



XXII SNPTEE
SEMINÁRIO NACIONAL
DE PRODUÇÃO E
TRANSMISSÃO DE
ENERGIA ELÉTRICA

BR/GOP/17
13 a 16 de Outubro de 2013
Brasília - DF

GRUPO – IX

GRUPO DE ESTUDO DE OPERAÇÃO DE SISTEMAS ELÉTRICOS - GOP

PROCESSAMENTO PARALELO PARA PROGRAMAÇÃO DINÂMICA DUAL DETERMINÍSTICA: APLICAÇÃO À PROGRAMAÇÃO DIÁRIA DA OPERAÇÃO HIDROTÉRMICA

Tiago Norbiato dos Santos(*)
COPPE/UFRJ

André Luiz Diniz
CEPEL/UERJ

Carmen Lucia Tancredo Borges
COPPE/UFRJ

RESUMO

O processamento paralelo tem sido utilizado de forma significativa em otimização estocástica, ao paralelizar a resolução dos subproblemas de diferentes cenários. Porém, em problemas determinísticos o emprego de mais de um processador não é tão direto, pois todos os subproblemas apresentam uma hierarquia temporal. Este artigo propõe uma estratégia para aplicar processamento paralelo na Programação Dinâmica Dual Determinística, paralelizando a resolução dos estágios e adotando uma estratégia eficiente para a utilização da condição inicial do estágio anterior e para a transferência dos cortes de Benders. A metodologia proposta é aplicada ao problema de Programação Diária da Operação de sistemas hidrotérmicos.

PALAVRAS-CHAVE: Processamento Paralelo, Programação Dinâmica Dual Determinística, Programação Diária da Operação de Sistemas Hidrotérmicos.

1.0 - INTRODUÇÃO

Em muitos setores da indústria, do comércio e do transporte, modelos (ou problemas) matemáticos são utilizados como ferramentas auxiliares em tarefas de análises de dados e/ou em procedimentos operacionais. Esses problemas podem exigir um grande esforço computacional para serem resolvidos e, em certos casos, devido a sua grande dimensão, pode não ser possível se obter uma solução. Para contornar este inconveniente pode-se dividir o problema em diversos subproblemas menores(1), que necessitem de recursos computacionais inferiores, mas que devem ser resolvidos mais de uma vez, de forma iterativa. Em muitos casos estes subproblemas podem ser resolvidos simultaneamente possibilitando a utilização de mais de um computador ao mesmo tempo.

Uma eficiente metodologia utilizada para decompor problemas de otimização de grandes dimensões e que apresentem certa estrutura de dependência temporal é a Programação Dinâmica Dual (PDD), que o transforma em E subproblemas (chamados de estágios) e resolve o problema como um todo por Decomposição de Benders multi-estágio(2). Nesta técnica, os estágios são resolvidos em passos *Forward* (onde os estágios fornecem condições iniciais para os estágios seguintes), e *Backward* (onde os estágios geram Cortes de Benders para os estágios anteriores). Para problemas de grande porte, onde é impossível percorrer todos os nós da árvore de cenários, pode-se aplicar a variante amostral desse método, conhecida como Programação Dinâmica Dual Estocástica (PDDE) (3). Esta técnica facilita que os subproblemas referentes a diferentes séries ou cenários, que não possuem dependência temporal entre si, sejam naturalmente resolvidos de forma simultânea, utilizando-se processamento paralelo e reduzindo o tempo computacional.

(*)Avenida Horácio Macedo, 354 - Cidade Universitária – RJ - CEP 21941-911 Rio de Janeiro, RJ – Brasil
Tel: (+55 21) 2598-6245 – Fax: (+55 21) 2598-6482 – Email: norbiato@cepel.br

Em estudos determinísticos, esta metodologia é denominada de Programação Dinâmica Dual Determinística (PDDD) e, neste caso, a abordagem de paralelização não é direta como em estudos estocásticos, pois, tradicionalmente os estágios são resolvidos de forma sequencial, devido à dependência temporal entre eles(4).

Ou seja, nos passos *Forward* todos os estágios (identificados pelo índice “ e ”), exceto o último, fornecem uma condição inicial para o estágio $e+1$ e nos passos *Backward* todos os estágios, exceto o primeiro, constróem um corte de Benders para o estágio $e-1$.

Na área de geração de energia elétrica, em particular em sistemas hidrotérmicos de grande porte como o brasileiro, é necessário um planejamento para se obter uma operação coordenada e ótima. Os problemas de otimização associados ao planejamento em um horizonte de tempo mais longo têm sido resolvidos em muitos casos aplicando-se as técnicas de PDD e PDDE e utilizando processamento paralelo, como pode ser visto em (5)-(9). Este trabalho propõe uma nova abordagem para a utilização de processamento paralelo no método de PDDD aplicado ao problema de programação diária da operação (PDO), onde técnicas de paralelização em geral não têm sido aplicadas pelo fato de se tratar de um problema determinístico.

Na abordagem proposta, todos os estágios da PDDD são resolvidos simultaneamente, um em cada processador. Para as restrições que utilizam variáveis que foram decididas no estágio anterior ($e-1$), utiliza-se o resultado obtido na iteração anterior e, a cada resolução de um estágio e , obtém-se um novo Corte de Benders para o estágio $e-1$ e novos valores para as variáveis de estado do estágio $e+1$, os quais serão utilizados na próxima iteração. Mostram-se resultados para o sistema brasileiro, com aproximadamente 130 usinas hidroelétricas e 60 termoelétricas. Os resultados mostram a validade da técnica proposta, obtendo-se o mesmo intervalo para o custo ótimo de operação que a técnica de Benders tradicional, porém com uma redução significativa no tempo computacional.

2.0 - PLANEJAMENTO DA OPERAÇÃO DE SISTEMAS HIDROTÉRMICOS

O Planejamento da Operação de Sistemas Hidrotérmicos é um complexo problema de otimização. As decisões são acopladas no tempo: os volumes armazenados nos reservatórios no futuro dependem da operação do sistema no tempo presente, as gerações das usinas hidráulicas e térmicas devem ser coordenadas não somente visando ao atendimento das restrições do sistema como demanda e a reserva de potência, mas também devem considerar a existência de diversas usinas em cascata. No Brasil, o Planejamento da Operação de Sistemas Hidrotérmicos é dividido em 3 principais etapas como definidas a seguir (10):

- Médio Prazo: nesta etapa é traçada uma estratégia para a operação considerando as incertezas do futuro com um horizonte de até 5 anos, resolvendo o problema estocástico aplicando-se PDDE;
- Curto Prazo: com a política de operação adotada na etapa de médio prazo, define-se uma meta de operação para a próxima semana, com um maior detalhamento das usinas e um horizonte de até um ano. As incertezas são representadas por uma árvore de cenários “completa” e o problema é resolvido por PDD.
- Programação Diária da operação (PDO): por fim, com as metas de operação definidas pela etapa de curto prazo define-se uma operação para o sistema. Como no Brasil as fontes intermitentes de energia como a solar e eólica ainda não têm um grande impacto na operação, pois consistem num pequeno percentual da potência instalada no país, nesta etapa o sistema é modelado de forma determinística, assumindo-se um conhecimento razoável das afluições às usinas hidroelétricas em um horizonte de até 1 semana

A metodologia proposta neste trabalho será aplicada ao problema de PDO, descrito a seguir.

2.1 Programação Diária da Operação (PDO)

Nesse problema, as usinas hidroelétricas são representadas de forma individualizada e a rede elétrica é considerada por um modelo DC linear. O objetivo é obter a geração para cada usina hidroelétrica e térmica do sistema que minimize o custo total de operação, dado pelo custo com o combustível utilizado pelas usinas termoelétricas e pelo valor esperado do custo futuro dado em função dos armazenamentos nos reservatórios. Maiores detalhes do modelo de PDO podem ser vistos em (11).

A Função objetivo definida na equação [1] consiste na minimização dos custos presentes, dado pela soma dos custos com as gerações térmicas e pelo Custo Futuro, avaliado a partir de uma Função de Custo Futuro (FCF) $\alpha^T(V^T)$ acoplada no final do horizonte de estudo. A Equação [2] é o Balanço Hidráulico que representa a lei de conservação da água,¹.

¹ Neste trabalho, desprezou-se o tempo de viagem da água (12) por questões de simplicidade.

A equação [3] é a Função de Produção Hidráulica Aproximada (FPHA) das usinas Hidroelétricas, que representa a conversão da energia potencial em energia elétrica, sendo que a variação da produtividade com a queda é modelada como uma função linear por partes das variáveis operativas associadas à usina (13). As restrições apresentadas em [4] são os limites de geração das usinas termoeletrônicas. O balanço de carga é definido pelo fluxo de potência por um modelo linear (DC) da rede elétrica [5]. Por fim a equação [6] define os limites de fluxos nas linhas da rede elétrica.

$$\min \sum_{t=1}^T \left(\sum_{j=1}^{NT} CT_j GT_j^t \right) + \alpha^T (V^T) \quad [1]$$

$$V_i^t = V_i^{t-1} - (Q_i^t + S_i^t) + \sum_{k \in M_i} (Q_k^t + S_k^t) + A_i^t, \quad i=1, NH; \quad t=1, T \quad [2]$$

$$GH_i^t \geq fpha(V_i^{t-1}, V_i^t, Q_i^t, S_i^t), \quad i=1, NH; \quad t=1, T \quad [3]$$

$$\underline{GT_j^t} \leq GT_j^t \leq \overline{GT_j^t} \quad j=1, NT; \quad t=1, T \quad [4]$$

$$P - D = B\Theta \quad [5]$$

$$-\overline{f_{km}^t} \leq f_{km}^t \leq \overline{f_{km}^t} \quad e \quad f_{km}^t = \frac{1}{\gamma_{km}} (\theta_k^t - \theta_m^t) \quad [6]$$

onde T é o número de intervalos de tempo (discretização do horizonte de estudo); NT é o número de usinas térmicas; CT_j é o custo incremental da usina térmica j ; GT_j^t é a geração da usina térmica j no período t ; $\alpha^T(V^T)$ é o Custo Futuro no fim do horizonte de estudo em função dos volumes armazenados no fim do estudo; V^T é o vetor com os volumes dos reservatórios no fim do estudo; NH é o número de usinas hidroelétricas; V_i^t é o volume armazenado na usina i ao final do período t ; Q_i^t e S_i^t são os volumes turbinado e vertido na usina i no período t ; M_i é o conjunto de usinas à montante da usina i ; A_i^t é o volume incremental natural afluente à usina i ; GH_i^t é a geração da usina hidroelétrica i no período t ; P é o vetor de gerações em cada barra; D é o vetor de carga de cada barra; B é a matriz de susceptância que define as ligações da rede elétrica; Θ é o vetor de ângulos das barras; f_{km} é o fluxo da linha que liga as barras k e m ; γ_{km} é a susceptância da linha que liga as barras k e m ; θ_k e θ_m são os ângulos das barras k e m .

2.2 Estratégia de Solução: Programação Dinâmica Dual Determinística - PDDD

O problema como definido na seção 2.1 pode ser resolvido por um único Problema de Programação Linear (PPL), ou por PDDD. Na estratégia de PDDD, troca-se a resolução de um PPL de grande dimensão por um processo iterativo de resolução de diversos PPL's, denominados "estágios", resolvendo-se o problema como um todo por decomposição de Benders multi-estágio(2). A Figura1 ilustra a comunicação entre os estágios. As setas vermelhas indicam o envio das condições iniciais x^0 e as setas azuis o envio dos cortes de Benders.

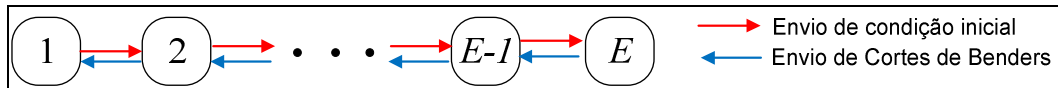


Figura1 – Comunicação entre os estágios na PDDD

Na PDDD tradicional, cada iteração é composta por três etapas:

- **Simulação Forward:** Nesta etapa todos os estágios são resolvidos em sequencia desde o estágio 1 até o estágio E . As variáveis de estado para o estágio e são obtidas a partir das soluções $(x^1, x^2, \dots, x^{e-1})$ dos subproblemas dos estágios anteriores. Ao final desta etapa obtêm-se um Limite Superior (Z_{sup}) para o custo total ótimo do problema original o qual é obtido pela soma dos custos imediatos de cada estágio.
- **Recursão Backward:** Nesta etapa os estágios são resolvidos em sequencia, desde o estágio E até o estágio 1. A partir da solução primal e dual de cada estágio e , constrói-se um corte de Benders para o estágio imediatamente anterior ($e-1$). A solução do estágio 1, nesta etapa, fornece um Limite Inferior (Z_{inf}) para o custo total ótimo do problema original obtido pelo custo total do estágio 1.
- **Teste de Convergência:** O critério de parada tem por base os valores obtidos para Z_{inf} e Z_{sup} nas etapas anteriores. O processo iterativo é finalizado quando a diferença percentual entre estes valores, definida por gap , for menor do que uma dada tolerância.

A Figura 3(a), na próxima seção, ilustra a ordem em que os estágios são resolvidos na PDDD tradicional, considerando um exemplo em que o problema original foi dividido em 4 estágios e o processo iterativo é iniciado por uma *Simulação Forward*. As setas verdes indicam a ordem dos estágios para o cálculo do valor de Z_{sup} de cada iteração.

Uma forma intuitiva de decompor o problema é definir cada intervalo de tempo t em um estágio e neste caso temos que $E=T$. Porém existem outras estratégias de decomposição como mostrado em (4), onde foi apresentada uma abordagem de agregar em um mesmo estágio mais de um intervalo de tempo, reduzindo assim o número de PPLs a serem resolvidos e o número de iterações, pois com a agregação as dependências temporais são representadas de forma explícita dentro do PPL, para a obtenção da solução do problema como um todo e, em alguns casos, obtendo um ganho no tempo computacional. Ressalta-se que a agregação dos intervalos de tempo não reduz a precisão do modelo, pois trata-se apenas de um rearranjo das variáveis e restrições.

Assim há uma decomposição ótima obtida através da análise de trade off entre o número de estágios e o seu tamanho: quanto maior o estágio, menor será o número de PPL's a serem resolvidos, porém cada estágio exigirá um esforço computacional maior; por outro lado, quando menor o estágio maior será o número de PPL's a serem resolvidos, porém o esforço computacional exigido por cada PPL será menor. Este ganho também depende do detalhamento do sistema a ser utilizado: por exemplo, se considerarmos as perdas na rede elétrica, será necessário um maior esforço para se resolver o problema com um único PPL e assim torna-se mais vantajosa a decomposição deste problema. Em contrapartida a consideração do tempo de viagem da água entre os reservatórios favorece resolver o problema por um único PPL, pois esta característica das usinas em cascatas acopla fortemente os intervalos de tempo. A Figura 2 ilustra a agregação de períodos em um mesmo estágio, considerando um exemplo onde em cada estágio estão agregados 2 intervalos de tempo.

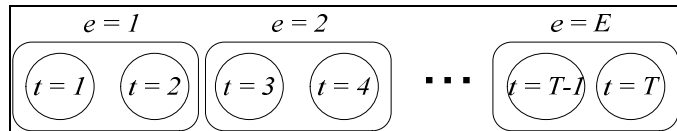


Figura 2 – Definição dos estágios na Programação Diária da Operação

3.0 - PROCESSAMENTO PARALELO APLICADO A PROGRAMAÇÃO DINÂMICA DUAL DETERMINISTICA

Na abordagem proposta, a decomposição do problema é feita da mesma forma como apresentado na seção anterior, porém todos os estágios são resolvidos simultaneamente, um em cada processador. Para as restrições que utilizam uma condição inicial fornecida pelo estágio anterior ($e-1$), como por exemplo, as restrições de balanço hidráulico [2], utiliza-se o resultado obtido na iteração anterior. A cada resolução de um estágio e , obtêm-se um novo corte de Benders para o estágio $e-1$ e um novo resultado que servirá de condição inicial para o estágio $e+1$ na próxima iteração. Ressalta-se que, apesar de não se seguir a sequência tradicional na resolução dos subproblemas, todos os cortes de Benders construídos são válidos, pois constituem uma aproximação inferior da função de recurso de cada estágio, a partir de um ponto inicial obtido pelo estágio anterior. Ao final do processo deve-se realizar um passo *Forward* de forma sequencial, ou seja, sem processamento paralelo, que fornecerá a solução.

A Figura 3 compara a ordem de execução e o fluxo de informações entre os estágios na metodologia tradicional (a), descrita na seção anterior, e na metodologia proposta neste artigo (b). Considerou-se um exemplo no qual o problema original foi dividido em 4 estágios e que o processo iterativo foi finalizado com 5 iterações na abordagem proposta e 2 iterações na metodologia tradicional. As linhas vermelhas representam os passos *Forward* e o envio das condições iniciais do estágio e para o estágio $e+1$. As linhas em azul representam os passos *Backward* e o envio dos cortes de Benders do estágio e para o estágio $e-1$. Por fim as linhas verdes indicam uma simulação completa para se obter um limite superior Z_{sup} para o valor ótimo da função objetivo, o qual deve corresponder a uma resolução sequencial dos subproblemas, pois consiste do custo de uma solução viável para o problema. Observe que, no método proposto, o valor de Z_{sup} também é obtido de forma sequencial, pois o estágio 2 é resolvido a partir da solução do estágio 1, e assim por diante.

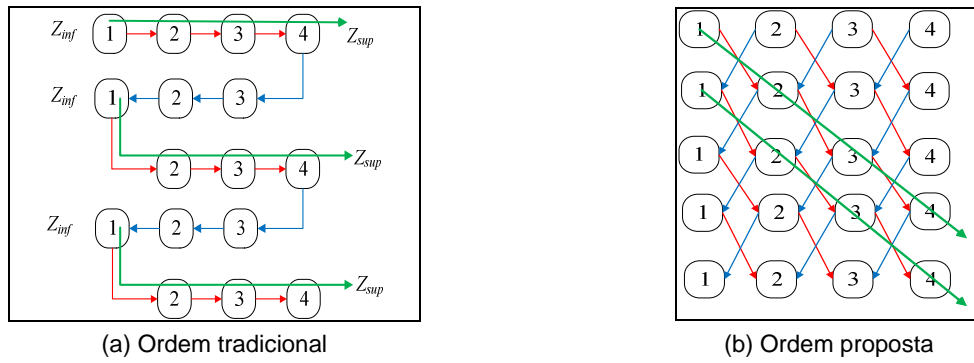


Figura 3 - Comparação entre a metodologia tradicional e proposta para a PDDD

Observe que na metodologia proposta são resolvidos, no total, 20 PPL's, sendo que cada processador resolve apenas 5. Na metodologia tradicional são resolvidos 16 PPL's, todos pelo mesmo processador, ou seja, um de cada vez. A vantagem da metodologia tradicional está no fato de que cada PPL sempre é resolvido tendo como estado inicial do sistema a solução do estágio anterior na mesma iteração, enquanto que na abordagem proposta os PPL's são resolvidos com a solução do estágio anterior na iteração anterior. Desta forma, nas primeiras iterações da abordagem proposta (vide Figura 4) pode ser utilizado um estado inicial para o sistema que não seja viável, resultando na construção de cortes desnecessários para a efetiva resolução do problema.

3.1 Análise de convergência e limites para a solução ótima

Como os estágios são resolvidos com uma condição inicial que não foi gerada pelo estágio anterior na mesma iteração, na metodologia proposta o limite superior Z_{sup} não pode ser obtido pela soma dos custos imediatos de uma mesma iteração. A Figura 4, a qual é um refinamento da Figura 3, ilustra o processo iterativo na metodologia proposta, detalhando o comportamento dos limites da solução ótima e qualificando os cortes obtidos ao longo das iterações. Observe que o valor de Z_{sup} deve ser obtido de forma "diagonal", para que os custos imediatos correspondam a resultados de problemas montados com condições iniciais para cada estágio que efetivamente foram obtidas percorrendo-se todos os estágios anteriores. Ressalta-se que o limite Z_{inf} será sempre válido, pois é a solução do primeiro estágio, que foi obtida com cortes de Benders válidos. Vale lembrar que os primeiros cortes construídos foram gerados por estágios que ainda não tinham informação "nem do futuro" e "nem do passado".

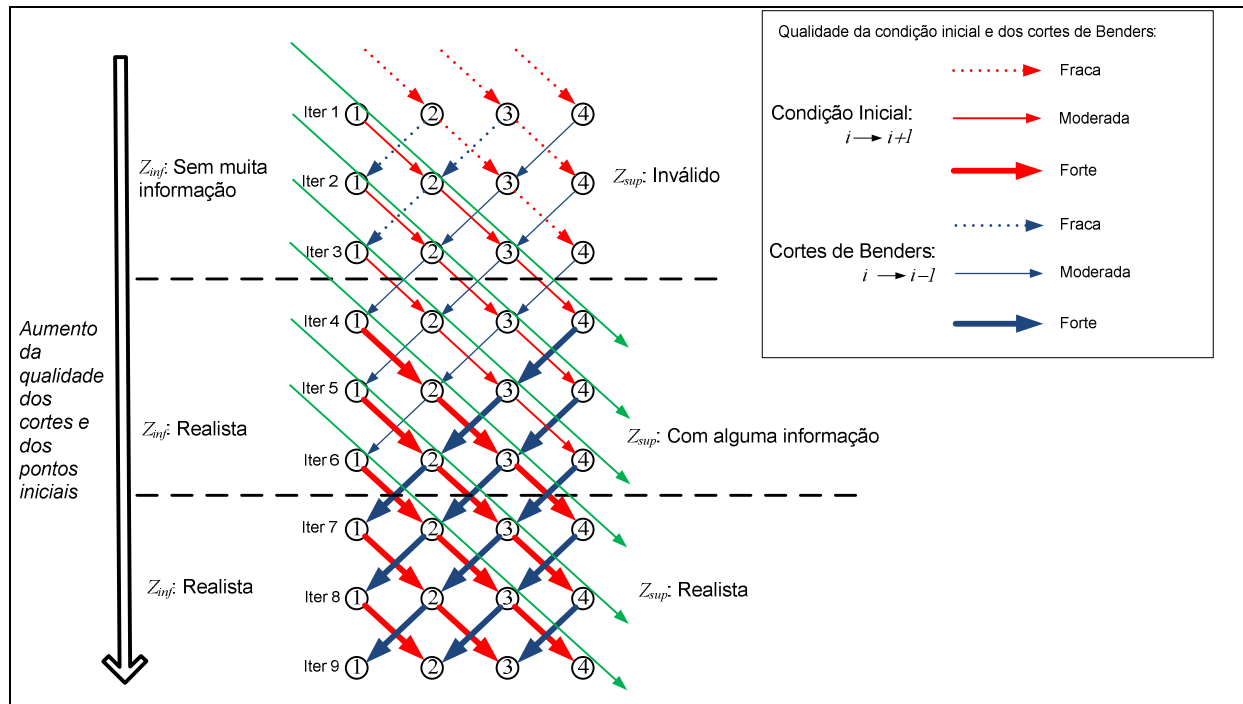


Figura 4 - Ordem de resolução dos estágios na metodologia proposta para a PDDD

Tomemos como exemplo o estágio 2. Na primeira iteração, sua condição inicial foi o ponto de partida do estudo e não uma condição gerada pelo estágio 1, ou seja, ele ainda não "conhecia o passado". O estágio 2 também foi resolvido sem nenhum Corte de Benders gerado pelo estágio 3, ou seja, também não "conhecia o futuro". Desta forma, a condição inicial gerada para o estágio 3 pode ser de baixa qualidade e o corte de Benders enviado ao estágio 1 com poucas informações sobre os estágios 3 e 4.

Na segunda iteração, o estágio 2 é resolvido com uma condição inicial gerada pelo estágio 1, porém com um corte de baixa qualidade gerado pelo estágio 3 (que ainda não conhecia o futuro na iteração 1). Na terceira iteração, o estágio 2 é resolvido com um corte de Benders gerado na 2ª iteração pelo estágio 3, que por sua vez foi resolvido com um corte de Benders gerado pelo estágio 4 na primeira iteração. Ou seja, na iteração de número 3, o estágio 2 gera um corte com informações de todos os estágios posteriores a ele.

Ressalta-se, nesse momento, ainda se considera a qualidade das informações do ponto inicial e dos cortes de Benders como "moderada", já que que nestas primeiras iterações os estágios 3 e 4 ainda geraram cortes sem ter informações sobre o estágio 1 e 2. Assim, serão necessárias algumas iterações para que a informação do estágio 1 seja transferida até o estágio 4 e a informação dos cortes de Benders do estágio 4 seja transferida até o estágio 1. Podemos concluir que, na abordagem proposta, cada processador resolve apenas um único estágio (várias vezes), porém as informações entre os mesmos precisa de mais iterações para serem transmitidas. Na Figura 4 podemos observar que apenas na 7ª iteração o estágio 1 é resolvido com cortes gerados pelos estágios 2, 3 e 4 e que foram construídos com base em uma condição inicial efetivamente obtida a partir da resolução dos estágios 1, 2 e 3 respectivamente.

Por fim, o limite Z_{sup} é obtido com as soluções dos estágios indicados pelas setas verdes. Observa-se que o custo imediato utilizado para compor esse valor é obtido com uma condição inicial gerada pelo estágio imediatamente anterior, como em uma simulação *Forward* da metodologia tradicional. O valor Z_{inf} é obtido pela resolução do primeiro estágio. O *gap* e o teste de convergência são os mesmos adotados no processamento tradicional.

Na figura 4 as linhas azuis representam o envio dos Cortes de Benders e as linhas vermelhas o envio das condições iniciais. Quanto mais grossa for a linha melhor as informações transmitidas entre os processadores.

3.2 Algoritmo

O algoritmo para a utilização do processamento paralelo na PDDD aplicada ao problema de PDO foi construído como base em 6 principais passos, como definido a seguir, e foi implementado utilizando o sistema de troca de mensagem *MPI – MessagePassing Interface* (14),(15). A topologia de processamento paralelo adotada foi a "Mestre – Escravos". A Figura 5 ilustra o fluxograma do processamento paralelo desse algoritmo:

- PPL(e): Resolve um estágio da PDDD, definido pelo parâmetro e ;
- Calcula Custo: Rotina para o calculo do custo total de Operação. Nesta rotina, todos os processos escravos enviam ao processo mestre o valor do custo de operação de cada estágio/processo através das rotinas *MPI_SEND* e *MPI_RECV*.
- Testa Convergência: O processador mestre obtém o valor de Z_{inf} e calcula o *gap* da iteração e verifica a convergência. Por fim o mestre, através da rotina *MPI_BCAST*, envia o resultado do teste de convergência para todos os processos escravos.
- Envio de Cortes: Cada processo gera e envia um Corte de Benders para o estágio/processo anterior. O envio e o recebimento são feitos através das rotinas *MPI_SEND* e *MPI_RECV* respectivamente.
- Envio das condições iniciais para os estágios: Cada processo envia o volume final dos reservatórios de cada estágio/processo para o estágio posterior. Esta rotina também utiliza as rotinas *MPI_SEND* e *MPI_RECV*.

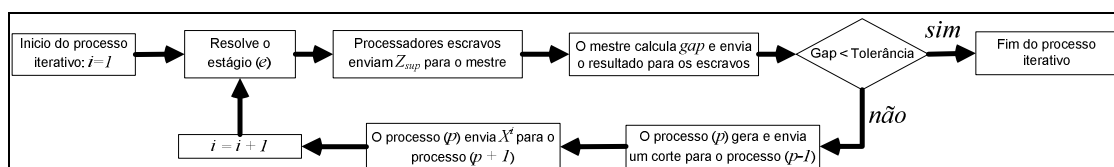


Figura 5– Fluxograma do algoritmo para a PDDD com processamento paralelo

4.0 - ESTUDO DE CASOS

A abordagem proposta foi aplicada no Problema de PDO, utilizando como base a Programação Mensal da Operação (PMO) de Novembro/Dezembro de 2012, processo coordenado pelo Operador Nacional do Sistema (ONS). O horizonte de estudo é de um dia com discretização horária.

Foram utilizados todos os possíveis números de estágios, ou seja, 2, 3, 4, 6, 8, 12 e 24 (no caso de 1 estágio não foi utilizada a PDDD). O número de processadores utilizado sempre foi igual ao número de estágios. O estudo foi executado tanto de forma serial (abordagem tradicional), quanto em paralelo (abordagem proposta) utilizando uma tolerância de otimalidade de $10^{-8}\%$. Com esta tolerância, muito inferior à adotada na prática, os valores de Z_{inf} e Z_{sup} devem ser muito próximos, o que permite avaliar a consistência da metodologia proposta. Os resultados para este caso são apresentados na Tabela 1.

Para a análise de desempenho do processamento paralelo foram utilizados os parâmetros *Speedup*, medido pela relação entre o tempo computacional para a execução em 1 processador e em "*p*" processadores, e a Eficiência que é medida pela relação entre o *Speedup* e o número de processos "*p*".

Tabela 1- Comparação entre a PDDD tradicional e a proposta

# de estágios	Abordagem	# de iterações	Tempo (s)	Speedup	Eficiência(%)	Z_{inf} (\$1000)	Z_{sup} (\$1000)
1	Serial	1	115,0	-	-		43.865.406,5
2	Serial	21	123,5	0,99	49,4	43.865.406,5	43.865.406,5
	Paralelo	40	125,0			43.865.406,5	43.865.406,5
3	Serial	30	195,9	1,20	39,9	43.865.406,5	43.865.406,5
	Paralelo	75	163,6			43.865.406,5	43.865.406,5
4	Serial	34	230,6	1,16	29,0	43.865.406,5	43.865.406,5
	Paralelo	131	198,6			43.865.406,5	43.865.406,5
6	Serial	65	466,7	1,61	26,8	43.865.406,5	43.865.406,5
	Paralelo	269	290,6			43.865.406,5	43.865.406,5
8	Serial	97	703,7	2,14	26,7	43.865.406,5	43.865.406,5
	Paralelo	327	328,9			43.865.406,5	43.865.406,5
12	Serial	125	971,9	1,35	11,2	43.865.406,5	43.865.406,5
	Paralelo	393	259,7			43.865.406,5	43.865.406,5
24*	Serial	197	1566,0	6,36	26,5	43.865.406,5	43.865.406,5
	Paralelo	628	251,1			43.865.415,6	43.865.406,5

*Apresentou "gap negativo"

Podemos observar que todos os valores de Z_{inf} e Z_{sup} são iguais ou muito próximos. Com isso, comprovamos que a metodologia está coerente em termos de resultados, pois apresentam o mesmo Custo Total Ótimo para a operação. A única execução em que o valor de Z_{inf} não foi igual aos outros foi quando se utilizou 24 processadores e neste caso o valor de Z_{sup} foi menor do que o de Z_{inf} , configurando o que se chama de "gap negativo". Este problema pode ter sido provocado por alguma dificuldade numérica durante a resolução do PPL na última iteração, pois verificou-se que na penúltima iteração (627) desse caso, o valor de Z_{inf} era exatamente igual ao valor dos demais casos. Podemos observar que em todos os casos, à exceção do caso com 2 estágios, a metodologia paralela teve um tempo de processamento menor que a sequencial, mesmo sendo processado um número muito maior de iterações, o que era esperado como discutido na seção 3.1

5.0 - CONCLUSÕES

Neste trabalho foi proposta uma nova estratégia para possibilitar a aplicação de processamento paralelo em problemas determinísticos, como por exemplo, a Programação Diária da Operação de Sistemas Hidrotérmicos. Os resultados mostram a validade da técnica proposta, obtendo-se o mesmo intervalo para o custo ótimo de operação que a técnica de Benders tradicional, porém com uma redução significativa no tempo computacional para resolver o problema. Verificou-se que o *SpeedUp* e a Eficiência do processamento paralelo não foram muito bons, porém se espera um ganho nestes parâmetros com os trabalhos futuros a serem desenvolvidos. Duas principais melhorias serão feitas no código fonte visando um melhor desempenho no processamento paralelo:

- Montagem do problema: na implementação utilizada nos estudos deste trabalho, o problema de programação linear de cada estágio era construído a cada iteração. Porém, na metodologia proposta esta montagem pode ser feita apenas na primeira iteração, pois nas seguintes deve-se apenas inserir novos cortes e atualizar as condições iniciais, já que cada processador resolverá sempre o Problema de Programação Linear referente ao mesmo estágio. Ressalta-se que, na metodologia tradicional (processamento serial), isso só é possível à custa de memória para armazenagem, em arquivo ou em variáveis replicadas, dos subproblemas referentes a todos os estágios da PDDD.
- Gerenciamento de cortes: nas primeiras iterações, os cortes construídos em cada estágio *e* não possuem muitas informações sobre os demais estágios. Por isso, são poucos eficientes e poderiam ser retirados do problema, reduzindo o tamanho (número de restrições) do PPL e o tempo para a sua resolução. Uma estratégia para gerenciamento de cortes de Benders em problemas resolvidos por PDD é apresentado em (16).

6.0 - REFERÊNCIAS BIBLIOGRÁFICAS

- (1) C.A. Sagastizábal, "Divide to conquer: decomposition methods for energy optimization", *Mathematical Programming Series B*, v.134, n.1, pp. 187-222, 2012.
- (2) J.R.Birge, "Decomposition and partitioning methods for multistage stochastic linear programs", *Operations Research*, v.33, n.5, pp. 989-1007, 1985.
- (3) M. V. F. Pereira, L. M. V. G. Pinto, "Multi-stage stochastic optimization applied to energy planning", *Mathematical Programming*, v. 52, n.1-3, pp. 359-375, May 1991
- (4) T. N. Santos, A. L. Diniz, "A New Multi-Period Stage Definition for the Multi-Stage Benders Decomposition Approach Applied to Hydrothermal Scheduling", *IEEE Transactions Power Systems*, Vol. 24, N°3, 2009.
- (5) M. Dempster, R.T. Thompson, "Parallelization and aggregation of nested Benders decomposition", *Annals of Operations Research*, v.81, pp. 163-187, 1998.
- (6) R. J. Pinto, "Aplicação de processamento paralelo ao problema de planejamento da operação de sistemas hidrotérmicos baseado em cluster de computadores.", Rio de Janeiro, UFRJ/COPPE, 2011, tese (doutorado), programa de engenharia elétrica
- (7) R. J. Pinto, C. L. T. Borges, M. E. P. Maceira, "An Efficient Parallel Algorithm for Large Scale Hydrothermal System Operation Planning", Approved for publication in *IEEE Transactions On Power Systems*.
- (8) E.L. da Silva, E.C. Finardi, "Parallel processing applied to the planning of hydrothermal systems", *IEEE Transactions on Parallel and Distributed Systems*, v.14, n.8, pp. 0721-0729, 2003
- (9) R. J. Pinto, A. L. G. P. Sabóia, R. N. Cabral, F. S. Costa, A. L. Diniz e M. E. P. Maceira, "Metodologia para Aplicação de Processamento Paralelo no Planejamento de Curto-Prazo (Modelo Decomp)", *SNPTEE - Seminário Nacional De Produção E Transmissão De Energia Elétrica*, 2009, Recife-PE.
- (10) 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.
- (11) T. N. Santos, C. E. V. Boas, F. P. Mourão, A. L. Diniz, "Restrições de Metas Semanais na Política de Operação do Sistema Elétrico Brasileiro.", *XII SEPOPE – Symposium of Specialists in Electric Operational and Expansion Planning*, Rio de Janeiro, Brasil, May 2012.
- (12) T. M. Souza, A. L. Diniz, "An Accurate Representation of Water Delay Times for Cascaded Reservoirs in Hydro Scheduling Problems". In: *2012 IEEE PES General Meeting*, 2012, San Diego. *Proceedings of the 2012 IEEE PES General Meeting*, 2012.
- (13) A. L. Diniz, M. E. P. Maceira, "A four-dimensional model of hydro generation for the short-term hydrothermal dispatch problem considering head and spillage effects", *IEEE Trans. Power Syst.*, v. 23, n.3, pp. 1298-1308, Aug. 2008.
- (14) MPI Forum, *MPI: A Message Passing Interface Standard*, Technical Report UT-CS-94-230, Department of Computer Science, Michigan State University, 1994.
- (15) Snir , M., Otto , S., Huss-Lederman , S., Walker, D., e Dongarra , J., "MPI: The Complete Reference - Volume 1 - The MPI Core", MIT Press, 1998
- (16) V.L. de Matos, A.B. Philpott, E.C. Finardi, "Improving the Performance of Stochastic Dual Dynamic Programming", *Optimization Online*, Jul 2012.

7.0 - DADOS BIOGRÁFICOS



Tiago Norbiato: nasceu em Cachoeiro de Itapemirim em 1981, possui graduação em Engenharia Elétrica (2004) pela UFJF, mestrado em Engenharia de Ciências de Computação (2009) e desde 2012 está cursando doutorado em Engenharia Elétrica, ambos pela UFRJ/COPPE. Desde 2004 atua como pesquisador no CEPEL na área de planejamento de operação de sistemas hidrotérmicos.

André Luiz Diniz: possui graduação em Engenharia Civil (1997), mestrado em Engenharia de Transportes (2000) e doutorado em Otimização em Engenharia de Sistemas (2007), todos pela UFRJ / COPPE. É pesquisador no CEPEL - Centro de Pesquisas de Energia Elétrica, do grupo ELETROBRÁS, e professor adjunto da UERJ - Universidade do Estado do Rio de Janeiro, no departamento de Estatística. Tem grande experiência na área de programação matemática e otimização, especialmente aplicada em problemas de planejamento e operação de Sistemas de Energia Elétrica.

Carmen Lucia Tancredo Borges: possui graduação em Engenharia Elétrica pela Universidade do Estado do Rio de Janeiro (1984), mestrado em Engenharia Elétrica pela Universidade Federal do Rio de Janeiro (1991) e doutorado em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (1998). Atualmente é Professor Associado III da Universidade Federal do Rio de Janeiro, atuando na pós-graduação da COPPE e na graduação da Escola Politécnica. Tem experiência de ensino, pesquisa e desenvolvimento de projetos na área de Engenharia Elétrica, com ênfase em Sistemas de Potência. Atua principalmente nas seguintes áreas: Análise, Simulação e Otimização, Confiabilidade, Geração Distribuída, Fontes Alternativas, Sistemas Inteligentes e Processamento de Alto Desempenho. Foi Coordenadora Acadêmica do Programa de Engenharia Elétrica da COPPE/UFRJ em 2007 e Vice-Coordenadora do Departamento de Engenharia Elétrica da Escola Politécnica/UFRJ de 2004 a 2011. Atualmente é Presidente do Capítulo PES-Power & Energy Systems da Seção Rio do IEEE com gestão de 2012 a 2014