
UM MÉTODO EFICIENTE PARA RESOLUÇÃO DO PROBLEMA DE DESPACHO ÓTIMO DE REATIVOS COM CONTROLE DISCRETO DO TAP DOS TRANSFORMADORES

Edilaine M. Soler*
edilaine@fc.unesp.br

Eduardo N. Asada†
easada@sc.usp.br

Geraldo R. M. da Costa†
geraldo@sc.usp.br

*Universidade Estadual Paulista Julio de Mesquita Filho
Faculdade de Ciências – Departamento de Matemática
Av. Engenheiro Edmundo Carrijo Coube, 14-01
CEP 17.033-360 - Bauru - SP

†Universidade São Paulo
Escola de Engenharia de São Carlos – Departamento de Engenharia Elétrica
Av. Trabalhador São-carlense, 400
CEP 13.566-590 – São Carlos – SP

ABSTRACT

An Efficient Method for Solving the Problem of Optimal Reactive Dispatch with Discrete Control of the Tap Transformers

The Optimal Reactive Dispatch problem is a nonlinear programming problem containing continuous and discrete control variables. Owing to the difficulty caused by discrete variables, this problem is usually solved assuming that all variables are continuous. The discrete variables are rounded off to the closest discrete value and the problem is solved again. This approach may provide no optimal or may be impossible to obtain feasible solutions. This paper presents an efficient handling of discrete variables by auxiliary function so that the problem becomes continuous and differentiable. Simulations with IEEE test systems were performed showing the efficiency of the proposed approach.

KEYWORDS: Optimal Power Flow, Newton method, Non-linear programming, Interior point method

RESUMO

O problema de Despacho Ótimo de Reativos é um problema de programação não linear com variáveis de controle discretas e contínuas. Devido à dificuldade de solução imposta pelas variáveis discretas o problema é normalmente resolvido considerando-se todas as variáveis como contínuas. Após a solução arredondam-se as variáveis discretas para os seus valores mais próximos e resolve-se novamente o problema. Esta abordagem pode levar a soluções não ótimas ou pode ser impossível obter soluções factíveis. Este artigo apresenta uma abordagem eficiente, no tratamento das variáveis discretas, via função auxiliar tornando o problema contínuo e diferenciável. Simulações com sistemas testes do IEEE foram realizados mostrando a eficiência da abordagem proposta.

PALAVRAS-CHAVE: Fluxo de Potência Ótimo, método de Newton, programação não linear, método dos pontos interiores

Artigo submetido em 29/08/2011 (Id.: 1372)
Revisado em 04/10/2011, 29/01/2012, 03/02/2012
Aceito sob recomendação do Editor Associado Prof. Julio Cesar Stacchini Souza

1 INTRODUÇÃO

O problema de Despacho Ótimo de Reativos (DOR) é um caso particular do problema de Fluxo de Potência Ótimo (FPO) no qual os controles relacionados com a potência ativa são fixados e a otimização somente considera as variáveis relacionadas com a potência reativa, isto é, magnitude da tensão nos geradores, compensadores síncronos, *taps* dos transformadores e bancos *shunt*. Porém, neste artigo é considerado somente a variável discreta *taps* dos transformadores. O FPO é um problema de otimização não linear, não convexo e de grande porte. As restrições de igualdade são as equações não lineares de balanço de potência ativa e reativa e as restrições de desigualdade correspondem aos limites de segurança e operativos dos equipamentos. Assim, o problema DOR é modelado como um problema de programação não linear, não convexo, com restrições de igualdade e desigualdade, e com variáveis contínuas e discretas. Alguns autores abordaram o problema. Liu et. al (1992) aplicaram a função penalidade para ajustar as variáveis discretas. Na linha metaheurística Bakirtzis et. al (2002) apresentaram um Algoritmo Genético Modificado. Liu et. al (2002) incorporaram a função penalidade a um algoritmo de pontos interiores primal-dual. Adibi et. al (2003) utilizaram a função Lagrangiana Aumentada Barreira Modificada considerando no início todas as variáveis contínuas. Ding e Wang (2004) aplicaram o método de Planos de Corte com Pontos Interiores. Liu et al. (2009) apresentaram uma abordagem com o método de Planos de Cortes Ponto Interior Melhorado, considerando inicialmente todas as variáveis como contínuas. Capitanescu e Wehenkel (2010) consideraram todas as variáveis contínuas e sensibilidade para ajustar as discretas. A presença de variáveis discretas em problemas de programação não linear dificulta sua resolução, por este motivo a maioria dos trabalhos considera todas as variáveis do problema como contínuas. Estas formulações estão longe da realidade de um sistema elétrico, pois algumas variáveis somente podem ser ajustadas por passos discretos.

Um problema de programação não linear com variáveis discretas e contínuas é normalmente resolvido através de métodos de programação inteira-mista, porém a complexidade computacional da abordagem combinatorial para soluções rigorosas de problemas deste tipo tende a aumentar exponencialmente com o número de variáveis discretas e, portanto, torna-se de difícil aplicação para problemas que envolvem um grande número de variáveis e restrições. Métodos como *Outer Approximation* (Leyffer (1993)), Decomposição de *Benders* (Geoffrion (1972)), *Branch-and-Bound* (Gupta e Ravindran (1985)) e Algoritmos de Planos de Corte (Westerlund e Petersson (1995)) são métodos que requerem grande esforço computacional quando aplicados a problemas de grande porte.

Devido à dificuldade de solução imposta pelas variáveis discretas em problemas de programação não linear, um método muito utilizado no tratamento das variáveis discretas no problema de DOR é o arredondamento. Inicialmente resolve-se o problema assumindo que todas as variáveis sejam contínuas, em seguida, as variáveis discretas são arredondadas e fixadas nos valores discretos mais próximos e o problema é novamente resolvido. Este método pode levar a soluções não-ótimas e além disso, o problema obtido ao se fixar valores discretos para o *tap* pode ser infactível o que torna impossível a obtenção de uma solução.

Assim, devido a importância do problema de DOR e a dificuldade de resolvê-lo, propõe-se neste artigo um método de solução para ajustar de maneira ótima as variáveis contínuas e discretas deste problema.

Neste artigo uma função auxiliar é agregada à função objetivo para tratar as variáveis discretas. Ao incorporar esta função à função objetivo as variáveis discretas são tratadas como contínuas e obtém-se uma sequência de problemas de Programação Não Linear (PNL) com variáveis apenas contínuas cujas soluções convergem para a solução do problema original, que contém variáveis discretas e contínuas. Os PNLs obtidos são resolvido pelo Método de Pontos Interiores com Filtro proposto em Wächter e Biegler (2006).

Na Seção 2 é apresentada a formulação matemática do problema de DOR com variáveis discretas e contínuas. Na Seção 3 é apresentado um Método de Solução para ajustar as variáveis discretas e contínuas do problema. Na Seção 4 são apresentados os testes computacionais realizados e na Seção 5, estão as conclusões.

2 O PROBLEMA DOR

O problema DOR é modelado matematicamente como um problema de programação não linear com variáveis discretas e contínuas e pode ser representado como:

$$\begin{aligned} & \text{Min} f(V, \theta, t) \\ & \text{s.a.} : \begin{cases} \Delta P_i(V, \theta, t) = 0, i = 1, 2, \dots, NBCCR \\ \Delta Q_j(V, \theta, t) = 0, j = 1, 2, \dots, NBC \\ Q_k \leq Q_k(V, \theta, t) \leq \bar{Q}_k, k = 1, 2, \dots, NBCR \\ \underline{V}_p \leq V_p \leq \bar{V}_p, p = 1, 2, \dots, NB \\ t_l \in D_{t_l}, l = 1, 2, \dots, NT \end{cases} \end{aligned} \quad (1)$$

(P1)

em que:

NBCCR é o número de barras de carga e de controle de reativos;

NBC é o número de barras de carga;

NBCR é o número de barras de controle de reativo;

NB é o número de barras do sistema elétrico;

NT é o número de transformadores com controle de tap;

$\underline{Q}_k, \overline{Q}_k$ são os limites mínimos e máximos de geração de potência reativa;

$\underline{V}_p, \overline{V}_p$ são os limites mínimos e máximos das magnitudes das tensões;

D_{t_l} é o conjunto de valores discretos que o tap do transformador l pode assumir.

Variáveis:

$V = (V_1, V_2, \dots, V_{NB})^t$ é o vetor das magnitudes de tensão nas barras 1,2,...,NB;

$\theta = (\theta_1, \theta_2, \dots, \theta_{NB})^t$ é o vetor dos ângulos de tensão nas barras 1,2,...,NB;

$t = (t_1, t_2, \dots, t_{NT})^t$ é o vetor dos taps dos transformadores 1,2,...,NT;

A seguir são descritas as funções que aparecem no Problema (P1), para isso considere:

Ω o conjunto de todas as linhas de transmissão;

Ω_k o conjunto de todas as barras vizinhas à barra k ;

$g_{km}, b_{km}, b_{km}^{sh}$ a condutância e as susceptâncias da linha km;

P_k^G, P_k^C as potências ativas geradas e consumidas, respectivamente;

Q_k^G, Q_k^C as potências reativas geradas e consumidas, respectivamente.

Com isso tem-se:

- A função escalar

$$f(V, \theta, t) \quad (2)$$

representa as perdas de potência ativa na transmissão e é dada por:

$$f(V, \theta, t) = \sum_{km \in \Omega} g_{km}(V_k^2 + V_m^2 - 2V_k V_m \cos \theta_{km}) \quad (3)$$

$$\theta_{km} = \theta_k - \theta_m \quad (4)$$

- As equações de balanço do sistema elétrico são dadas por:

- Balanço de potência ativa pra as barras de carga e de controle reativo:

$$\begin{aligned} \Delta P_i(V, \theta, t) &= P_i^G - P_i^C - \sum_{m \in \Omega_i} P_{im}(V, \theta, t) = 0; \\ P_{im} &= (t_{im} V_i)^2 g_{im} - (t_{im} V_i) V_m (g_{im} \cos \theta_{im} + b_{im} \text{sen} \theta_{im}). \end{aligned} \quad (5)$$

- Balanço de potência reativa para as barras de carga:

-

$$\begin{aligned} \Delta Q_j(V, \theta, t) &= Q_j^G - Q_j^C - \sum_{m \in \Omega_j} Q_{jm}(V, \theta, t) = 0; \\ Q_{jm} &= -(t_{jm} V_j)^2 (b_{jm} + b_{jm}^{sh}) + \\ &+ (t_{jm} V_j) V_m (b_{jm} \cos \theta_{jm} - g_{jm} \text{sen} \theta_{jm}). \end{aligned} \quad (6)$$

- Geração de potência reativa injetada nas barras de controle de reativo:

$$\begin{aligned} Q_k &= \sum_{m \in \Omega_k} \left[-(tV_k)^2 (b_{km} + b_{km}^{sh}) + \right. \\ &\left. + (tV_k) V_m (b_{km} \cos \theta_{km} - g_{km} \text{sen} \theta_{km}) \right]. \end{aligned} \quad (7)$$

O modelo matemático do problema de DOR, apresentado em (P1), é de difícil resolução, pois é um problema de programação não linear, com funções não convexas, e com variáveis discretas e contínuas. A função objetivo em (P1) representa as perdas ativas nas linhas de transmissão, essa função é não separável e não permite simplificações, segundo Monticelli e Liu (1992) este fato dificulta ainda mais a resolução do problema.

Para resolver o problema de DOR dado por (P1) é proposto, neste trabalho, o Algoritmo de Pontos Interiores Modificado-Discreto (Algoritmo PIMD). Este método de solução é descrito a seguir.

3 MÉTODO DE SOLUÇÃO

Baseado nos trabalhos de Davydov e Sigal (1972) e Fu et. al (1991), é proposto um método para ajustar as variáveis discretas em problemas de otimização com não linearidades. O método proposto consiste em resolver uma sequência de Problemas de Programação Não Linear com variáveis contínuas somente (PNL) cujas soluções convergem para a solução do

Problema de Programação Não Linear com Variáveis Discretas e Contínuas (PNLDC). Cada PNL é obtido incorporando à função objetivo do problema original uma função que penaliza a função objetivo quando as variáveis discretas assumem valores não discretos (Soler e da Costa (2010)). Ao incorporar esta função auxiliar na função objetivo as variáveis discretas podem ser tratadas como contínuas. Os PNLs obtidos são resolvidos pelo Método de Pontos Interiores com Filtro proposto em Wächter e Biegler (2006). A seguir é detalhado o método proposto.

3.1 PNL equivalente ao PNLDC

Considere um problema de programação não linear com variáveis discretas e contínuas dado por:

$$\begin{aligned} & \text{Min}f(x, y) \\ \text{s.a : } & \begin{cases} h(x, y) = 0 \\ g(x, y) \geq 0 \\ \underline{x} \leq x \leq \bar{x} \\ y_i \in D_{y_i}, i = 1, 2, \dots, n_y \end{cases} \end{aligned} \quad (8)$$

em que $x = (x_1, x_2, \dots, x_{n_x})$ e $y = (y_1, y_2, \dots, y_{n_y})$ são variáveis de decisão, D_{y_i} é o conjunto de valores discretos para a variável y_i , para $i = 1, 2, \dots, n_y$, com $f(x, y)$, $h(x, y) = (h_1(x, y), h_2(x, y), \dots, h_m(x, y))$ e $g(x, y) = (g_1(x, y), g_2(x, y), \dots, g_p(x, y))$ funções de classe C^2 . Os vetores $\underline{x} \in \mathbb{R}^{n_x}$ e $\bar{x} \in \mathbb{R}^{n_x}$ indicam os limites inferior e superior da variável x .

Desenvolveu-se uma função auxiliar dada por:

$$\Phi(y) = \gamma \sum_{i=1}^{n_y} \left[\text{sen} \left(\frac{y_i}{s_i^u - s_i^l} \pi + \alpha \right) \right]^2 \quad (9)$$

em que:

$\gamma > 0$ é o fator de ajuste que determina a amplitude da função;

s_i^l é o valor discreto mais próximo inferiormente de y_i ;

s_i^u é o valor discreto mais próximo superiormente de y_i ;

α é uma constante tal que $0 \leq \alpha < \pi$ escolhida de modo que a função $\Phi(y)$ se anule somente nos valores discretos de y .

Assim, tem-se que:

$$\Phi(y) = \begin{cases} 0, \text{ se } y_i \in D_{y_i}, i = 1, 2, \dots, n_y \\ \delta > 0, \text{ caso contrário} \end{cases} \quad (10)$$

ou seja, $\Phi(y)$ assume valores positivos somente se y não assumir valores discretos.

Assim, para valores adequados de γ encontrar uma solução ótima para o Problema (P2) é equivalente a resolver o Problema Modificado (P3), que contém apenas variáveis contínuas:

$$\begin{aligned} & \text{Min}f(x, y) + \Phi(y) \\ \text{s.a : } & \begin{cases} h(x, y) = 0 \\ g(x, y) \geq 0 \\ \underline{x} \leq x \leq \bar{x} \\ \underline{y} \leq y \leq \bar{y} \end{cases} \end{aligned} \quad (11)$$

em que $\underline{y} = (y_1, y_2, \dots, y_{n_y})$, com $y_i = \min D_{y_i}$ para $i = 1, 2, \dots, n_y$, e $\bar{y} = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_{n_y})$, com $\bar{y}_i = \max D_{y_i}$, para $i = 1, 2, \dots, n_y$.

Referimo-nos à função objetivo do Problema Modificado (P3) dada por $f(x, y) + \Phi(y)$ como função objetivo aumentada, pois esta consiste da função objetivo do problema original acrescida da função auxiliar $\Phi(y)$.

O Problema Modificado (P3) é resolvido pelo Método de Pontos Interiores com Filtro proposto em Wächter e Biegler (2006). Este método é descrito a seguir.

3.2 Método de Pontos Interiores com Filtro

O Problema Modificado (P3) pode ser escrito da forma dada em (P4) em que $z = (x, y)$ e $q(z) = f(x, y) + \Phi(y)$.

$$\begin{aligned} & \text{Min}q(z) \\ \text{s.a : } & \begin{cases} h_j(z) = 0, j = 1, 2, \dots, m \\ g_i(z) \geq 0, i = 1, 2, \dots, p \\ \underline{z}_l \leq z_l \leq \bar{z}_l, l = 1, 2, \dots, n \end{cases} \end{aligned} \quad (12)$$

Acrescentando variáveis de folga e excesso positivas o Problema (P4) é transformado no Problema equivalente (P5).

$$\begin{aligned} & \text{Min}q(z) \\ \text{s.a : } & \begin{cases} h_j(z) = 0, j = 1, 2, \dots, m \\ g_i(z) - s_{1i} = 0, i = 1, 2, \dots, p \\ -z_l + \bar{z}_l - s_{2l} = 0, l = 1, 2, \dots, n \\ z_l - \underline{z}_l - s_{3l} = 0 \\ s_{1i}, s_{2l}, s_{3l} \geq 0 \end{cases} \end{aligned} \quad (13)$$

Os vetores $s_1 = (s_{11}, s_{12}, \dots, s_{1p})$, $s_2 = (s_{21}, s_{22}, \dots, s_{2n})$ e $s_3 = (s_{31}, s_{32}, \dots, s_{3n})$ são denominados vetores auxiliares.

As restrições de não negatividade das variáveis de folga e excesso são tratadas pela Função Barreira, assim, essas restrições são incorporadas na função objetivo do problema pela Função Barreira Logarítmica, obtendo-se o Problema Barreira (P6), que possui somente restrições de igualdade.

$$\begin{aligned} \text{Min} \phi_{\mu}(z, s_1, s_2, s_3) &= \\ &= q(z) - \mu \left(\sum_{i=1}^p \ln(s_{1i}) + \sum_{l=1}^n \ln(s_{2l}) + \sum_{l=1}^n \ln(s_{3l}) \right) \\ \text{s.a. : } &\begin{cases} h_j(z) = 0, j = 1, 2, \dots, m \\ g_i(z) - s_{1i} = 0, i = 1, 2, \dots, p \\ -z_l + \bar{z}_l - s_{2l} = 0, l = 1, 2, \dots, n \\ z_l - \underline{z}_l - s_{3l} = 0 \end{cases} \end{aligned} \quad (14)$$

em que $\mu > 0$ é o parâmetro de barreira.

A sequência de parâmetros $\mu_j, j = 1, 2, 3, \dots$ gera uma sequência de Problemas Barreira (P6) e uma sequência de soluções desses problemas $z^*(\mu_j)$. Se $\mu_j \rightarrow 0$ então $z^*(\mu_j) \rightarrow z^*$, em que z^* é um ótimo local do Problema (P4).

O Método de Pontos Interiores com Filtro consiste em encontrar uma solução aproximada para o problema (P6), com um tolerância $\in_{tol}^{\mu_j}$, para um valor fixo $\mu = \mu_j$. Esse parâmetro é decrescido e obtém-se μ_{j+1} , o problema barreira com $\mu = \mu_{j+1}$ é resolvido usando como ponto inicial a solução aproximada obtida para o problema barreira com $\mu = \mu_j$. O processo é repetido até que se obtenha uma solução para o problema (P4), com uma tolerância \in_{tol} .

Os parâmetros $\in_{tol}^{\mu_j}$ são calculados por:

$$\in_{tol}^{\mu_j} = \kappa_{\in} \mu_j \quad (15)$$

em que $\kappa_{\in} > 0$ é uma constante.

O parâmetro de barreira é decrescido pela regra proposta por Wächter e Biegler (2006) dada por:

$$\mu_{j+1} = \max \left\{ \frac{\in_{tol}}{10}, \min \{ \kappa_{\mu} \mu_j, \mu_j^{s_{\mu}} \} \right\} \quad (16)$$

com constantes $\kappa_{\mu} \in (0, 1)$, $s_{\mu} \in (1, 2)$. Na implementação utilizada são valores padrões $\kappa_{\in} = 10$, $\kappa_{\mu} = 0, 2$, $s_{\mu} = 1, 5$, $\in_{tol} = 10^{-8}$ e $\mu_0 = 0, 1$. Estes valores foram determinados por Wächter e Biegler (2006).

Para resolver o Problema Barreira (P6) para um determinado parâmetro de barreira $\mu = \mu_j$, associa-se a este problema a Função Lagrangiana dada em (17) :

$$\begin{aligned} L(z, s_1, s_2, s_3, \lambda, \pi_1, \pi_2, \pi_3) &= q(z) + \\ &- \mu \sum_{i=1}^p \ln(s_{1i}) - \mu \sum_{l=1}^n \ln(s_{2l}) + \\ &- \mu \sum_{l=1}^n \ln(s_{3l}) - \sum_{j=1}^m \lambda_j h_j(z) - \sum_{i=1}^p \pi_{1i} (g_i(z) - s_{1i}) + \\ &- \sum_{l=1}^n \pi_{2l} (-z_l + \bar{z}_l - s_{2l}) - \sum_{l=1}^n \pi_{3l} (z_l - \underline{z}_l - s_{3l}) \end{aligned} \quad (17)$$

em que $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, $\pi_1 = (\pi_{11}, \pi_{12}, \dots, \pi_{1p})$, $\pi_2 = (\pi_{21}, \pi_{22}, \dots, \pi_{2n})$ e $\pi_3 = (\pi_{31}, \pi_{32}, \dots, \pi_{3n})$ são os multiplicadores de Lagrange.

Aplicam-se as Condições Necessárias de Primeira Ordem para a Função Lagrangiana o que resulta no sistema de equações não lineares dado por (18) .

$$\nabla L(z, s_1, s_2, s_3, \lambda, \pi_1, \pi_2, \pi_3) = 0 \quad (18)$$

As variáveis desse sistema são as componentes dos vetores $z, s_1, s_2, s_3, \lambda, \pi_1, \pi_2$ e π_3 , em que z, s_1, s_2 e s_3 são variáveis primais e, λ, π_1, π_2 e π_3 são variáveis duais. Utiliza-se o Método de Newton para resolver esse sistema não linear. Seja J a Matriz Jacobiana associada a (18) . A solução do sistema linear (19) gera direções $\Delta d = (\Delta z, \Delta s_1, \Delta s_2, \Delta s_3, \Delta \lambda, \Delta \pi_1, \Delta \pi_2, \Delta \pi_3)$ para atualização das variáveis:

$$J \Delta d = -\nabla L \quad (19)$$

Seja k o contador de iterações do Método de Newton. A partir de um ponto $z^{(k)}, s_1^{(k)}, s_2^{(k)}, s_3^{(k)}, \lambda^{(k)}, \pi_1^{(k)}, \pi_2^{(k)}, \pi_3^{(k)}$, calcula-se a matriz J e o vetor ∇L neste ponto. Ao resolver o sistema linear (19) , obtém-se as direções de busca $\Delta z^{(k)}, \Delta s_1^{(k)}, \Delta s_2^{(k)}, \Delta s_3^{(k)}, \Delta \lambda^{(k)}, \Delta \pi_1^{(k)}, \Delta \pi_2^{(k)}, \Delta \pi_3^{(k)}$. Utilizando estas as variáveis são atualizadas por:

$$\begin{aligned} z^{(k+1)} &= z^{(k)} + \alpha_k \Delta z^{(k)} \\ s_1^{(k+1)} &= s_1^{(k)} + \alpha_k \Delta s_1^{(k)} \\ s_2^{(k+1)} &= s_2^{(k)} + \alpha_k \Delta s_2^{(k)} \\ s_3^{(k+1)} &= s_3^{(k)} + \alpha_k \Delta s_3^{(k)} \\ \lambda^{(k+1)} &= \lambda^{(k)} + \alpha_k \Delta \lambda^{(k)} \\ \pi_1^{(k+1)} &= \pi_1^{(k)} + \alpha_k \Delta \pi_1^{(k)} \\ \pi_2^{(k+1)} &= \pi_2^{(k)} + \alpha_k \Delta \pi_2^{(k)} \\ \pi_3^{(k+1)} &= \pi_3^{(k)} + \alpha_k \Delta \pi_3^{(k)} \end{aligned} \quad (20)$$

em que $\alpha_k, \alpha_k^{\pi} \in (0, 1]$ são os passos utilizados para atualização das variáveis.

O passo α_k^π utilizado para atualização das variáveis π_1, π_2 e π_3 é diferente do passo utilizado para atualização das demais variáveis. Wächter e Biegler (2006) constataram por meio de testes que esta estratégia produz melhores resultados.

Como $s_1, s_2, s_3, \pi_1, \pi_2, \pi_3 > 0$ em uma solução ótima do Problema Barreira (P6), esta propriedade deve ser mantida em todas as iterações do Método de Newton. Portanto calcula-se os tamanhos máximos para os passos pela regras (21) e (22). Escolhendo $\alpha_k \in (0, \alpha_k^{\max}]$ e $\alpha_k^\pi \in (0, \alpha_k^{\pi \max}]$ tem-se $s_1^{(k+1)}, s_2^{(k+1)}, s_3^{(k+1)}, \pi_1^{(k+1)}, \pi_2^{(k+1)}, \pi_3^{(k+1)} > 0$.

$$\alpha_k^{\pi \max} = \max\{\alpha \in (0, 1] : \pi^{(k)} + \alpha \Delta \pi^{(k)} \geq (1 - \tau_j) \pi^{(k)}\} \quad (21)$$

$$\alpha_k^{\max} = \max\{\alpha \in (0, 1] : s^{(k)} + \alpha \Delta s^{(k)} \geq (1 - \tau_j) s^{(k)}\} \quad (22)$$

em que $\pi = (\pi_1, \pi_2, \pi_3)$, $s = (s_1, s_2, s_3)$, $\Delta \pi = (\Delta \pi_1, \Delta \pi_2, \Delta \pi_3)$, $\Delta s = (\Delta s_1, \Delta s_2, \Delta s_3)$ e $\tau_j \in (0, 1)$ é calculado por $\tau_j = \max\{0, 99, 1 - \mu_j\}$.

Realizando experimentos numéricos Wächter e Biegler (2006) constataram que a escolha $\alpha_k^\pi = \alpha_k^{\pi \max}$ é eficiente.

O passo $\alpha_k \in (0, \alpha_k^{\max}]$ é calculado por um procedimento de busca unidimensional explorando a sequência de valores decrescentes $\alpha_{k,l} = 2^{-l} \alpha_k^{\max}$ (com $l = 1, 2, 3, \dots$) por meio de uma variante do Método de Filtros proposto por Fletcher e Leyffer (2002). Esse procedimento é globalmente convergente (Wächter e Biegler (2005)).

A seguir é apresentado de maneira resumida o Algoritmo de Pontos Interiores com Filtro proposto em Wächter e Biegler (2006), e implementado no pacote IPOPT, o qual foi utilizado para resolver o Problema Modificado (P3). Detalhes rigorosos sobre o método e a implementação podem ser obtidos em Wächter (2002), e Wächter e Biegler (2006). Considere $s = (s_1, s_2, s_3)$ e $\pi = (\pi_1, \pi_2, \pi_3)$. Neste algoritmo $Erro(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)})$ denota o erro de otimalidade do ponto $(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)})$ no Problema (P5) e é calculado pelo valor máximo da norma vetorial das equações obtidas ao aplicar as Condições Necessárias de Primeira Ordem para o Problema (P5) no ponto $(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)})$. E $Erro_{\mu_j}(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)})$ é o erro de otimalidade do Problema (P6) com $\mu = \mu_j$, para o ponto $(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)})$, e é calculado pelo valor máximo da norma vetorial das equações (18) com $\mu = \mu_j$ no ponto $(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)})$.

Algoritmo de Pontos Interiores com Filtro

1. Entrada $j = 0, k = 0, z^{(0)}, s^{(0)}, \lambda^{(0)}, \pi^{(0)}, \mu_0$
 2. **Enquanto** $Erro(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)}) > \epsilon_{tol}$ **faça**
 3. **Se** $Erro_{\mu_j}(z^{(k)}, s^{(k)}, \lambda^{(k)}, \pi^{(k)}) < \epsilon_{tol}^{\mu_j}$ **então**
 4. Atualize o parâmetro de barreira μ_{j+1}
 5. $j = j + 1$
 6. **Fim Se**
 7. Calcule J e $-\nabla L$
 8. Resolva o sistema $J \Delta d = -\nabla L$
 9. Calcule os passos α_k e α_k^π .
 10. Atualize as variáveis do problema:

$$z^{(k+1)} = z^{(k)} + \alpha_k \Delta z, s^{(k+1)} = s^{(k)} + \alpha_k \Delta s$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \alpha_k \Delta \lambda, \pi^{(k+1)} = \pi^{(k)} + \alpha_k^\pi \Delta \pi$$
 11. $k = k + 1$
 12. **Fim Enquanto**
-

3.3 Algoritmo de Pontos Interiores Modificado-Discreto (Algoritmo PIMD)

Para resolver um PNLDC como dado em (P2) é proposto o Algoritmo de Pontos Interiores Modificado Discreto apresentado na .

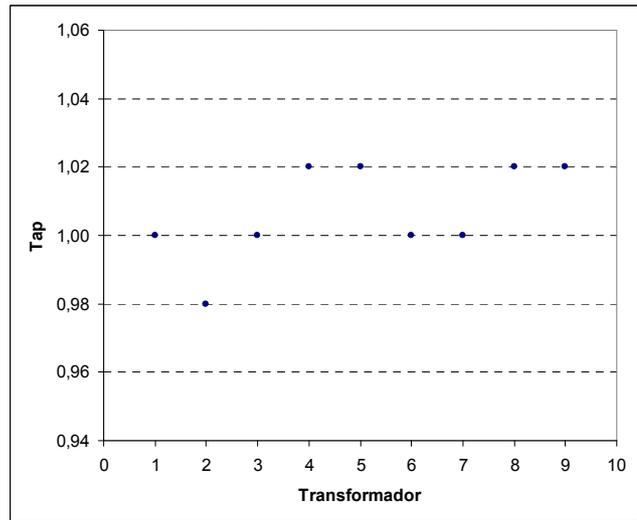


Figura 1: Algoritmo PIMD

No Algoritmo PIMD apresentado, em cada iteração um PNL (Problema Modificado (P3)) é resolvido. Se na solução obtida para este PNL as variáveis y (variáveis discretas no problema original) assumem valores discretos, admitindo-se uma tolerância ϵ , o algoritmo é finalizado e esta é uma solução para o Problema (P2). Caso contrário, a amplitude da penalidade $\gamma^{(K)}$ é atualizada e um novo PNL é resolvido. O processo prossegue até que as variáveis $y_i, i = 1, 2, \dots, n_y$ assumam valores discretos segundo uma tolerância, ou seja,

até que $|y_i^* - y_i^{*'}| < \epsilon$ para $i = 1, 2, \dots, n_y$, onde $y_i^{*'}$ é o valor discreto mais próximo de y_i^* e ϵ é a tolerância de convergência. A amplitude da função auxiliar é atualizada por $\gamma^{(K+1)} = c\gamma^{(K)}$, em que c é uma constante tal que $1 < c < 2$. Para c escolhido dessa maneira os valores de $\gamma^{(K)}$ crescem gradualmente no decorrer das iterações.

O valor de $\gamma^{(1)}$ depende da ordem de grandeza da função objetivo. Para valores de $\gamma^{(1)}$ escolhidos adequadamente, no início do processo iterativo as soluções encontradas para (P3) assumem valores próximos da solução do Problema (P2') – relaxação contínua de (P2). As soluções vão gradualmente convergindo para a solução ótima de (P2) com y assumindo valores discretos ou valores muito próximos de valores discretos, até que a tolerância de convergência seja alcançada.

4 RESULTADOS NUMÉRICOS

4.1 Testes Numéricos

Para avaliar a eficiência do método de solução proposto foram realizados testes numéricos com o Algoritmo de Pontos Interiores Modificado-Discreto, apresentado na Seção 3.3, aplicado ao problema de DOR. Os sistemas elétricos de potência usados para verificar a eficiência da abordagem proposta foram 118 e 300 barras. Os dados destes sistemas foram obtidos na página <http://www.ee.washington.edu/research/pstca> (acessado em outubro de 2009). O pacote IPOPT foi utilizado com a interface do GAMS - *General Algebraic Modeling System* (<http://www.gams.com>) para resolver o Problema Modificado do Algoritmo PIMD.

Nos testes realizados os limites mínimos e máximos das magnitudes de tensão das barras dos sistema foram $\underline{V}_p = 0,9p.u.$ e $\overline{V}_p = 1,1p.u.$. As restrições de igualdade de balanço de potência ativa e reativa têm tolerância de convergência de 10^{-4} p.u.. Nos testes realizados com o sistema 118 barras foi definido que os *taps*, variáveis de natureza discreta, devem pertencer ao conjunto $\{0,96; 0,98; 1; 1,02; 1,04\}$. Nos testes com o sistema 300 barras foi definido que os *taps* devem pertencer ao conjunto $\{0,90; 0,92; 0,94; 0,96; 0,98; 1,00; 1,02; 1,04; 1,06; 1,08; 1,10\}$. Em todos os testes apresentados neste trabalho foi considerado $c = 1,3$, $\alpha = 0$, $\epsilon = 0,0005$ e $\gamma^{(1)} = 10^{-4}$.

4.1.1 Sistema Elétrico 118 Barras

O Sistema Elétrico 118 barras tem as seguintes características:

- 1 barra de geração (*slack*);
- 53 barras de controle de reativo;

- 64 barras de carga;
- 186 linhas de transmissão;
- 9 transformadores com *tap* variável.

O modelo matemático do problema de DOR possui:

- 181 restrições de igualdade;
- 360 restrições de desigualdade;
- 235 variáveis contínuas;
- 9 variáveis discretas (*taps*).

O Algoritmo PIMD forneceu uma solução discreta na 17ª iteração. A solução obtida é apresentada nas Figuras 2 e 3. Nesta solução as perdas de potência ativa do sistema são 110,91 MW.

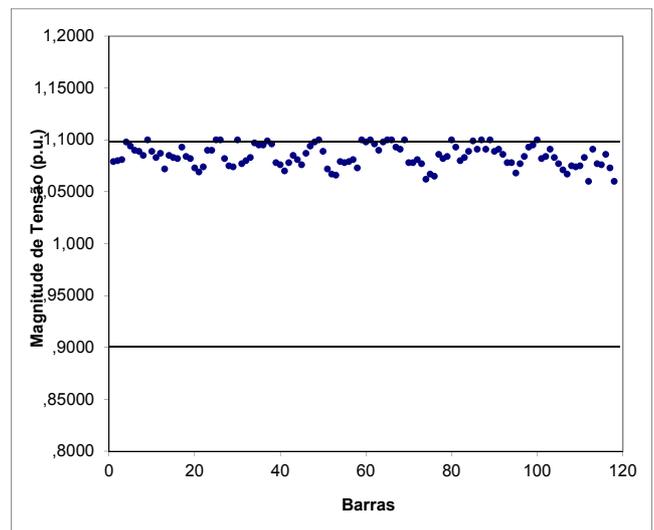


Figura 2: Magnitude de Tensão nas Barras – 118 barras

4.1.2 Sistema Elétrico 300 Barras

O Sistema Elétrico 300 barras tem as seguintes características:

- 1 barra de geração (*slack*);
- 68 barras de controle de reativo;
- 231 barras de carga;
- 409 linhas de transmissão;
- 50 transformadores com *tap* variável.

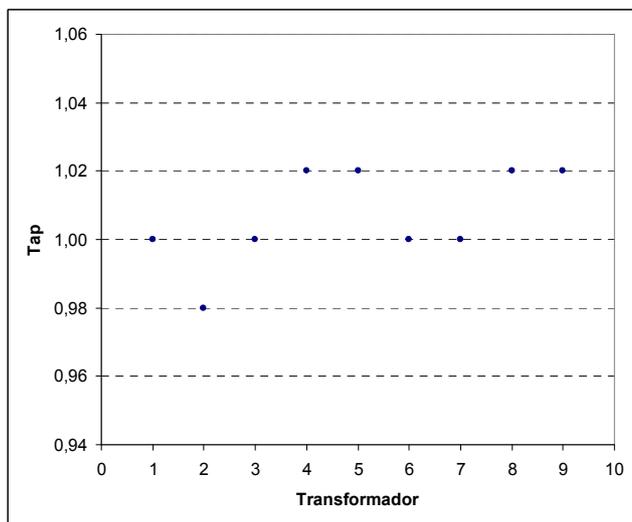


Figura 3: Taps dos Transformadores – 118 barras.

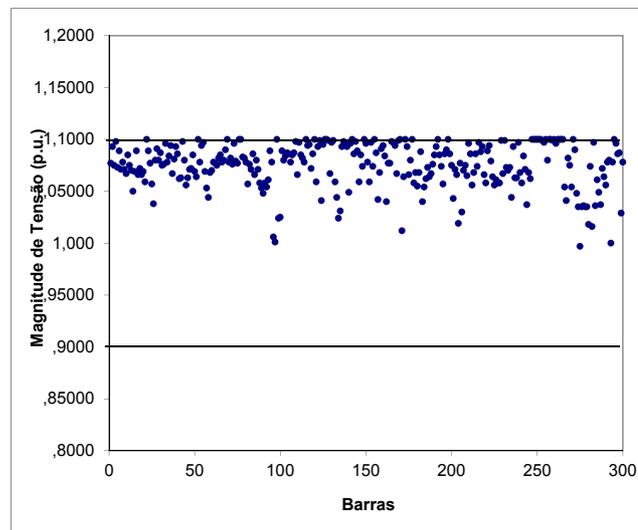


Figura 4: Magnitude de Tensão nas Barras – 300 barras

O modelo matemático do problema de DOR possui:

- 530 restrições de igualdade;
- 836 restrições de desigualdade;
- 599 variáveis contínuas;
- 50 variáveis discretas (*taps*).

O Algoritmo PIMD forneceu uma solução discreta na 21^a iteração. A solução obtida é apresentada nas Figuras 4 e 5. Nesta solução as perdas de potência ativa do sistema são 344,02 MW.

4.2 Análise dos resultados

Para analisar a qualidade das soluções encontradas para o problema de DOR, modelado com variáveis contínuas e discretas, as soluções obtidas pelo Algoritmo PIMD (apresentadas na Seção 4) foram comparadas com as soluções obtidas por Arredondamento da Solução Contínua e com as soluções obtidas pelos Métodos *Branch-and-Bound* (foi utilizado o pacote BONMIN) e *Outer-Approximation* (foi utilizado o pacote DICOPT).

4.2.1 Algoritmo PIMD × Arredondamento

Para obter uma solução por arredondamento resolveu-se, inicialmente, o problema de DOR ignorando-se a natureza discreta dos *taps*, ou seja, considerando todas as variáveis do modelo como contínuas. Na solução obtida, os valores dos *taps* (variáveis originalmente discretas) são aproximados

para seus valores discretos mais próximos, e fixadas nestes valores. Finalmente, resolve-se o problema de DOR com *tap* fixo. Esses problemas de programação não linear são resolvidos pelo Método de Pontos Interiores com Filtro descrito na Seção 3.2, implementado no pacote IPOPT, e utilizado em interface com o *software* GAMS.

É importante destacar que o Algoritmo PIMD proposto neste trabalho não consiste de um arredondamento para as variáveis discretas. As Figuras 6 e 7 ilustram, para os sistemas 118 e 300 barras, os valores obtidos para os *taps* quando se resolve o problema pelo Método de Pontos Interiores com Filtro considerando todas as variáveis contínuas, os valores para os *taps* obtidos por arredondamento da solução contínua, e a solução discreta obtida pelo método apresentado neste trabalho. Nas Figuras 6 e 7 nota-se que as soluções obtidas pelo Algoritmo PIMD diferem da solução contínua arredondada. Na Tabela 1 são apresentados os valores da função objetivo e o tempo total de processamento para estas soluções. Para o sistema 118 barras não foi possível obter uma solução discreta por arredondamento pois ao se fixar valores para os *taps* o problema torna-se infactível, o que demonstra a inviabilidade de utilização deste método. Para o sistema 300 barras a solução obtida por Arredondamento apresenta um baixo tempo de processamento, no entanto, na solução obtida pelo Algoritmo PIMD tem-se um valor menor para as perdas de potência ativa na transmissão.

4.2.2 Algoritmo PIMD × Outer Approximation

O DICOPT (*DI*crete and *C*ontinuous *OPT*imizer) é um pacote para resolver problemas de programação não linear com variáveis inteiras e contínuas. O Algoritmo DICOPT é ba-

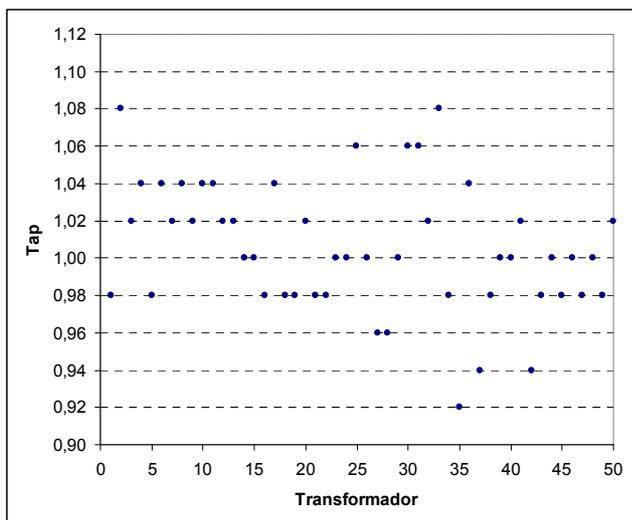


Figura 5: Taps dos Transformadores – 300 barras.

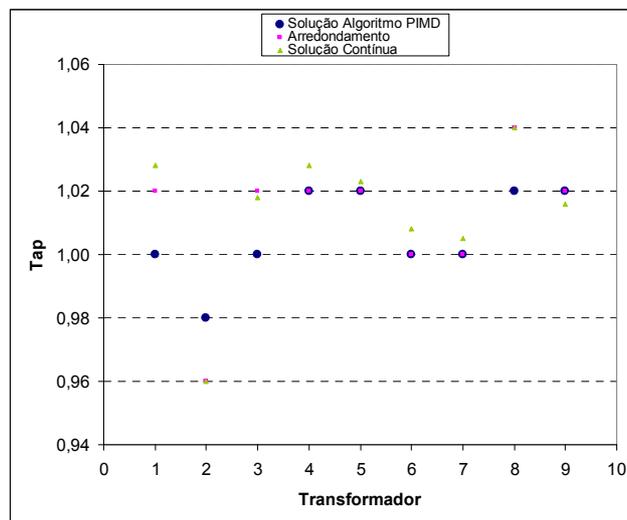


Figura 6: Solução PIMD x Arredondamento x Solução Contínua - 118 barras.

Tabela 1: Comparação dos resultados - Arredondamento

Sistema Elétrico	PIMD		Sol. Contínua		Arredondamento	
	Perdas	tempo(s)	Perdas	tempo(s)	Perdas	tempo(s)
118	110,91	9,369	110,59	0,952	—	—
300	344,02	25,165	343,16	1,185	344,55	3,04

seado em extensões do Método *Outer Approximation* e estratégias de relaxamento das restrições de igualdade. O DICOPT foi executado em interface com o *software* GAMS (<http://www.gams.com/dd/docs/solvers/dicopt.pdf>). O pacote é capaz de resolver problemas que envolvem funções não convexas, porém nestes casos não há garantias de que a solução obtida é um ótimo global.

A apresenta os valores da função objetivo e o tempo total de processamento de cada método para cada um dos problemas testados, e a relação do tempo de processamento entre Algoritmo PIMD e pacote DICOPT, dada em porcentagem, calculada por:

$$t_{AD} = \frac{t_A}{t_D} 100 \quad (23)$$

em que t_{AD} é a porcentagem do tempo consumido pelo Algoritmo PIMD sob o DICOPT, t_D é o tempo consumido pelo DICOPT na resolução do problema e t_A é o tempo de processamento do Algoritmo PIMD na resolução do problema.

Nesta tabela observa-se que o Algoritmo PIMD é competitivo pela qualidade das soluções obtidas e pelo tempo de processamento. O tempo de processamento do pacote DICOPT foi melhor que o tempo de processamento do Algo-

Tabela 2: Comparação dos resultados – DICOPT

Sistema Elétrico	PIMD		DICOPT		PIMD DICOPT (%)
	Perdas	tempo(s)	Perdas	tempo(s)	
118	110,91	9,369	110,65	3,929	238,46
300	344,02	25,165	—	7200	0,35

ritmo PIMD para os testes com o sistema elétrico 118 barras. No teste com o sistema 300 barras o tempo de processamento do pacote foi limitado em 2 horas. Ressalta-se que o pacote DICOPT não encontrou uma solução discreta para o problema de DOR para o sistema 300 barras neste tempo, enquanto o Algoritmo PIMD resolveu este problema eficientemente em 25 segundos, o que demonstra a inviabilidade de utilização do pacote DICOPT para a resolução do problema de DOR de sistemas elétricos de grande porte.

4.2.3 Algoritmo PIMD × Branch-and-Bound

O BONMIN (*Basic Open-source Non-linear Mixed Integer programming*) é um pacote para resolver problemas de programação não linear com variáveis inteiras e contínuas, o qual é distribuído gratuitamente pelo projeto COIN-OR. Esse pacote possui cinco algoritmos disponíveis, dos quais foi utilizado o algoritmo B-BB pois entre os algoritmos disponíveis no pacote este é o recomendado para se resolver problemas que envolvem funções não convexas. O algoritmo B-BB consiste do método *Branch-and-Bound* e utiliza o pacote IPOPT para resolver os PNL da árvore de busca. O BONMIN foi executado em interface com o *software* GAMS (<http://www.gams.com/dd/docs/solvers/coin.pdf>).

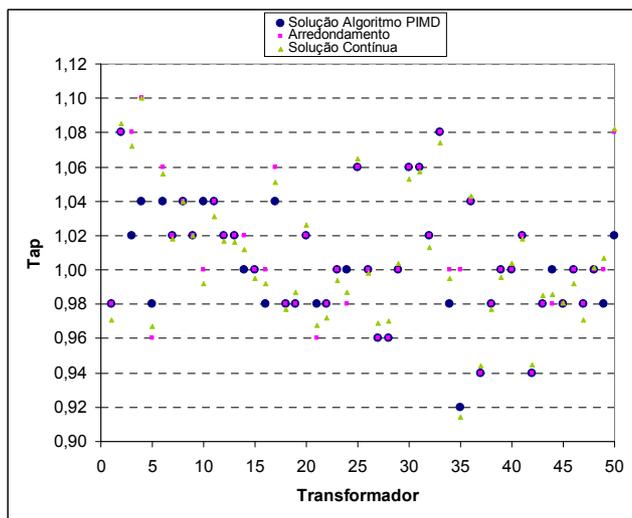


Figura 7: Solução PIMD x Arredondamento x Solução Contínua – 300 barras.

Tabela 3: Comparação dos resultados - BONMIN

Sistema Elétrico	PIMD		CoinBomin		$\frac{PIMD}{Bonmin} (\%)$
	Perdas	tempo(s)	Perdas	tempo(s)	
118	110,91	9,369	110,64	6,340	147,78
300	344,02	25,165	343,88	145,508	17,29

A apresenta os valores da função objetivo e o tempo de processamento de cada método para cada um dos sistemas elétricos testados, e a relação do tempo de processamento entre Algoritmo PIMD e o pacote BONMIN, dada em porcentagem, calculado por:

$$t_{AB} = \frac{t_A}{t_B} 100 \quad (24)$$

em que t_{AB} é a porcentagem do tempo consumido pelo Algoritmo PIMD sob o Algoritmo BONMIN, t_B é o tempo consumido pelo BONMIN na resolução do problema e t_A é o tempo consumido pelo Algoritmo PIMD na resolução do problema.

Nesta tabela observa-se que o Algoritmo PIMD é competitivo pela qualidade das soluções obtidas e pelo baixo tempo computacional.

Nos testes com o sistema 118 barras o pacote BONMIM resolveu o problema de DOR em tempo inferior ao Algoritmo PIMD. Nos testes com o sistema elétrico 300 barras o Algoritmo PIMD teve um desempenho melhor do que o pacote BONMIN. Ressalta-se que o tempo de processamento do pacote BONMIN foi de mais de 2 minutos para a resolução do problema de DOR do sistema elétrico 300 barras,

enquanto que o Algoritmo PIMD resolveu este problema eficientemente em 25 segundos. O desempenho do Algoritmo PIMD foi superior ao desempenho do pacote BONMIN na resolução do problema de DOR para o sistema elétrico 300 barras pelo baixo tempo computacional, o que comprova que o Algoritmo PIMD é mais eficiente que o pacote BONMIN na resolução do problema de DOR de sistemas elétricos de grande porte.

5 CONCLUSÃO

DOR é um problema de programação não linear com variáveis contínuas e discretas. Uma função auxiliar foi desenvolvida e agregada à função objetivo tornando o problema contínuo e diferenciável. Uma sequência de problemas de PNL, com variáveis apenas contínuas, converge para a solução do problema original, que contém variáveis contínuas e discretas. Verifica-se que a solução por arredondamentos tem problemas que podem comprometer a obtenção de uma solução para o problema. Os resultados obtidos demonstram o potencial da abordagem proposta para ajustar de maneira ótima as variáveis contínuas e discretas do problema DOR. O tempo computacional observado para o método proposto para o sistema de maior porte foi melhor que os tempos apresentados pelos pacotes BONMIN e DICOPT, além da questão da obtenção da solução inteira, fato não conseguido com o DICOPT para o sistema de 300 barras. A abordagem proposta mostra-se portanto de potencial relevante para resolução de problemas não-lineares além do problema abordado neste estudo.

AGRADECIMENTOS

Esta pesquisa contou com o apoio financeiro da FAPESP e do CNPq.

REFERÊNCIA BIBLIOGRÁFICA

- Adibi, M. M.; Polyak, R. A.; Griva, I. A. and Mili, L. (2003). Optimal transformer tap selection using modified barrier-augmented lagrangian method. IEEE Transactions on Power Systems, Vol. 18, No. 1, pp. 251-257.
- Bakirtzis, A. G. Biskas, P. N., Zoumas, C. E. and Petridis, V. (2002). Optimal power flow enhanced genetic algorithm. IEEE Transactions on Power Systems, Vol. 17, No. 2, pp.229-236.
- Capitanescu, F. and Wehenkel, L. (2010). Sensitivity-based approaches for handling discrete variables in optimal power flow computations. IEEE Transactions on Power Systems, Vol. 25, No. 4, pp. 1780-1789.

- Davydov, E. G. and Sigal, I. Kh. (1972). Application of the penalty function method in integer programming problems. *Engineering Cybernetics*, Vol. 10, pp. 21-24.
- Ding X. and Wang, X. (2004). Interior point cutting plane method for optimal power flow. *IMA Journal of Management Mathematics*, Vol. 15, No. 4, pp. 355-368.
- Fletcher, R., and Leyffer, S. (2002). Nonlinear programming without a penalty function. *Mathematical Programming*, Vol. 91, No. 2, pp. 239-269.
- Fu, J.-F.; Fenton, R. G. and Cleghorn, W. L. (1991). A mixed integer-discrete-continuous programming method and its application to engineering design optimization. *Engineering Optimization*, Vol. 17, No. 4, pp. 263-280.
- Geoffrion, A. M. (1972). Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, Vol. 10, pp. 237-260.
- Gupta, O. K. and Ravindran, A. (1985). Branch and bound experiments in convex nonlinear integer programming. *Management Science*, Vol. 31, pp. 1533-1546.
- Liu E. W.-H.; Papelexopoulos, A. D. and Tinney, W. F. (1992). Discrete shunt controls in a Newton optimal power flow. *IEEE Transactions on Power Systems*, Vol. 7, No. 4, pp.1509-1518.
- Liu, M.; Tso, S.K. and Cheng, Y. (2002). An extended nonlinear primal-dual interior-point algorithm for reactive-power optimization of large-scale power systems with discrete control variables. *IEEE Transactions on Power Systems*, Vol. 17, No. 4, pp.982-991.
- Liu, L.; Wang, X.; Ding, X. and Chen, H. (2009). A robust approach to optimal power flow with discrete variables. *IEEE Transactions on Power Systems*, Vol. 24, No. 3, pp. 1182-1190.
- Leyffer, S. (1993). *Deterministic Methods for Mixed Integer Nonlinear Programming*. Tese (Doutorado). Department of Mathematics and Computer Science, University of Dundee.
- Soler E. M. e da Costa, G.R.M. (2010). Ajuste ótimo discreto dos taps dos transformadores no problema de fluxo de potência ótimo. In: XVIII Congresso Brasileiro de Automática, Bonito-MS, v. 1, pp. 584-591.
- Wächter, A. (2002). An interior point algorithm for large-scale nonlinear optimization with applications in process engineering, Tese (Doutorado). Carnegie Mellon University.
- Wächter, A. and Biegler, L. T. (2005). Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM Journal on Optimization*, Vol. 16, No. 1, 1-31.
- Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, Vol. 1, 25-57.
- Westerlund, T. and Petersson, F. (1995). A cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering*, Vol. 19, pp. 131-136.