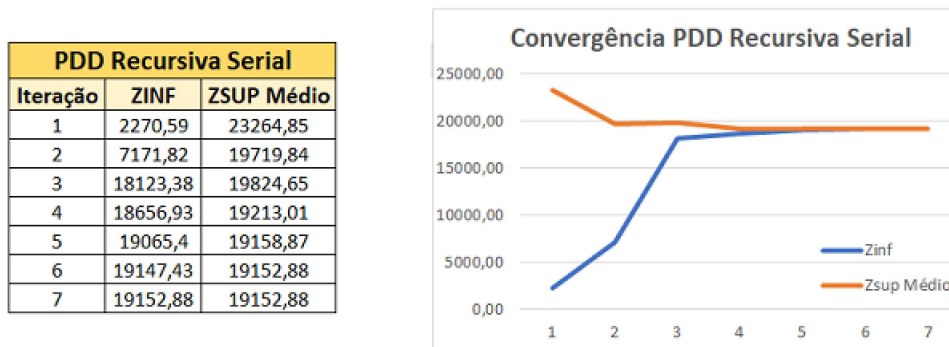
VAH_{min} = Volume de armazenamento mínimo da usina hidrelétrica do sistema tutorial

VAH_{max} = Volume de armazenamento máximo da usina hidrelétrica do sistema tutorial

VTH_{max} = Vazão máxima que pode ser turbinada pela usina hidrelétrica do sistema tutorial

VVH_{max} = Vazão máxima que pode ser vertida pela usina hidrelétrica do sistema tutorial

Figura 29 – Convergência PDD recursiva serial sistema tutorial

Sistema Tutorial - Árvore de Cenários com 6 estágios e 4 aberturas

Fonte: Elaborada pelo autor

A subseção seguinte mostra os resultados encontrados pela comparação das metodologias PDD recursiva serial e PDD recursiva paralelizada de forma assíncrona com o PL único equivalente do problema definido pelo sistema tutorial, Figura 26. O problema foi resolvido para uma árvore de cenários com seis estágios e quatro aberturas e uma árvore de cenários com doze estágios e duas aberturas. Além disso, a validação do algoritmo genético como otimizador para encontrar o melhor agrupamento de nós para a árvore de cenários também é apresentada.

4.1.1 Resultados sistema tutorial

Primeiramente o problema foi simulado para a usina de Camargos com um horizonte de tempo de seis estágios com um árvore de cenários com quatro aberturas. A PDD implementada de forma recursiva sem paralelização apresentou convergência como esperado, a Figura 29 apresenta a solução encontrada.

A metodologia PDD recursiva paralelizada de forma assíncrona também encontrou convergência para o mesmo problema apresentado anteriormente, como pode ser visto na Figura 30. Além de alcançar a convergência, ela obteve um tempo de solução menor do que metodologia PDD recursiva serial, apresentando um ganho computacional de 47,25%, Figura 31.

Outra importante constatação é a de que, como esperado, ambas metodologias convergiram para o valor encontrado pelo PL único equivalente do problema, validando as soluções encontradas e as metodologias implementadas. O mesmo sistema foi simulado para uma árvore de doze estágios com duas aberturas e os resultados de convergência das metodologias bem como as comparações dos tempos computacionais para esta configuração estão na Figura 32 e Figura 33.

Percebe-se que mesmo para árvores maiores ambas metodologias convergiram

Figura 30 – Convergência PDD recursiva paralelizada de forma assíncrona sistema tutorial

Sistema Tutorial - Árvore de Cenários com 6 estágios e 4 aberturas

PDD Recursiva Paralela Assíncrona		
Iteração	ZINF	ZSUP Médio
1	2270,59	27362,37
2	7151,19	23484,86
3	7611,86	22707,54
4	16989,82	23416,18
5	17491,49	21767,88
6	18082,98	21104,07
7	19121,95	19987,23
8	19152,88	19017,48
9	19152,88	19152,88



Fonte: Elaborada pelo autor

Figura 31 – Comparação dos tempos computacionais para sistema tutorial(1)

Sistema Tutorial - Árvore de Cenários com 6 estágios e 4 aberturas

Metodologia	Tempo (s)	Solução Encontrada (\$)
PL Único Equivalente	118,89	19152,88
PDD Recursiva Serial	48,61	19152,88
PDD Recursiva Paralela Assíncrona	25,64	19152,88

Ganho Computacional (%)	
PDD Recursiva Paralela Assíncrona x PL Único Equivalente	PDD Recursiva Paralela Assíncrona x PDD Recursiva
78,43%	47,25%



Fonte: Elaborada pelo autor

para o mesmo valor do PL único equivalente e tiveram um ganho considerável de tempo quando comparadas a ela. Observa-se também que a PDD recursiva paralelizada de forma assíncrona continua apresentando um ganho computacional com relação à PDD recursiva serial, neste exemplo obteve um ganho de 74,58%.

Para evidenciar a necessidade de se escolher o agrupamento de nós correto para o problema, a Figura 34 mostra os resultados computacionais e número de iterações até a convergência encontrados para diferentes agrupamentos utilizados na metodologia PDD recursiva paralelizada assíncrona para a resolução do problema do sistema tutorial para a árvore de cenários com seis estágios e quatro aberturas e árvore de cenários de doze estágios e duas aberturas.

A Figura 34 destaca em vermelho os agrupamentos ótimos encontrados pelo algoritmo genético para cada um dos problemas. Percebe-se que existem agrupamentos que resultam em tempos de solução piores do que a metodologia PDD recursiva sérial

Figura 32 – Convergência das metodologias para sistema tutorial

Sistema Tutorial - Árvore de Cenários com 12 estágios e 2 aberturas

PDD Recursiva Serial		
Iteração	ZINF	ZSUP Médio
1	2270,59	111956,68
2	8205,75	56962,99
3	52796,86	58472,27
4	53040,78	53499,88
5	52074,37	53107,13
6	53074,37	53086,63
7	53074,37	53074,37



PDD Recursiva Paralela Assíncrona		
Iteração	ZINF	ZSUP Médio
1	2270,58	121466,22
2	21917,94	116633,74
3	28877,17	65108,68
4	29165,81	55166,19
5	52796,86	54703,83
6	53040,78	53681,82
7	53074,37	53074,37



Fonte: Elaborada pelo autor

Figura 33 – Comparação dos tempos computacionais para sistema tutorial(2)

Sistema Tutorial - Árvore de Cenários com 12 estágios e 2 aberturas

Metodologias	Tempo (s)	Solução Encontrada (\$)
PL Único Equivalente	241,92	53074,37
PDD Recursiva Serial	144,52	53074,37
PDD Recursiva Paralela Assíncrona	36,74	53074,37

Ganho Computacional (%)	
PDD Recursiva Paralela Assíncrona x PL Único Equivalente	PDD Recursiva Paralela Assíncrona x PDD Recursiva
84,81%	74,58%



Fonte: Elaborada pelo autor

Figura 34 – Comparação dos tempos computacionais para diferentes agrupamentos

Diferentes agrupamentos para o Sistema Tutorial com árvore de cenários de 6 estágios e 4 aberturas		
Variável Indivíduo Adotada	Tempo de Execução (s)	Iterações até Convergência
[1, 0, 1, 0, 1, 0]	25,64	9
[1, 1, 0, 1, 0, 0]	30,07	10
[1, 0, 1, 0, 0, 0]	30,27	10
[1, 1, 1, 1, 0, 0]	40,33	14
[1, 0, 0, 1, 1, 1]	55,38	9

Diferentes agrupamentos para o Sistema Tutorial com árvore de cenários de 12 estágios e 2 aberturas		
Variável Indivíduo Adotada	Tempo de Execução (s)	Iterações até Convergência
[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	36,74	7
[1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0]	57,56	6
[1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]	58,17	6
[1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0]	62,31	10
[1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0]	82,18	12

Fonte: Elaborada pelo autor

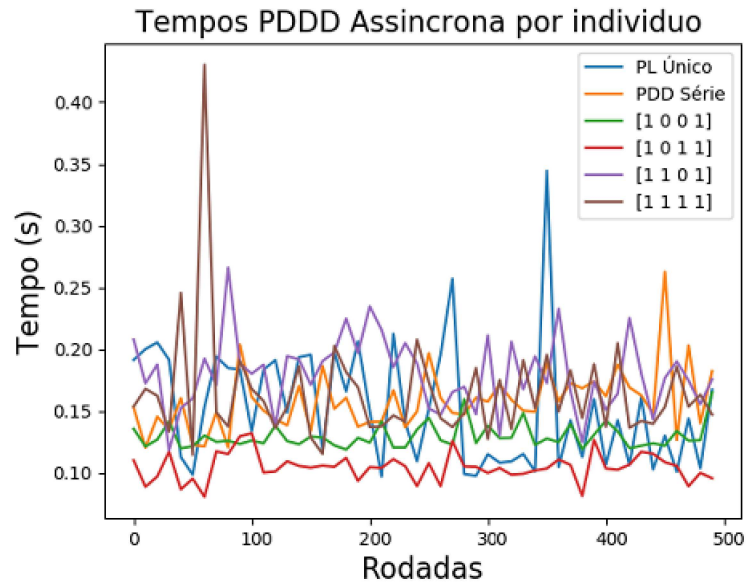
e alguns, que apesar do ganho computacional em relação à mesma, possuem um ganho inferior ao tempo de solução do agrupamento ótimo encontrado. Por isso, a escolha de como agrupar os nós é fundamental para que a metodologia PDD recursiva paralelizada de forma assíncrona encontre ganhos computacionais relevantes.

Para testar a acurácia do algoritmo genético implementado, o mesmo problema descrito nesta seção foi resolvido para uma árvore de cenários com quatro estágios e duas aberturas. Utilizou-se uma árvore mais simples para que todos os problemas resultantes dos diferentes agrupamentos possíveis de nós fossem resolvidos. Assim, espera-se verificar a eficácia do algoritmo genético em encontrar o agrupamento que resulta na solução do problema com um menor tempo computacional.

O problema foi resolvido quinhentas vezes para a mesma configuração, porém com valores de aflúências à usina hidrelétrica diferentes. A Figura 35 mostra os valores de tempo de solução de cada metodologia, PL único, PDD recursiva serial e todas as possibilidades de agrupamento de nó para a PDD recursiva paralela assíncrona. Os seus diferentes agrupamentos estão representados através da sua variável indivíduo correspondente.

Ainda na Figura 35, em destaque vermelho está o indivíduo encontrado pelo algoritmo genético como sendo o ótimo para cada execução, em 93,8% dos casos o agrupamento encontrado pelo algoritmo genético foi o de menor tempo na rodada, validando assim sua utilização para definir o agrupamento de subproblemas a ser utilizado pela metodologia PDD paralela assíncrona.

Figura 35 – Avaliação Algoritmo Genético



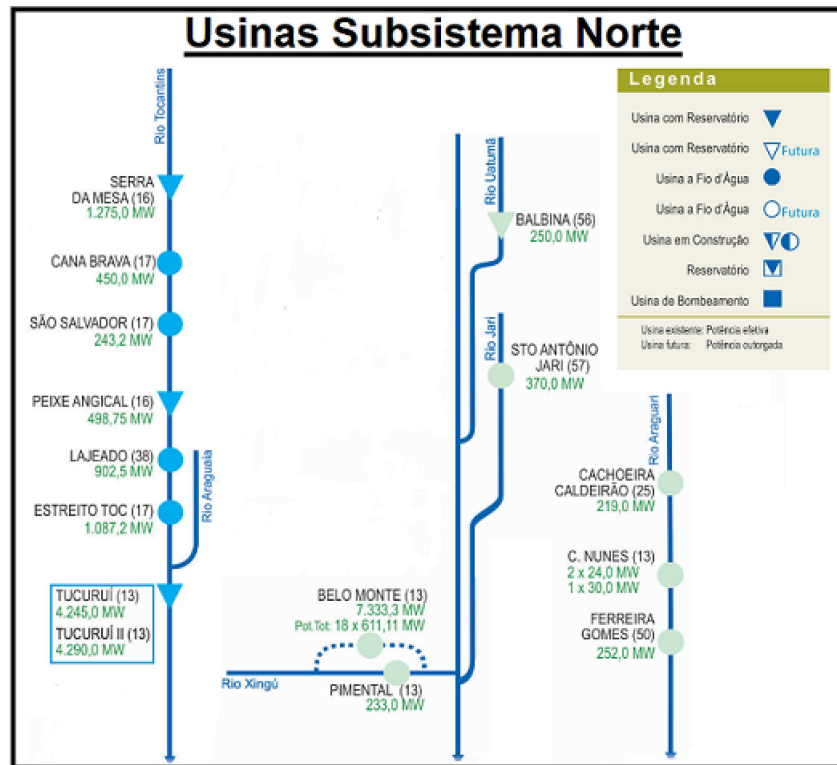
4.2 SUBSISTEMA NORTE

Para avaliar os resultados das metodologias para sistemas mais próximos de um sistema real, considerou-se o subsistema norte do Brasil como um sistema isolado. O problema de planejamento energético do subsistema foi modelado como descrito no capítulo 3. Como o subsistema foi considerado isolado, não foram considerados valores de intercâmbio com os demais subsistemas do Sistema Interligado Nacional brasileiro (SIN). As informações das usinas bem como do mercado foram retiradas da leitura dos arquivos do PMO para o mês de Junho de 2019. Como descrito na subseção 3.1.5, uma aproximação linear por partes para a Função de Produção Hidráulica (FPH), como em [44], foi utilizada. Os coeficientes necessários para a construção dessas funções foram previamente calculados e apenas inseridas na modelagem do problema.

As usinas hidroelétricas do subsistema norte podem ser vistas abaixo na Figura 36. Além delas o subsistema possui 22 térmicas com uma potência instalada total de 3884 MW, Figura 37. Para definir as afluências de cada nó da árvore de cenários o mesmo procedimento descrito na seção anterior e exemplificado pela Figura 27 foi utilizado.

A próxima subseção mostra os resultados encontrados para a metodologia PDD recursiva e PDD recursiva paralelizada de forma assíncrona para o subsistema norte considerando uma árvore de cenários com seis estágios e quatro aberturas e uma árvore de cenários com doze estágios e duas aberturas.

Figura 36 – Usinas Hidrelétricas Subsistema Norte



Fonte: Elaborada pelo autor

Figura 37 – Usinas Térmicas Subsistema Norte

Usina Térmica - Subsistema Norte	Potência Instalada (MW)
APARECIDA	166
C, ROCHA	85
FLORES LT1	40
FLORES LT2	40
GERAMAR I	166
GERAMAR II	166
IRANDUBA	25
JARAQUI	75
MANAUARA	67
MARANHAO III	519
MARANHAO IV	338
MARANHAO V	338
MAUA 3	591
MAUA B3	110
MAUA B4	150
N, VENECIA 2	178
O, CANOAS 1	0
PARNAIBA IV	56
PONTA NEGRA	66
PORTO ITAQUI	360
SUZANO MA	255
TAMBAQUI	93

Fonte: Elaborada pelo autor

Figura 38 – Convergência PDD recursiva serial subsistema Norte

Subsistema Norte - Árvore de Cenários com 6 estágios e 4 aberturas

PDD Recursiva Serial		
Iteração	ZINF	ZSUP Médio
1	16,15	64417,85
2	29,62	382,51
3	32,35	57,66
4	36,12	49,33
5	49,08	53,27
6	49,08	49,79
7	49,08	49,5
8	49,08	49,17
9	49,08	49,1
10	49,08	49,09
11	49,08	49,08

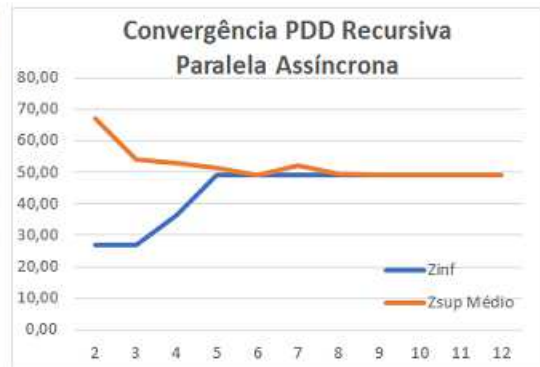


Fonte: Elaborada pelo autor

Figura 39 – Convergência PDD recursiva paralelizada de forma assíncrona susistema Norte

Subsistema Norte - Árvore de Cenários com 6 estágios e 4 aberturas

PDD Recursiva Paralela Assíncrona		
Iteração	ZINF	ZSUP Médio
1	16,15	45804,00
2	26,89	67,00
3	26,89	53,94
4	36,31	52,86
5	49,08	51,35
6	49,08	49,1
7	49,08	52,09
8	49,08	49,30
9	49,08	49,09
10	49,08	49,09
11	49,08	49,09
12	49,08	49,08



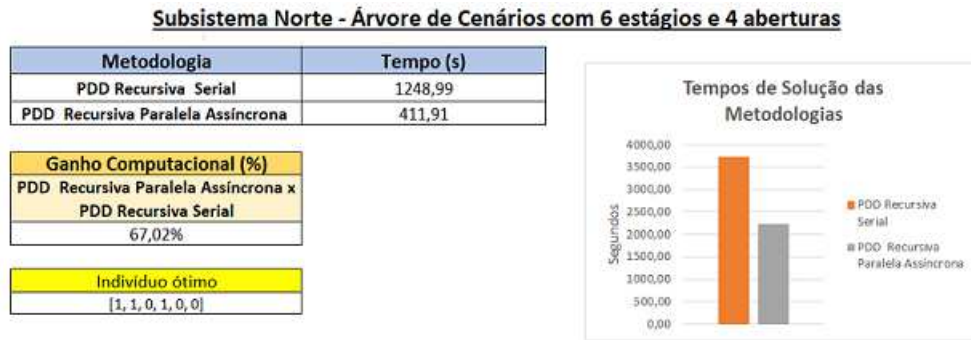
Fonte: Elaborada pelo autor

4.2.1 Resultados subsistema norte

Primeiramente adotou-se um horizonte de tempo de seis estágios com uma árvore de cenários com quatro aberturas. Para este problema ambas as metodologias convergiram como esperado. Os resultados podem ser vistos nas Figuras 38 e 39. O gráfico da Figura 38 está omitindo os valores das duas primeiras iterações e o gráfico da Figura 39 omitindo o valor da primeira iteração, para permitir uma melhor visualização da convergência da metodologia.

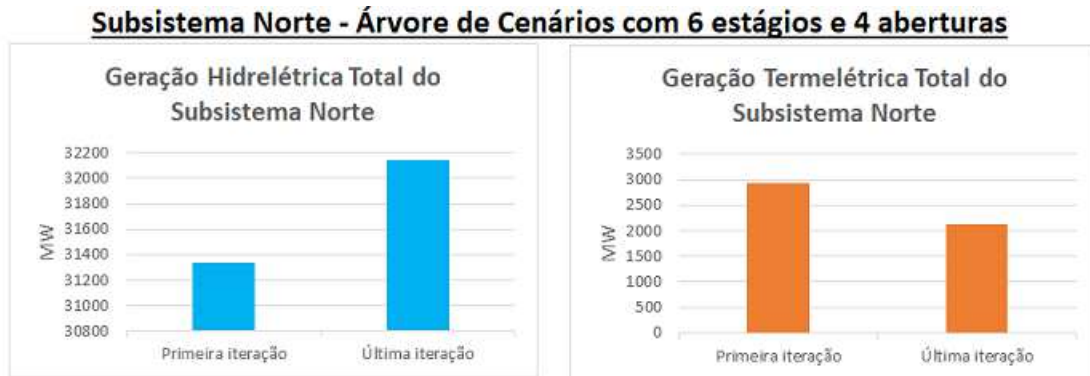
Além de alcançar a convergência, a metodologia PDD recursiva paralelizada de

Figura 40 – Comparação dos tempos computacionais para subsistema Norte(1)



Fonte: Elaborada pelo autor

Figura 41 – Geração Hidrelétrica e Termelétrica Total do Subsistema



Fonte: Elaborada pelo autor

forma assíncrona obteve um tempo de solução menor, apresentando um ganho computacional de 67,02%, Figura 40. O agrupamento utilizado pela PDD paralelizada de forma assíncrona para este exemplo, foi o encontrado como ótimo pelo algoritmo genético que está representado na forma da variável indivíduo na Figura 40.

Para verificar a solução após a convergência, acompanhou-se os valores encontrados pela PDD recursiva durante o processo iterativo, as gerações hidrelétricas e térmicas totais para a primeira e última iteração estão na Figura 41.

Como esperado a geração hidrelétrica após a convergência é maior do que a encontrada após a primeira iteração e a geração termelétrica após a convergência é menor do que a encontrada após a primeira iteração. Além disso, pode-se ver pela Figura 42 que o volume de água total armazenado ao final do primeiro estágio é maior ao final do processo iterativo quando comparado ao valor dele para a primeira iteração, o que evidencia que a metodologia tende a armazenar água para os estágios futuros a fim de baratear o custo total da operação.

Figura 42 – Armazenamento de água no primeiro estágio

Subsistema Norte - Árvore de Cenários com 6 estágios e 4 aberturas

Fonte: Elaborada pelo autor

O mesmo subsistema foi simulado para uma árvore com doze estágios duas aberturas e os resultados de convergência das metodologias bem como as comparações dos tempos computacionais para esta configuração estão na Figura 43 e Figura 44. O agrupamento utilizado pela PDD paralelizada de forma assíncrona para este exemplo, foi o encontrado como ótimo pelo algoritmo genético que está representado na forma da variável indivíduo na Figura 44.

Mesmo para uma árvore de cenários maior, ambas metodologias convergiram, os valores para as duas primeiras iterações estão omitidos nos gráficos da Figura 43 para uma melhor visualização da convergência das metodologias. Observa-se na Figura 44 que a PDD recursiva paralelizada de forma assíncrona continua apresentando um ganho computacional com relação à PDD recursiva, neste exemplo obteve um ganho de 83,40%.

Por fim, realizou-se um *speed up test* para as duas soluções de agrupamento ótimo encontradas. Primeiramente para a árvore de cenários com 6 estágios e 4 aberturas, adotou-se o agrupamento ótimo definido pela variável indivíduo $[1,1,0,1,0,0]$ e o problema foi resolvido através da PDD paralelizada de forma assíncrona com o número de processos simultâneos disponíveis no Cluster variando de 1 a 160. O resultado encontra-se na figura 45.

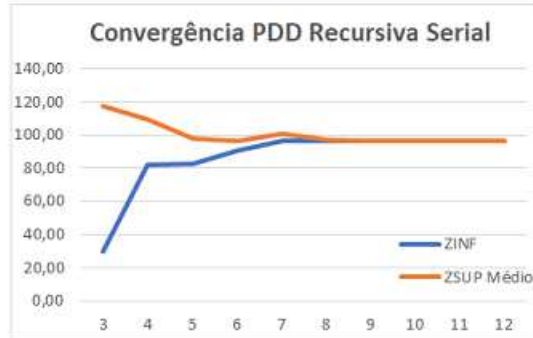
Nota-se que o menor tempo encontrado foi o de 411,91 segundos utilizando 69 processos simultâneos. Resultado esperado uma vez que o agrupamento ótimo encontrado divide o problema em 69 agrupamentos distintos.

Na figura 46 encontra-se o *speed up test* para a árvore de cenários com 12 estágios e 2 aberturas, adotou-se o agrupamento ótimo definido pela variável indivíduo $[1,1,0,0,1,0,0,0,0,0,0,0]$ e o problema através da PDD paralelizada de forma assíncrona com o número de processos simultâneos disponíveis no Cluster variando de 1 a 160.

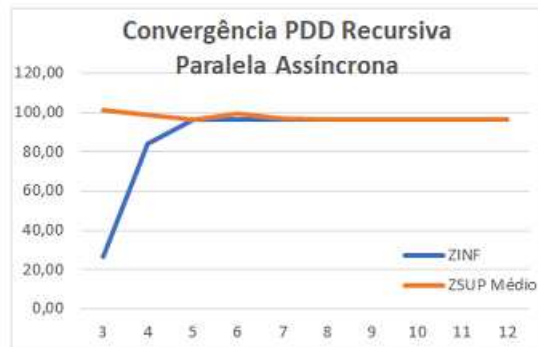
Figura 43 – Convergência das metodologias para subsistema Norte

Subsistema Norte - Árvore de Cenários com 12 estágios e 2 aberturas

PDD Recursiva Serial		
Iteração	ZINF	ZSUP Médio
1	16,15	64465,12
2	29,62	430,02
3	29,62	117,26
4	82,09	109,42
5	82,7	97,62
6	91	96,56
7	96,46	100,63
8	96,46	96,96
9	96,46	96,63
10	96,46	96,51
11	96,46	96,47
12	96,46	96,46

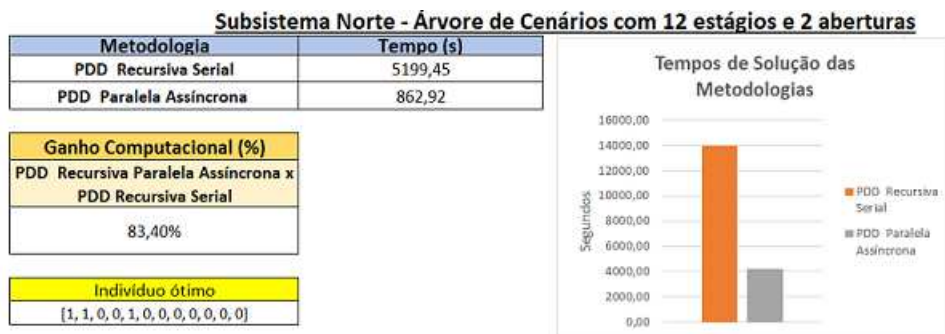


PDD Recursiva Paralela Assíncrona		
Iteração	ZINF	ZSUP Médio
1	16,15	18907,39
2	26,89	23770,48
3	26,63	101,49
4	84,18	99,06
5	96,46	96,57
6	96,46	99,41
7	96,46	96,84
8	96,46	96,66
9	96,46	96,54
10	96,46	96,49
11	96,46	96,49
12	96,46	96,46



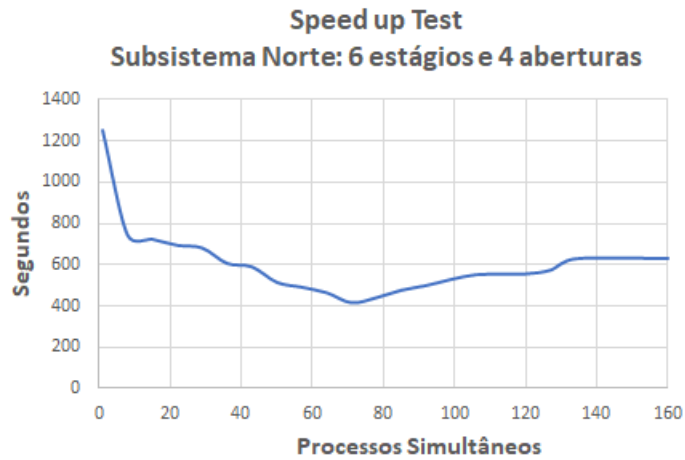
Fonte: Elaborada pelo autor

Figura 44 – Comparação dos tempos computacionais para subsistema Norte(2)



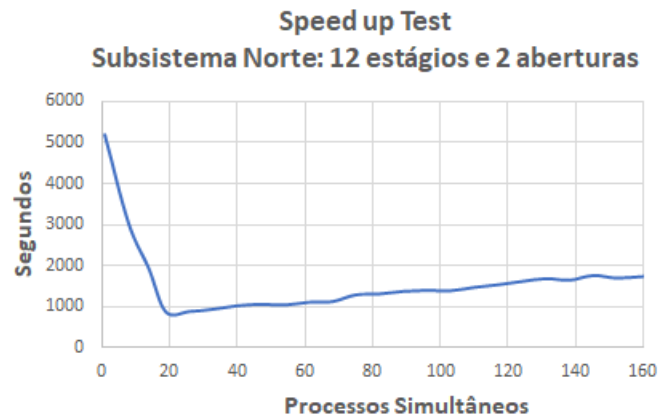
Fonte: Elaborada pelo autor

Figura 45 – Speed up Test Subsistema Norte(1)



Fonte: Elaborada pelo autor

Figura 46 – Speed up Test Subsistema Norte(2)



Fonte: Elaborada pelo autor

Nota-se que o menor tempo encontrado foi o de 862,92 segundos utilizando 19 processos simultâneos. Resultado esperado uma vez que o agrupamento ótimo encontrado divide o problema em 19 agrupamentos distintos.

5 CONCLUSÃO E TRABALHOS FUTUROS

5.1 Conclusões

Devido ao acoplamento temporal do planejamento da operação do Sistema Elétrico Brasileiro, as decisões tomadas no presente impactam diretamente nos custos operativos de um estágio futuro. Então, é de fundamental importância que as decisões operativas sejam tomadas visando um período futuro, para que se tenha o mínimo custo operativo que atenda a demanda do mercado. Contudo, a decisão operativa ótima não é simples de ser encontrada, uma vez que as afluições às hidrelétricas no futuro são estocásticas e estas exercem influência direta sobre a geração hidrelétrica, cujo custo de operação é o mais barato.

Para a solução deste problema várias metodologias podem ser aplicadas, entre elas a Programação Dinâmica Dual (PDD), porém para problemas de grande porte, como o problema de planejamento da operação do Sistema Interligado Nacional brasileiro (SIN), esta metodologia pode demandar um esforço computacional elevado. Com a motivação de obter ganhos computacionais para a solução desse tipo de problema e dando continuidade ao realizado pelo trabalho [16], o trabalho realizou a implementação da PDD de forma recursiva juntamente com a paralelização assíncrona dela. Para aprimorar a paralelização assíncrona da PDD apresentada em [16] utilizou-se um algoritmo genético para auxiliar na decisão de como agrupar os subproblemas do problema original a serem distribuídos para os diferentes processadores. Desta forma, a metodologia proposta é uma PDD recursiva paralelizada de forma assíncrona juntamente com o agrupamento ótimo dos nós da árvore de cenários.

Os resultados mostram que a metodologia proposta é eficaz para encontrar o ponto ótimo de operação dos sistemas e pode-se verificar que ela apresenta significativa redução no tempo computacional necessário para atingir convergência do problema ao ser comparada com a metodologia PDD recursiva implementada de forma serial. Dos resultados verifica-se também que a escolha do agrupamento dos nós é fundamental para esse ganho computacional e o algoritmo genético implementado é capaz de realizar a otimização desta escolha de forma satisfatória.

5.2 Trabalhos futuros

Como trabalho futuro, propõe-se o aprimoramento no algoritmo proposto de forma que o mesmo seja capaz de explorar todas as possíveis decomposições da árvore de cenários do problema, decomposição por nó, decomposição por cenário, por subárvore, e por cenário completo, como apresentado em [19]. Também sugere-se adotar outras estratégias de melhoria da Programação Dinâmica Dual (PDD), como a seleção de cortes, para

aprimorá-lo.

Outro aprimoramento importante a ser realizado é a migração do algoritmo implementado para uma linguagem de programação compilada como o C++, por exemplo. Como o python 3 é uma linguagem de programação interpretada e, portanto, menos eficiente quando se trata de velocidade de execução, espera-se uma diminuição significativa nos tempos de execução das metodologias implementadas com esta migração.

Além disso, propõe-se o melhoramento da modelagem do problema de planejamento de sistemas hidrotérmicos. Pode-se explorar o problema a partir de reservatórios equivalentes de energia, incluir os limites de intercâmbio para possibilitar a simulação do Sistema Interligado Nacional completo e acoplar na modelagem as funções de custo futuro geradas pelo modelo NEWAVE.

REFERÊNCIAS

- [1] ONS, 2019. Operador Nacional do Sistema Elétrico. Web page: <http://www.ons.org.br>, acessado em setembro de 2019
- [2] FORTUNATO, L. A. M., NETO, T. A. A., ALBUQUERQUE, J. C. R., PEREIRA, M. V. F., Introdução ao Planejamento da Expansão e Operação de Sistemas de Produção de Energia Elétrica. Universidade Federal Fluminense: Niterói, 1990.
- [3] MARCATO, A. L. M., (2002). Representação híbrida de sistemas equivalentes e individualizados para o planejamento da operação a médio prazo de sistemas de potência de grande porte. Rio de Janeiro: Tese de Doutorado, DEE, PUC-Rio.
- [4] A. L. Diniz, F. S. Costa, M. E. P. Maceira, T. N. Santos, L. C. Brandão, R. N. Cabral, "Short/Mid-Term Hydrothermal Dispatch and Spot Pricing for Large-Scale Systems - the Case of Brazil", 20th Power Systems Computation Conference, Dublin, Ireland, June 2018.
- [5] CEPEL, Centro de Pesquisas de Energia Elétrica, "Programação Diária da Operação de Sistemas Hidrotérmicos com Representação Detalhada das Unidades Geradoras, Consideração da Rede Elétrica e Restrições de Segurança - Modelo DESSEM", Manual de Metodologia, Abril/2019
- [6] MORAES, L. A. M., Considerações dos Contratos de Gás natural com Cláusulas Take-or-Pay no Planejamento Energético a Médio Prazo, Master's Thesis, IMPA, Rio de Janeiro, 2007.
- [7] TERRY, L. A., PEREIRA, M. V. F., NETO, T. A. A., SILVA, L. F. A., SALES, P. R. H., "Brazilian national hydrothermal electrical generating system". In: Interfaces, 1986.
- [8] M.E.P. Maceira, L.A. Terry, F.S. Costa, J. M. Damazio, A C. G. Melo, "Chain of optimization models for setting the energy dispatch and spot price in the Brazilian system", Proceedings of the Power System Computation Conference - PSCC'02, Sevilla, Spain, June 2002.
- [9] M.E.P. Maceira, D.D.J. Penna, A.L. Diniz, R.J. Pinto, A.C.G. Melo, C.V. Vasconcellos, C.B. Cruz , "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 PSCC - Power Syst. Comp. Conf., Dublin, Ireland, June, 2018.
- [10] BIRGE, J. R., LOUVEAUX, F. V., 1997, Introduction to Stochastic Programming. New York, Springer-Verlag. ISBN: 978-0-387-22618-7.
- [11] SHAPIRO, A., DENTCHEVA, D., RUSZCZYNSKI, A., 2014, Lectures on Stochastic Programming: Modeling and Theory, Second Edition. Philadelphia, PA, USA, Society for Industrial and Applied Mathematics. ISBN: 1611973422, 9781611973426.
- [12] J.R. Birge, C.J. Donohue, D.F. Holmes and O.G. Svintsitski, ND-UM version 1.0 computer code for the nested decomposition algorithm, Department of Industrial and Operations Engineering, University of Michigan, August 1994.

- [13] BENDERS, J. F., 1962, Partitioning procedures for solving mixed-variables programming problems", *Numerische Mathematik*, v. 4, n. 1, pp. 238252. ISSN: 0945-3245. doi: 10.1007/BF01386316.
- [14] SLYKE, R. M. V., WETS, R., 1969, L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming", *SIAM Journal on Applied Mathematics*, v. 17, n. 4, pp. 638663. doi: 10.1137/0117061.
- [15] J.R. Birge, "Decomposition and partitioning methods for multistage stochastic linear programs", *Operations Research*, v.33, n.5, pp. 989-1007, 1985.
- [16] BRANDÃO, L.C. Accelerating dual dynamic programming applied to hydrothermal coordination problems. 2018. Dissertação (Mestrado em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro, . Orientador: André Luiz Diniz Souto Lima.
- [17] BRANDÃO, L.C. ; DINIZ, A.L. ; SIMONETTI, L. G. . Accelerating Dual Dynamic Programming for Stochastic Hydrothermal Coordination Problems. In: 20th Power Systems Computation Conference, 2018. Proceedings of the 20th Power Systems Computation Conference, 2018.
- [18] DEMPSTER, M.A.H; THOMPSON, R.T. EVPI-based importance sampling solution procedures for multistage stochastic linear programmes on parallel MIMD architectures. In: 1999 Annals of Operations Research.
- [19] JESÚS M. Latorre, 2008, Analysis of stochastic problem decomposition algorithms in computational grids. Doi: 10.1007/s10479-008-0476-1
- [20] T. N. Santos, A. L. Diniz, C. T. Borges, "A New Nested Benders Decomposition Strategy for Parallel Processing Applied to the Hydrothermal Scheduling Problem", *IEEE Transactions on Smart Grid*, v. 8, n.3, pp. 1504-1512, 2017.
- [21] WOLF, C., KOBERSTEIN, A., 2013, sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped method", *European Journal of Operational Research*, v. 230, n. 1, pp. 143 156. ISSN: 0377-2217. doi: <http://dx.doi.org/10.1016/j.ejor.2013.04.017>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221713003159>>.
- [22] GASSMANN, H. I., 1990, : A computer code for the multistage stochastic linear programming problem", *Mathematical Programming*, v. 47, n. 1, pp. 407423. ISSN: 1436-4646. doi: 10.1007/BF01580872. Disponível em: <<http://dx.doi.org/10.1007/BF01580872>>.
- [23] WITTROCK, R. J., 1985, nested decomposition of staircase linear programs". In: Cottle, R. W. (Ed.), *Mathematical Programming Essays in Honor of George B. Dantzig Part I*, pp. 6586, Berlin, Heidelberg, Springer Berlin Heidelberg. ISBN: 978-3-642-00919-8. doi: 10.1007/BFb0121043. Disponível em: <<http://dx.doi.org/10.1007/BFb0121043>>.
- [24] MORTON, D. P., 1996, enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling", *Annals of Operations Research*, v. 64, n. 1, pp. 211235. ISSN: 1572-9338. doi: 10.1007/BF02187647. Disponível em: <<http://dx.doi.org/10.1007/BF02187647>>.

- [25] ALTENSTEDT, F., 2003, Aspects on asset liability management via stochastic programming. Tese de Doutorado, Chalmers University of Technology and Goteborg University.
- [26] LATORRE, J. M., CERISOLA, S., RAMOS, A., et al., 2008, of stochastic problem decomposition algorithms in computational grids", *Annals of Operations Research*, v. 166, n. 1, pp. 355373. ISSN: 1572-9338. doi: 10.1007/s10479-008-0476-1. Disponvel em: <<http://dx.doi.org/10.1007/s10479-008-0476-1>>.
- [27] DEMPSTER, M. A. H., THOMPSON, R. T., 1998, and aggregation of nested Benders decomposition", *Annals of Operations Research*, v. 81, n. 0, pp. 163188. ISSN: 1572-9338. doi: 10.1023/A:1018996821817. Disponível em: <<http://dx.doi.org/10.1023/A:1018996821817>>.
- [28] DOS SANTOS, T. N., DINIZ, A. L., 2009, A New Multiperiod Stage Definition for the Multistage Benders Decomposition Approach Applied to Hydrothermal Scheduling", *IEEE Transactions on Power Systems*, v. 24, n. 3 (Aug), pp. 13831392. ISSN: 0885-8950. doi: 10.1109/TPWRS.2009.2023265.
- [29] BIRGE, J. R., DONOHUE, C. J., HOLMES, D. F., et al., 1996, A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs", *Mathematical Programming*, v. 75, n. 2, pp. 327352. ISSN: 1436-4646. doi: 10.1007/BF02592158. Disponvel em: <<http://dx.doi.org/10.1007/BF02592158>>.
- [30] LINDEROTH, J., WRIGHT, S., 2003, Algorithms for Stochastic Programming on a Computational Grid", *Computational Optimization and Applications*, v. 24, n. 2, pp. 207250. ISSN: 1573-2894. doi: 10.1023/A:1021858008222. Disponvel em: <<http://dx.doi.org/10.1023/A:1021858008222>>.
- [31] MORITSCH, H. W., PFLUG, G. C., SIOMAK, M., 2001, Asynchronous nested optimization algorithms and their parallel implementation", *Wuhan University Journal of Natural Sciences*, v. 6, n. 1 (Mar), pp. 560 567. ISSN: 1993-4998. doi: 10.1007/BF03160302. Disponvel em: <<https://doi.org/10.1007/BF03160302>>.
- [32] A. E. Eiben and C. Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1):3550, 1998.
- [33] DARWIN, C., A Origem das Espécies e a Seleção Natural. 2000
- [34] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag, Berlin, 2003.
- [35] MONTES, E. M., Técnicas Alternativas para o Tratamento de Restrições na Otimização Evolucionária, Ph.D. Thesis, Centro de Investigação de Estudos Avançados do Instituto Politécnico Nacional - México, 2004
- [36] HOLLAND, J., *Adaptation in Natural and Artificial Systems*. 1975
- [37] GOLDBERG, D., *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley Publishing Co., 1989
- [38] GOLDBERG, D., L.B., B., HOLLAND, J., "Classifier systems and genetic algorithms", *Machine Learnings: Paradigms and Methods - MIT/Press Elsevier* pp. 235-282, 1989

- [39] MIRANDA, V., SRINIVASAN, D. e PROENÇA, L. M., Evolutionary computation in power systems, *International Journal of Electrical Power Energy Systems*, vol. 20, n.2, pp. 89-98, 1998.
- [40] LEITE, P. T., CARNEIRO, A. A. F. M., e CARVALHO, A. C. P. L. F., Energetic operation planning using genetic algorithms., *IEEE Transactions on Power Systems*, vol. 17, n.1, pp. 173–179, 2002a.
- [41] LEITE, P. T.; CARNEIRO, A. A. F. M.; CARVALHO, A. C. P. L. F., “Aplicação de Algoritmos Genéticos na Determinação da Operação Ótima de Sistemas Hidrotérmicos de Potência”. *Revista Controle e Automação*, v.17, n.1, 2006.
- [42] SOARES, M. P., Otimização Multicritério da Operação de Sistemas Hidrotérmicos Utilizando Algoritmos Genéticos. Dissertação de mestrado, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, 2008.
- [43] 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 J PWRS* , v. 23, n. 3, pp. 1298–1308, 2008.
- [44] RAMOS, T. P. Modelagem Híbrida para o Planejamento da Operação de Sistemas Hidrotérmicos Considerando as Não Linearidades das Usinas Hidráulicas. Tese de Doutorado — Universidade Federal de Juiz de Fora, 2015.
- [45] GRAHAM, R., “An Efficient Algorithm for Determining the Convex Hull of a Finite Point Set”, *Info. Proc. Letters*, v. 1, pp. 132–133, 1972
- [46] BARBER, C. B., DOBKIN, D. P., HUHDANPAA, H., “The quickhull algorithm for convex hulls”, *ACM Transactions on Mathematical Software (TOMS)*, v. 22, n. 4, pp. 469–483, 1996.
- [47] PMO, 2019. PROGRAMA MENSAL DA OPERAÇÃO. Web page: <http://ons.org.br>. Acessado: 2019-07-07.