

# DESPACHO COM RESTRIÇÕES DE SEGURANÇA E CONTROLE CORRETIVO – UMA IMPLEMENTAÇÃO PARALELA

Mario J. Teixeira, Herminio J.C.P. Pinto e Mário V.F. Pereira  
CEPEL – Centro de Pesquisas de Energia Elétrica  
C.P. 2754, 20001 – Rio de Janeiro, RJ

## RESUMO

Este artigo descreve a implementação em máquina paralela do despacho de energia elétrica com restrições de segurança e controle corretivo, que é um problema de otimização utilizado no planejamento da operação do sistema elétrico de potência. A metodologia de solução baseia-se no emprego do algoritmo de decomposição de Benders que permite a paralelização de grande parte da solução. Esta implementação mostra a viabilidade da adaptação para máquinas paralelas de outros problemas de otimização que utilizem a mesma abordagem. A versão paralela desta aplicação foi inicialmente simulada numa máquina sequencial e em seguida transportada para uma máquina paralela. Os resultados do simulador e implementação foram coerentes e mostraram uma aceleração significativa pelo emprego de paralelismo, tendo em vista o fraco acoplamento entre as partes da solução. São também sugeridas maneiras de aumentar a eficiência da solução.

## ABSTRACT

This paper describes the implementation in a parallel machine of the security constrained dispatch with corrective rescheduling, which is an optimization problem used in electric power systems operation planning. The solution methodology is based on the use of Benders decomposition algorithm that allows a major part of the solution to be executed in parallel. This implementation shows that it is possible to adapt other optimization problems to parallel machines using the same approach. The parallel version of this application was initially simulated on a sequential machine and afterwards implemented in a parallel machine. Results from the simulation and implementation were coherent and showed a remarkable speed-up by the use of parallelism, since most parts of the solution are loosely coupled. Ways to improve the performance of the solution are also suggested.

## 1. INTRODUÇÃO

O objetivo do despacho de potência ativa com restrições de segurança e controle corretivo é minimizar o custo de operação do sistema elétrico, mantendo-o dentro dos padrões de segurança especificados. O conjunto de restrições a serem atendidas é composto pelas equações de atendimento à demanda de energia (fluxo de potência) e pelas restrições de operação (limites de geração e carregamento dos equipamentos). Este conjunto de restrições deve ser satisfeito mesmo quando ocorrem desligamentos forçados de circuitos (contingências).

O problema definido acima pode ser modelado como um problema de programação linear que possui centenas de milhares de restrições e variáveis, mesmo para sistemas elétricos de pequenas dimensões. Pode-se porém resolvê-lo através de um método de decomposição, conhecido como Método de Benders, que permite o desacoplamento do problema original em subproblemas de menores dimensões, associados a cada configuração do sistema (antes e após cada contingên-

cia). Este esquema adicionalmente se presta a uma paralelização dos processos que fazem a análise do conjunto de contingências, conforme descrito em [1].

Do ponto de vista de análise de paralelismo, este tipo de problema se enquadra na classe "fracamente acoplada". Vários outros problemas, como por exemplo a análise de confiabilidade por métodos de Monte-Carlo e o estudo da expansão do sistema de transmissão, pertencem a esta classe. Podem-se empregar várias arquiteturas para resolver problemas desta classe sem alterar substancialmente o desempenho dos algoritmos, pois a quantidade de informações trocadas entre processos é pequena. Entretanto, arquiteturas do tipo memória compartilhada permitem uma maior flexibilidade na programação e algumas vezes a obtenção de um código mais elegante.

Para paralelizar problemas desta classe, o CEPEL tem utilizado o seguinte procedimento :

- . simulação da versão paralela em máquina sequencial
- . transporte e execução do programa em máquina paralela.

Esta abordagem é justificada pelo fato das aplicações visadas já estarem implantadas em máquina sequencial (VAX 11/780) e pelas facilidades oferecidas por este ambiente para depuração de programas. O hardware com capacidade de processamento paralelo utilizado pelo CEPEL é o Processador Preferencial (PP) desenvolvido pelo CPqD da Telebrás, com 16 UCP's baseadas no iAPX286/287. O CEPEL desenvolveu software que permite o carregamento ordenado de sistemas operacionais compatíveis com o MS-DOS em cada placa processadora, bem como serviços básicos de comunicação e sincronização entre tarefas. A descrição completa do ambiente de desenvolvimento utilizado está contida em [2].

Este artigo descreve inicialmente o algoritmo de decomposição de Benders aplicado ao problema do despacho com restrições de segurança. Em seguida a implementação efetuada é detalhada e os resultados obtidos com o simulador e com o PP são mostrados. Finalmente, são sugeridas modificações no algoritmo visando um aumento de seu desempenho.

## 2. O ALGORITMO DE DECOMPOSIÇÃO

O despacho com restrições de segurança e controle corretivo pode ser formulado como o seguinte problema de programação matemática [3]:

$$\begin{aligned} z &= \min c^t x_0 \\ \text{s.a.:} \quad & [a_0] x_0 \leq b_0 \quad (1) \\ & \left. \begin{aligned} [a_i] x_i &\leq b_i \\ \|x_0 - x_i\| &\leq \Delta_i \end{aligned} \right\} p/i = 1, \dots, m \quad (2) \quad (3) \end{aligned}$$

onde  $x_0$  é o conjunto das variáveis de estado e de controle do sistema na configuração básica,  $x_i$  representa um conjunto análogo após a ocorrência da  $i$ -ésima contingência, e  $c^t x_0$  é o custo de operação do sistema no estado  $x_0$ .

As restrições de carga e de geração para o estado  $x_0$  e cada estado  $x_i$ , i.e., as equações de fluxo de potência, os limites impostos aos fluxos nas linhas de transmissão e os limites na potência fornecida pelos geradores, são representadas pelas restrições (1) e (2) respectivamente. Na formulação apresentada neste artigo, o sistema elétrico é representado por um modelo de fluxo de potência linearizado. As restrições (1) e (2) correspondem às seguintes forma (ver [4]):

$$\begin{aligned} B \theta + P_g &= d \\ |\gamma T^t \theta| &\leq f_{\max} \\ P_{\min} &\leq P_g \leq P_{\max} \end{aligned}$$

onde  $B$  é a matriz de susceptâncias,  $\theta$  é o vetor dos ângulos das tensões,  $P_g$  é o vetor dos níveis de geração

ativa,  $d$  é o vetor dos níveis de carga,  $T^t$  matriz de incidência transposta,  $\gamma$  matriz diagonal de susceptância de cada linha de transmissão,  $f_{\max}$  vetor dos fluxos máximos de energia em cada ramo da rede elétrica e  $P_{\min}$  e  $P_{\max}$  respectivamente os limites mínimo e máximo de potência ativa fornecida pelos geradores.  $P_g$  e  $\theta$  constituem o conjunto de variáveis  $x_0$  e são denominados ponto de operação do sistema.

A capacidade corretiva do sistema após a ocorrência da contingência, i.e., sua capacidade em reprogramar os valores de potência sendo fornecida pelos geradores, pode ser expressa através do conjunto de restrições (3), onde  $\Delta_i$  é o vetor do desvio máximo das variáveis de controle ( $P_g$ ).

A solução do problema acima formulado é baseada em técnicas de programação linear junto com decomposição de Benders. Este problema pode ser interpretado como um processo de decisão em dois estágios [3]:

- No primeiro, calcula-se um ponto de operação  $x_0$  para o problema de "despacho econômico":

$$\begin{aligned} z &= \min c^t x_0 \\ \text{s.a.:} \quad & [a_0] x_0 \leq b_0. \end{aligned}$$

- No segundo, dado um ponto de operação  $x_0$ , calcula-se um ponto de operação  $x_i$  que satisfaça às restrições de segurança (2) e às restrições de acoplamento (3) para cada configuração pós-contingência  $i \in [1, m]$ .

A função objetivo consiste em minimizar o custo de operação  $z = c^t x_0$  enquanto se garante a viabilidade do segundo estágio. Este processo de decisão é ilustrado na Figura 1.

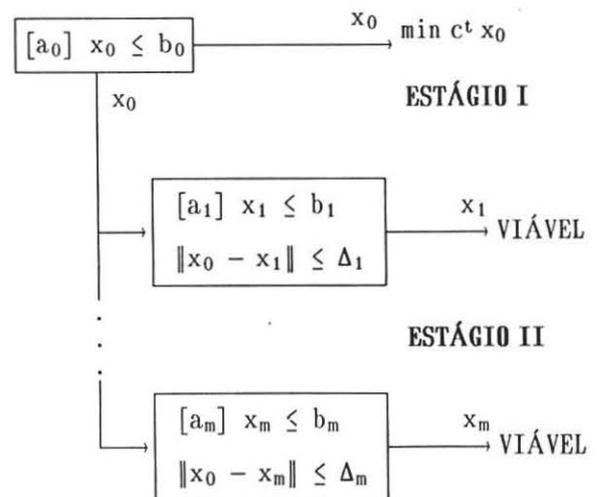


Figura 1 – Processo de decisão em dois estágios

O esquema de decomposição de Benders consiste na obtenção de uma aproximação do grau de inviabilidade

do ponto de operação  $x_0$  para cada configuração pós-contingência. Este grau de inviabilidade é representado por uma restrição linear denominada **corte de Benders**, que é incorporada ao problema de despacho econômico do primeiro estágio, conforme a figura 2 [4].

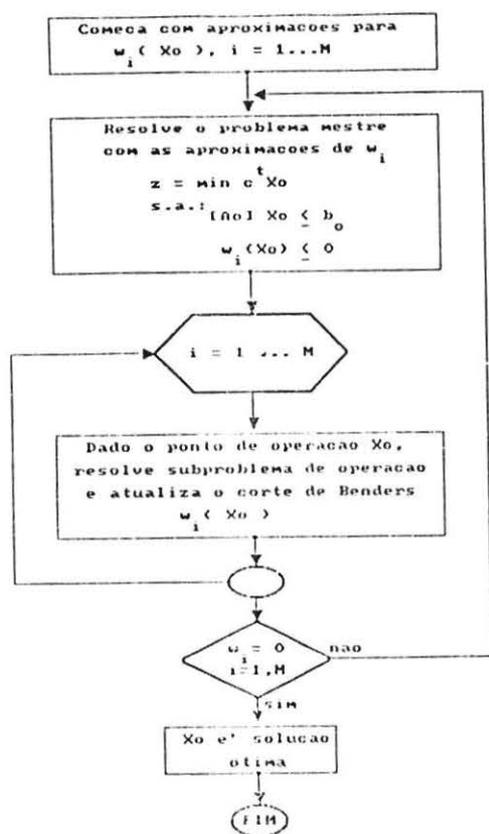


Figura 2 – Esquema de decomposição de Benders

### 3. IMPLEMENTAÇÃO E RESULTADOS OBTIDOS

Um aspecto que pode ser destacado na figura 2 é a possibilidade da solução de cada subproblema de operação ser calculada independentemente das outras. Além disso, a quantidade de informação trocada entre os subproblemas e o módulo de despacho econômico é pequena em relação ao processamento envolvido, o que caracteriza um problema fracamente acoplado.

Neste tipo de problema, a escolha de diferentes modelos de programação (memória compartilhada ou troca de mensagens) não possui um impacto determinante no desempenho final do algoritmo. Entretanto, um modelo de memória comum pode permitir uma flexibilidade maior na programação, como ilustrado pela implementação descrita a seguir. Supõe-se no que se segue uma arquitetura hardware em que cada processador possui uma memória local e todos podem acessar uma memória compartilhada, o que é o caso do Processador Preferencial.

Na implementação efetuada, um dos processadores (chamado mestre) faz a seleção das contingências correspondente a cada subproblema de operação e

resolve o despacho econômico. Os demais processadores (chamados escravos) calculam os cortes de Benders correspondentes às configurações pós-contingência do sistema, conforme mostrado na figura 3.

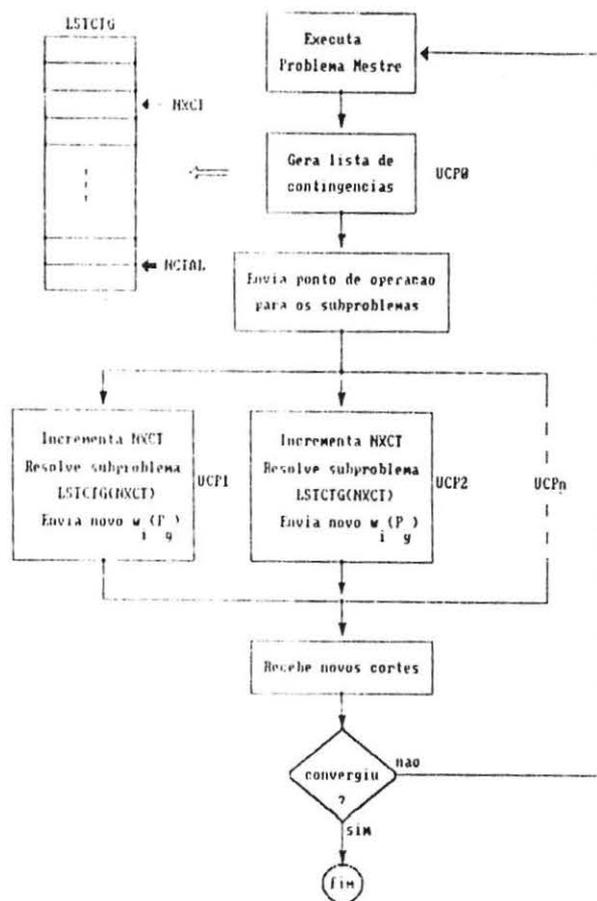


Figura 3 – Particionamento da solução

A atribuição do despacho econômico a um processador deveu-se a limitações práticas de memória. Em sua versão sequencial, o programa ocupa mais que os 512 kbytes disponíveis em cada placa processadora do PP. Optou-se portanto pela criação de 2 programas diferentes: um programa mestre ocupando 470 kbytes e um programa escravo ocupando 350 kbytes.

Os códigos dos programas e os dados do sistema elétrico (topologia da rede, parâmetros, etc.) são guardados em memória local de cada processador. O ponto de operação calculado pelo mestre é enviado a todos os escravos através de uma área de comunicação localizada em memória compartilhada. Uma estrutura FIFO, também localizada em memória compartilhada, é utilizada pelos escravos para enviarem os cortes de Benders para o mestre.

A primeira etapa do trabalho consistiu em, a partir de uma versão sequencial do programa, simular a solução paralelizada num VAX 11/780. Nesta etapa, o programa foi dividido levando-se em conta as características

do hardware do PP, de tal modo a proporcionar uma migração imediata entre os dois ambientes. As primitivas para sincronização e comunicação disponíveis no VAX/VMS foram utilizadas diretamente ou adaptadas ao simulador.

Foi criado um mecanismo de contagem de tempo, que é ativado logo após a criação dos processos escravos e é interrompido sempre que há troca de informação entre o mestre e os escravos. Esta característica do simulador está associada à dificuldade de se modelar o tempo gasto em comunicação entre os processadores. Cabe lembrar que o objetivo principal deste simulador é permitir a depuração da versão paralela de uma aplicação qualquer fracamente acoplada no ambiente de desenvolvimento onde a versão sequencial está implantada. Uma descrição detalhada deste simulador encontra-se em [2].

As estruturas dos programas mestre e escravo são mostradas na figura 4 em FORTRAN. O mestre é encarregado da leitura dos dados em disco, assim como da inicialização das variáveis utilizadas por ambos os programas mestre e escravo. A rotina CRIA tem no mestre a função de criar tantos processos escravos quanto forem especificados em seu argumento. Esta rotina, quando chamada por um processo escravo, faz com que este fique esperando a criação dos demais escravos.

A rotina ENVIA\_PO tem duas funções: a primeira é passar, na primeira vez em que é chamada, o conteúdo de todas as variáveis que foram inicializadas pelo mestre mas que serão utilizadas também pelos escravos. A segunda é, a cada iteração, passar o ponto de operação (potência gerada e ângulo das tensões) para os escravos. Todos os dados enviados pela rotina ENVIA\_PO são lidos pelos escravos através da rotina REC\_PO.

A lista de tarefas, mostrada na figura 3, é gerada pelo mestre através da rotina DISTR\_TAREFAS e enviada aos escravos via memória comum, isto é, LSTCTG, NCTAL e NXCT são variáveis acessíveis a todos os processos. Cada processo escravo, ao se executar, deve procurar qual a contingência a ser analisada e incrementar o apontador para a próxima contingência a ser analisada (NXCT), o que é feito sob seção crítica (rotinas LOCK e UNLOCK). Caso a lista de tarefas tenha terminado, ele deverá voltar para a rotina REC\_PO e aguardar até que um novo ponto de operação esteja disponível ou que a sinalização de fim de execução seja ativada. Caso contrário, ele deverá resolver o subproblema de operação indicado pela variável LSTCTG.

```

PROGRAM MESTRE
.
. leitura dos dados
. inicialização
.
call CRIA( NM_PRC )

do while( .not. CONVERGIU )
call DISTR_TAREFAS
call ENVIA_PO

NM_TERM = 0
do while(NM_TERM.ne.NM_PRC)
call REC_CORTE
end do
.
. resolve despacho econômico
. calcula novo ponto de operação
.
end do

call FIM
stop

```

```

PROGRAM ESCRAVO

call CRIA

do while( .TRUE. )
call REC_PO
call LOCK
NX = NXCT
NXCT = NXCT + 1
call UNLOCK

do while( NX .le. NCTAL )
NSO = LSTCTG( NX )
.
. resolve subproblema
. de operação NSO
.
call ENV_CORTE
call LOCK
NX = NXCT
NXCT = NXCT + 1
call UNLOCK
end do
end do

stop

```

Figura 4 – Estrutura dos programas mestre e escravo

Os cortes são gerados pelos escravos de maneira assíncrona, e suas incorporações pelo mestre podem se dar em vários pontos de sua execução. Uma primeira possibilidade, cujos resultados são mostrados neste artigo, consiste em o mestre aguardar o término de todos os escravos para só então determinar se o algoritmo convergiu e, se não, calcular um novo ponto de operação. O mestre lê os cortes transmitidos pelos escravos (rotina ENV\_CORTE) através de chamada à rotina REC\_CORTE. Caso o mestre verifique que a solução convergiu, ele chama a rotina FIM, cuja função

é ativar a sinalização de fim de execução para os escravos.

Para ilustrar a metodologia foi utilizado um caso correspondente a um sub-sistema do Sul-Sudeste Brasileiro. Este sistema possui 504 barras, 880 linhas de transmissão e transformadores, 72 geradores controláveis com capacidade corretiva que permite a um gerador variar rapidamente a sua potência gerada em até 20% da sua capacidade máxima, de maneira a atender os conjuntos de restrições (2) e (3) do item 2. A função objetivo foi especificada como sendo o desvio mínimo em relação ao ponto de operação ótimo obtido sem levar em contas as restrições de segurança. Foram consideradas 718 contingências simples de linha.

O problema de despacho com restrições de segurança e controle corretivo para este caso foi resolvido, tanto no VAX quanto no PP, com o número de escravos variando de 1 a 15. O "speed-up" obtido é mostrado na figura 5. O resultado do simulador é um pouco otimista, o que se pode atribuir ao fato dele não computar o tempo gasto com a comunicação entre mestre e escravos. Resultados coerentes entre o simulador e o PP também foram obtidos com outros sistemas. Em termos absolutos, foram necessários 6 processadores escravos com co-processador 80287 para executar este programa em tempo de CPU inferior ao de um VAX 11/780 com acelerador de ponto flutuante.

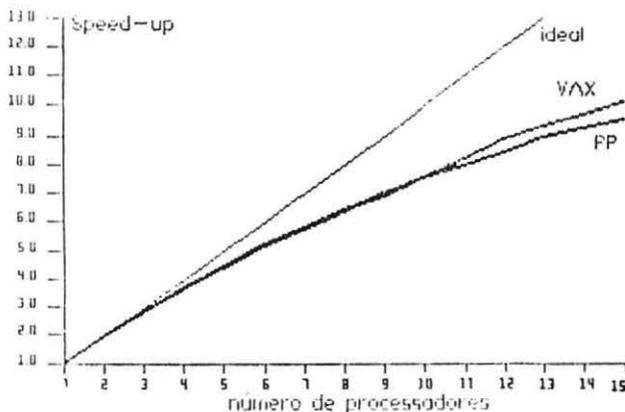


Figura 5 - "Speed-up" obtido

#### 4. CONCLUSÕES E EXTENSÕES

A parte mais demorada da paralelização do problema consistiu na identificação e separação de variáveis entre pertencentes ao mestre, ao escravo e compartilhadas em memória global, e na implementação das estruturas de dados necessárias ao processamento distribuído.

Na parte de depuração o simulador foi de extrema utilidade, permitindo descobrir no ambiente original do programa problemas de sincronização típicos de software paralelo, que seriam mais difíceis de serem encontrados uma vez feito o transporte para o PP,

devido à mudança de ambiente de programação e à inexistência de um depurador distribuído. O transporte para o PP foi feito em uma semana de trabalho, o que mostra o acerto do método empregado.

Para melhorar o desempenho do algoritmo, está-se implantando uma versão do mestre que incorpora os cortes gerados pelos escravos assim que eles são gerados. Isto é possível pois o método de solução utilizado pelo mestre é o dual-simplex, que permite a incorporação de restrições suplementares ao longo de sua execução. Esta nova estratégia certamente melhorará o desempenho do algoritmo, principalmente em casos menores, onde o tempo de execução do mestre é grande em relação ao tempo de análise de 1 contingência pelos escravos. Outra estratégia possível seria a interrupção dos escravos assim que o mestre gerasse um novo ponto de operação; neste caso uma previsão teórica de melhoria do desempenho do algoritmo é mais difícil, sendo necessária experimentação para otimizar este procedimento.

#### 5. REFERÊNCIAS

- [1] M.V.F. Pereira, M.J. Teixeira e L.A. Terry, "Aplicações de processamento paralelo em sistemas elétricos de potência", 1º Cong. Bras. Proc. Paralelo, Gramado, RS, maio 1987.
- [2] M.J. Teixeira, M.V.F. Pereira, L.A. Terry e H.J.C.P. Pinto, "Ambiente para desenvolvimento de programas paralelos fracamente acoplados", 7º Cong. SBA, agosto 1988.
- [3] M.V.F. Pereira, A. Monticelli e L.M.V.G. Pinto, "Security constrained dispatch with corrective rescheduling", Proc. IFAC Elec. En. Syst., Rio de Janeiro, 1985.
- [4] S. Granville, H.J.C.P. Pinto e M.V.F. Pereira, "Despacho com restrição de segurança e controle corretivo - modelo linearizado", rel. técnico Cepel, 1987.