



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação
Departamento de Sistemas de Energia Elétrica

Relatório Técnico
(Bolsa de Doutorado Sandwich no Exterior)

**Método de Pontos Interiores com Técnicas de Região de
Garantia para Resolver o Problema de Fluxo de Potência
Ótimo Reativo**

Autor: Marina Lavorato de Oliveira
Orientador: Prof. Dr. Ariovaldo Verandio Garcia
Supervisor: Prof. Dr. Francisco Javier Heredia Cervera (GNOM – UPC)

O presente relatório traz as principais atividades que foram desenvolvidas pela autora durante o seu estágio na Universitat Politècnica de Catalunya - Barcelona - Espanha, no período de 10/04/2008 a 19/04/2008.

Campinas, Novembro de 2008.

Resumo

Neste relatório técnico são propostos modelos matemáticos e técnicas de solução para resolver o problema de fluxo de potência ótimo reativo, apresentando:

a) uma nova formulação para resolver o problema de fluxo de potência ótimo reativo utilizando equações de injeção de corrente, com características esparsas que diminuem o esforço computacional especialmente quando é usado um método de pontos interiores;

b) uma comparação entre cinco diferentes programas de otimização de programação não linear para resolver o problema de fluxo de potência ótimo reativo utilizando equações de potência, avaliando o tempo computacional total, tempo computacional de avaliação das funções, número de iterações e casos de não convergência;

c) uma comparação do esforço computacional entre a formulação do fluxo de potência ótimo reativo utilizando equações de injeção de corrente e a formulação do fluxo de potência ótimo reativo utilizando equações de potência, mostrando que a primeira formulação é aproximadamente 41% mais rápida que a segunda formulação; e

d) um algoritmo de método de pontos interiores com técnicas de região de garantia para resolver a formulação do fluxo de potência ótimo reativo utilizando equações de potência, mostrando ser um algoritmo rápido e capaz de assegurar uma convergência global do problema.

Palavras-chave: Fluxo de potência ótimo reativo; Equações de injeção de corrente; Equações de potência; Programas de otimização comerciais; Problemas de programação não-linear; Otimização de sistemas de potência; AMPL.

Sumário

Lista de Figuras	v
Lista de Tabelas	vi
1 Introdução	1
2 Método de Pontos Interiores	4
2.1 O Problema Original	4
2.2 O Método de Pontos Interiores Primal-Dual	5
2.2.1 Atualização das Variáveis Primais e Duais	7
2.2.2 Máximo Tamanho de Passo	7
2.2.3 Redução do Parâmetro de Barreira	7
2.2.4 Critério de Convergência	8
2.2.5 Algoritmo Geral	8
2.2.6 Ponto Inicial	9
2.2.7 Tempo de Execução	10
3 O Problema de Fluxo de Potência Ótimo Reativo (FPOR)	11
3.1 FPOR Utilizando Formulação de Potência com Tensões em Coordenadas Retangulares	12
3.2 Formulação de Injeção de Corrente com Tensões em Coordenadas Retangulares para um Problema de Fluxo de Potência	13
3.3 FPOR Utilizando Formulação de Injeção de Corrente com Tensões em Coordenadas Retangulares	16
3.4 Estrutura da Matriz Hessiana (\mathbf{W}) para as Formulações de Potência e Corrente com Tensões em Coordenadas Retangulares	17
4 Comparação de Programas de Otimização Não-Lineares para Resolver o problema de FPOR	20
4.1 Programas de Otimização Não-Linear	21
4.1.1 IPOPT (Interior Point OPTimizer)	21

4.1.2	KNITRO (Nonlinear Interior-point Trust Region Optimizer)	21
4.1.3	LOQO	22
4.1.4	MINOS (Modular In-core Nonlinear Optimization System)	22
4.1.5	SNOPT (Sparse Nonlinear OPTimizer)	23
4.2	Perfil de Desempenho	23
4.3	Provas e Resultados	24
5	Comparação Computacional entre as formulações do FPOR Utilizando Equações de Potência e Injeção de Corrente	30
6	Método de Pontos Interiores com Técnicas de Região de Garantia	36
6.1	Formulação do Problema	37
6.2	Função de Mérito	38
6.3	Critério de Convergência	39
6.3.1	Critério de Convergência Geral	39
6.3.2	Teste de Convergência do Problema de Barreira	40
6.4	Transição Entre o Método de Pontos Interiores e de Região de Garantia	41
6.5	Algoritmo Geral	41
6.6	Método de Região de Garantia	42
6.7	Função de Mérito para a Região de Garantia	43
6.8	Algoritmo para o Método de Região de Garantia	45
6.9	Testes e Resultados	46
7	Conclusão	49
	Referências bibliográficas	51
A	Artigo submetido	57
B	Avaliações dos Professores Brasileiro e Estrangeiro	58
C	Bilhetes de Passagem Aérea	59
D	Confirmação de endereço e Telefones de Contato no Brasil	60
D.1	Endereço Residencial	60
D.2	Endereço da Universidade	60

Lista de Figuras

3.1	Sistema Ilustrativo	17
3.2	Matriz \mathbf{W} do problema (3.1) para o sistema ilustrativo da Fig. 3.1	18
3.3	Matriz \mathbf{W} do problema (3.21) para o sistema ilustrativo da Fig. 3.1	18
4.1	Perfil de Desempenho para o Numero de Iterações	27
4.2	Perfil de Desempenho para o Tempo de Avaliação de Funções	28
4.3	Perfil de Desempenho para o Tempo Total de Processamento	29
5.1	Tempo computacional total gasto para cada contingência	34
5.2	Tempo de avaliação das funções do problema para cada contingência	35
6.1	Trajectoria de Convergência da Função Objetivo	47
6.2	Trajectoria de Convergência	48

Lista de Tabelas

4.1	Características dos Sistemas Testados	25
5.1	Resumo das Características de Otimização	31
5.2	Tempo total e de avaliação das funções do problema para o caso base	32
5.3	Tempo computacional total gasto pela análise das contingências	33
6.1	Resumo das Características de Otimização	46

Capítulo 1

Introdução

Os programas de Fluxo de Potência Ótimo (FPO) são cada vez mais utilizados como ferramentas para a análise de operação dos sistemas de potência em regime permanente, e isto é devido a: a) O surgimento de novos e potentes recursos computacionais a um custo relativamente baixo, os quais permitem reduzir o esforço requerido na solução do problema de otimização; b) A consolidação das técnicas matemáticas para a resolução de problemas de grandes dimensões; c) Uma maior consideração dos interesses econômicos na operação dos sistemas de energia elétrica, devido à reestruturação dos mercados elétricos.

A primeira formulação do problema de FPO é atribuída a Carpentier na década de 60, com base em um problema de despacho econômico [1], e era excessivamente complexa, razão pela qual não foi adotada posteriormente. Simplificações deste modelo resultaram em uma formulação na qual se resolve um único problema de otimização, sujeito às equações de Fluxo de Potência (FP) ou Fluxo de Carga e à minimização de um índice de desempenho, considerando as limitações de operação do sistema. Dependendo do índice de desempenho, podem ser definidos diferentes problemas de otimização para um mesmo sistema [2].

O problema de FPO tem como objetivo principal determinar o estado de operação ótimo de um sistema de potência em regime permanente podendo ser modelado como um problema de Programação Não-Linear (PNL), em que deve ser minimizada uma função objetivo sujeita a um conjunto de restrições técnicas e econômicas [1]. Existem diversos métodos clássicos propostos para resolver o problema de FPO, que podem ser agrupados como: a) o Método Gradiente [1] e [3], primeira proposta para resolver o problema de fluxo de potência ótimo. Apesar de seu rigor matemático este método utiliza algumas aproximações que levam a uma convergência lenta e em zig-zag perto da solução ótima; b) a Programação Quadrática Sucessiva [4] e [5], utiliza derivadas de segunda ordem e melhoram a convergência do método gradiente. Este último método é baseado em um processo Quasi-Newton utilizando uma aproximação da matriz Hessiana (construída iterativamente) da função de Lagrange, porém a matriz Hessiana é cheia e torna este método lento quando o número de variáveis de controle é muito grande; c) os Métodos tipo Newton [6], que buscam um ponto que satisfaça diretamente as condições de Karush-Kuhn-Tucker (KKT). No entanto, poucos métodos são confiáveis ou rápidos, o desafio é identificar as restrições de desi-

gualdade que são ativas na solução ótima, para isso são aplicados alguns métodos heurísticos; d) a Programação Linear Sucessiva [7] e [8], que se baseia na linearização da função objetivo e das restrições do problema. Para a solução deste problema, é adotado um modelo linear incremental e um algoritmo dual-Simplex; e mais recentemente e) o Método de Pontos Interiores (MPI) [9], criado inicialmente para resolver problemas de programação linear [10], apresenta um melhor desempenho computacional para problemas de grandes dimensões em comparação com os métodos clássicos, como o método Simplex. Apesar de seu reconhecimento como ferramenta importante na decisão da operação do sistema de potência, a dimensão do problema de FPO e as não-linearidades associadas à sua modelagem, dificultam sua utilização em tempo real.

Uma formulação apropriada do problema e uma efetiva metodologia de solução são necessárias para a aplicação real do FPO em sistemas de potência. Atualmente, o FPO é implementado na maioria dos novos centros de controle. No entanto, é utilizado efetivamente em poucos sistemas. Alguns desafios que o FPO tem que superar, a fim de ser uma ferramenta confiável para a operação em tempo real, são: a) Resposta rápida; b) Robustez da solução; c) Número de controles afetados na solução; d) Modelamento das variáveis discretas; e) Inclusão das transações econômicas bilaterais.

Nos últimos anos, tem sido mostrado na literatura especializada, o ótimo desempenho da aplicação do MPI na otimização da operação de sistemas de potência (SEP). Em [11] – [13] é utilizado o MPI para resolver diferentes problemas de FPO. Também tem sido utilizado para resolver o problema de estimação de estados [14], cálculo de máximo carregamento do sistema [15], mínimo corte de carga [16] e [17], análise de estabilidade de tensão [18], coordenação hidrotérmica [19] e despacho econômico ótimo e seguro [20].

Os resultados mostram que o MPI tem um grande potencial para resolver problemas de sistemas de potência, quando comparados com os métodos tradicionais, apesar de serem uma extensão direta dos MPI para programação linear, sem fazer nenhuma consideração quando o problema não é convexo. Alguns trabalhos já foram utilizados para melhorar a trajetória de convergência do MPI considerando a convexidade do problema de otimização da operação dos SEP [21] – [24]. Uma característica destes métodos de estratégia de modificar o sistema de KKT é usado no cálculo das direções de busca. A região de garantia é uma região no espaço do problema de otimização no qual pode-se garantir que um modelo quadrático é um modelo adequado da função objetivo, esta técnica pertence relativamente a uma nova classe de algoritmos de otimização. A medida do processo de convergência é o diâmetro da região fechada chamada “região de garantia”, o qual é uma quantidade controlável, que pode ser ampliada ou reduzida em função de como o modelo local prediz o comportamento da função objetivo. Um forte apoio teórico, assim como eficiência prática e robustez, foram os principais motivos para se iniciar a aplicação da técnica de região de garantia no problema de otimização da operação dos SEP [25] – [32].

Em [33] são incorporadas ao MPI duas poderosas ferramentas para a solução do problema de programação não linear: programação quadrática sucessiva (PQS) e técnicas de região de garantia (MPI-TRG). A PQS trabalha de forma eficiente as não linearidades das restrições, enquanto a região de garantia permite que o algoritmo trate os problemas convexos e não convexos, permitindo o uso direto das informações das derivadas de segunda ordem e proporcionando uma proteção contra

a dependência linear dos gradientes das restrições. A formulação e a solução dos subproblemas de barreira com restrições de igualdade geradas pelo MPI-TRG são muito importantes e determinam os passos do algoritmo. A formulação dos subproblemas fornece as características da iteração primal ou primal-dual e asseguram que as variáveis de folga permaneçam positivas. A técnica usada para resolver os subproblemas tem um grande impacto na eficiência e robustez do algoritmo. Novas estratégias para acelerar o processo de convergência deste método são mostradas na referência [34].

O método de programação não linear mencionado anteriormente demonstra ser eficiente para resolver o problema de programação não linear de grande porte e não convexos, como o problema de fluxo de potência ótimo. As características não lineares do problema de fluxo de potência ótimo têm sido pouco utilizadas para explorar a característica de convergência do método apresentado em [33].

Um novo algoritmo de conjunto ativo (ACA) para problemas de programação não linear de grande porte é apresentado em [35]. Este algoritmo calcula a direção de busca em duas etapas. Na primeira etapa um problema de programação linear é solucionado para identificar o conjunto ativo de restrições na solução. Na segunda etapa, um problema de programação quadrática com restrições de igualdade (PCRI) é solucionado considerando somente restrições ativas da solução do problema de programação linear. O problema de PCRI incorpora técnicas de região de garantia e é solucionado, de forma aproximada, utilizando um método gradiente conjugado projetado. Algumas estratégias para acelerar o processo de convergência deste método são mostradas na referência [36].

Uma comparação do desempenho computacional (tempo computacional total, tempo computacional de avaliação das funções, número de iterações e casos de não convergência) entre o MPI-TRG e o ACA, mostrou que o ACA apresenta piores resultados no que diz respeito a tempo computacional e convergência global quando comparado com MPI-TRG (ver os resultados mostrados no Cap. 4). Com os resultados desta pesquisa inicial optou-se por dedicar mais tempo para o entendimento, análise e implementação do MPI-TRC para resolver o problema de fluxo de potência ótimo.

Este relatório técnico está organizado da forma descrita a seguir: No Capítulo 2 é feita uma breve revisão do método de pontos interiores Primal–Dual. No Capítulo 3 apresenta-se uma nova formulação para resolver o problema de fluxo de potência ótimo reativo utilizando equações de injeção de corrente (FPOR-IC). No Capítulo 4 apresenta-se uma comparação entre cinco diferentes softwares de programação não linear para resolver o problema de fluxo de potência ótimo reativo utilizando equações de potência (FPOR-EP). No Capítulo 5 apresenta-se uma comparação do esforço computacional entre a formulação do FPOR-IC e a formulação do FPOR-EP. No Capítulo 6 apresenta-se um algoritmo de método de pontos interiores com técnicas de região de garantia para resolver a formulação do FPOR-EP. No anexo A é mostrado o artigo que foi submetido para a revista *IET Generation, Transmission & Distribution*. Nos apêndices B, C e D estão os comprovantes solicitados pelo CNPq.

Capítulo 2

Método de Pontos Interiores

Neste capítulo será apresentada uma descrição dos passos mais importantes do MPI de forma a mostrar as principais matrizes que são necessárias montar no processo iterativo.

2.1 O Problema Original

Um problema de PNL pode ser representado pela Eqs. (2.1):

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.a.} \quad & \mathbf{g}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \tag{2.1}$$

Onde $\mathbf{x} \in \mathcal{R}^{nx}$ representa as variáveis de decisão, $f(\mathbf{x}) : \mathcal{R}^{nx} \rightarrow \mathcal{R}$ é a função objetivo do problema, $\mathbf{g}(\mathbf{x}) : \mathcal{R}^{nx} \rightarrow \mathcal{R}^{ng}$ são as restrições de igualdade e $\mathbf{h}(\mathbf{x}) : \mathcal{R}^{nx} \rightarrow \mathcal{R}^{nh}$ são as restrições de desigualdade do problema. Caso existam variáveis canalizadas estas são transformadas em restrições de desigualdade.

O primeiro passo na derivação do MPI é transformar todas as restrições de desigualdade de (2.1) em restrições de igualdade adicionando as variáveis de folga não negativas \mathbf{s} .

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.a.} \quad & \mathbf{g}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) + \mathbf{s} = \mathbf{0} \\ & \mathbf{s} \geq \mathbf{0} \end{aligned} \tag{2.2}$$

A adição das variáveis de folga modifica a dimensão do problema de otimização aumentando o número de variáveis. Entretanto, o problema original é transformado num problema de otimização restrito exclusivamente a restrições de igualdade. Com esta metodologia, a utilização de funções de penalidade e a determinação do conjunto de restrições de desigualdade ativas na solução deixam de ser necessárias. As condições de não negatividade ($\mathbf{s} \geq \mathbf{0}$) da expressão (2.2)

podem ser incorporadas na função objetivo original como termos de barreira logarítmica como é apresentado na expressão (2.3).

$$\begin{aligned} \min \quad & f(\mathbf{x}) - \mu \sum_{i=1}^{nh} \ln s_i \\ \text{s.a.} \quad & \mathbf{g}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) + \mathbf{s} = \mathbf{0} \end{aligned} \quad (2.3)$$

Onde μ^k é o parâmetro de barreira que decresce de forma monotônica até zero no processo iterativo ($\mu^k > 0$). A seqüência de parâmetros $\{\mu^k\}_{k=0}^k$ gera uma seqüência de sub-problemas dados por (2.3) baseados no teorema dado por [37], se $\mu^k \downarrow 0$ a seqüência $\{x(\mu^k)\}_{k=0}^k$ de soluções de (2.3) tende a x^* , um mínimo local de (2.1). A seqüência de sub-problemas dado pela Eq.(2.3) descreve uma trajetória, nos espaços primal e dual, conhecida como Trajetória Central ou Rota Central. A função Lagrangeana do problema de programação não linear com restrições de igualdade (2.3) é:

$$L(x, s, \lambda, \pi, \mu) = f(\mathbf{x}) - \mu \sum_{i=1}^{ndh} \ln s_i + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\pi}^T (\mathbf{h}(\mathbf{x}) + \mathbf{s}) \quad (2.4)$$

em que $\boldsymbol{\lambda} \in \mathfrak{R}^{ng}$, $\boldsymbol{\pi} \in \mathfrak{R}^{nh}$, são os vetores multiplicadores de Lagrange e são chamados variáveis duais. Um ponto mínimo local \mathbf{x}^* do problema (2.3) pode ser calculado em termos do ponto estacionário da função Lagrangeana, no qual deve satisfazer as condições necessárias de otimalidade de primeira ordem (CNOPO) de Karush-Kuhn-Tucker (KKT), apresentada na expressão (2.5) [11, 12, 13, 17]:

$$\begin{aligned} \nabla f(\mathbf{x}) + \mathbf{J}_{\mathbf{g}}(\mathbf{x})^T \boldsymbol{\lambda} + \mathbf{J}_{\mathbf{h}}(\mathbf{x})^T \boldsymbol{\pi} &= \mathbf{0} \\ \mathbf{g}(\mathbf{x}) &= \mathbf{0} \\ \mathbf{h}(\mathbf{x}) + \mathbf{s} &= \mathbf{0} \\ \mathbf{S}\boldsymbol{\pi} &= \mu^k \mathbf{e} \end{aligned} \quad (2.5)$$

Onde $\nabla f(\mathbf{x})$ é o gradiente da função objetivo, $\mathbf{J}_{\mathbf{g}}(\mathbf{x}) \in \mathfrak{R}^{ng \times nx}$ é a matriz Jacobiana das restrições de igualdade $\mathbf{g}(\mathbf{x})$ e $\mathbf{J}_{\mathbf{h}}(\mathbf{x}) \in \mathfrak{R}^{nh \times nx}$ é a matriz Jacobiana das restrições de desigualdade $\mathbf{h}(\mathbf{x})$. \mathbf{S} é uma matriz diagonal com s_i 's compondo os elementos da diagonal; e \mathbf{e} é um vetor de tamanho apropriado formado por elementos iguais a 1.

2.2 O Método de Pontos Interiores Primal-Dual

Mesmo que o sistema de KKT (2.5) seja um sistema de equações não lineares, a sua solução é geralmente aproximada por uma única iteração do método de Newton (a direção de Newton é somente uma medida para seguir um trajeto de minimizar, parametrizada por μ^k). Aplicando o método de Newton para resolver o sistema (2.5), o sistema de equações lineares indefinido (2.6) é obtido (ver [12]). O método de Newton necessita que sejam definidos os seguintes pontos iniciais: parâmetro de barreira logarítmica μ^0 , variáveis primais \mathbf{x}^0 e \mathbf{s}^0 e variáveis duais $\boldsymbol{\lambda}^0$ e $\boldsymbol{\pi}^0$. O método

de Newton consiste em um processo iterativo no qual se aproxima um ponto inicial $(\mathbf{x}^0, \mathbf{s}^0, \boldsymbol{\lambda}^0, \boldsymbol{\pi}^0)$ ao ponto de solução $(\mathbf{x}^*, \mathbf{s}^*, \boldsymbol{\lambda}^*, \boldsymbol{\pi}^*)$ através de uma seqüência de pontos $(\mathbf{x}^k, \mathbf{s}^k, \boldsymbol{\lambda}^k, \boldsymbol{\pi}^k)$ que indica a trajetória percorrida durante o processo iterativo. A cada iteração k do método de Newton, o ponto de solução tem que satisfazer as condições de não negatividade.

$$\mathbf{s}^k, \boldsymbol{\pi}^k \geq 0$$

Assim, a solução do sistema de equações dado pela Eq. (2.5) utilizando o método de Newton, implica em resolver um sistema de equações lineares do tipo $\mathbf{Ax} = \mathbf{b}$ a cada iteração k , como é apresentado em [11, 12, 13, 17], dado por (2.6).

$$\begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{J}_g(\mathbf{x})^T & \mathbf{J}_h(\mathbf{x})^T \\ \mathbf{0} & \mathbf{S}^{-1}\boldsymbol{\Pi} & \mathbf{0} & \mathbf{I} \\ \mathbf{J}_g(\mathbf{x}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_h(\mathbf{x}) & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \mathbf{s}^k \\ \Delta \boldsymbol{\lambda}^k \\ \Delta \boldsymbol{\pi}^k \end{bmatrix} = \begin{bmatrix} \nabla f(\mathbf{x}) + \mathbf{J}_g(\mathbf{x})^T \boldsymbol{\lambda} + \mathbf{J}_h(\mathbf{x})^T \boldsymbol{\pi} \\ \mathbf{S}\boldsymbol{\pi} - \mu \mathbf{e} \\ \mathbf{g}(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) + \mathbf{s} \end{bmatrix} \quad (2.6)$$

Em que a matriz \mathbf{W} é a matriz Hessiana das restrições de igualdade $\mathbf{H}_{g_j}(\mathbf{x})$ e desigualdade $\mathbf{H}_{h_j}(\mathbf{x})$ e da função objetivo $\mathbf{H}_f(\mathbf{x})$, dada por:

$$\mathbf{W} = \mathbf{H}_f(\mathbf{x}) + \sum_{j=1}^{ng} \lambda_j \mathbf{H}_{g_j}(\mathbf{x}) + \sum_{j=1}^{nh} \pi_j \mathbf{H}_{h_j}(\mathbf{x}) \quad (2.8)$$

e, $\Delta \mathbf{x}^k$, $\Delta \boldsymbol{\lambda}^k$, e $\Delta \boldsymbol{\pi}^k$ são as direções do método de Newton da iteração k .

As condições necessárias para a utilização do método de Newton na solução de (2.5) são:

1. Existência de um ponto estacionário $(\mathbf{x}^*, \mathbf{s}^*, \boldsymbol{\lambda}^*, \boldsymbol{\pi}^*)$ que seja solução do sistema de (2.5) e que satisfaça as condições de KKT.
2. Existência e continuidade das segundas derivadas parciais locais no ponto estacionário.
3. O conjunto das restrições de igualdade e de desigualdade ativas no ponto estacionário devem ser linearmente independente.
4. Existência da complementaridade estrita $(\mathbf{s}^T \boldsymbol{\pi})$ no ponto estacionário.

As condições teóricas para a utilização do método de Newton geralmente não são avaliadas nas aplicações reais para sistemas de potência de grande porte devido à complexidade de cálculo. Assim, não é garantido que a solução obtida para (2.1) seja um ótimo global. Mesmo assim, o processo de otimização fornece, em geral, soluções melhores que os métodos clássicos. A fim de dar continuidade à análise teórica do MPI, são consideradas satisfeitas as condições para a aplicação do método de Newton para resolução iterativa de equações não lineares.

2.2.1 Atualização das Variáveis Primais e Duais

A cada iteração k é resolvida a Eq. (2.6) e depois é feita uma estimação dos valores das variáveis do problema, obtidos por:

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathbf{x}^k + \alpha_p^k \Delta \mathbf{x}^k \\ \mathbf{s}^{k+1} &= \mathbf{s}^k + \alpha_p^k \Delta \mathbf{s}^k \\ \lambda^{k+1} &= \lambda^k + \alpha_d^k \Delta \lambda^k \\ \boldsymbol{\pi}^{k+1} &= \boldsymbol{\pi}^k + \alpha_d^k \Delta \boldsymbol{\pi}^k\end{aligned}\quad (2.9)$$

2.2.2 Máximo Tamanho de Passo

$\alpha_k \in (0, 1]$ é o tamanho de passo. O máximo tamanho de passo, para cada iteração é dado por (2.10), como foi proposto em [13].

$$\alpha_p^k = \min_i \left\{ 1, \min_{\Delta s_i^k < 0} \left(\frac{-s_i^k}{\Delta s_i^k} \right) \right\} \quad (2.10)$$

$$\alpha_d^k = \min_i \left\{ 1, \min_{\Delta \pi_i^k < 0} \left(\frac{-\pi_i^k}{\Delta \pi_i^k} \right) \right\}$$

$$\alpha_k = \min\{\alpha_p^k, \alpha_d^k\}$$

O valor escalar $\gamma \in (0, 1)$ é um fator de segurança para assegurar que o próximo ponto satisfaça as condições de não negatividade. Um valor típico para este fator é $\gamma = 0.9995$.

2.2.3 Redução do Parâmetro de Barreira

O valor residual da condição de complementaridade é chamado gap de complementaridade e é calculado em cada iteração k por:

$$\rho^k = (\mathbf{x}^k)^T \boldsymbol{\pi}^k \quad (2.11)$$

A seqüência $\{\rho^k\}_{k=0}^{\infty}$ deve convergir para zero, e a relação entre o ρ^k e μ^k , implícito nas condições de (2.5), sugere que μ^k poderia ser reduzido em cada iteração k em função da diminuição do gap de complementaridade, dada pela expressão (2.12).

$$\mu^{k+1} = \beta^k \frac{\rho^k}{nh} \quad (2.12)$$

em que $\beta \in (0, 1)$ é o decréscimo esperado de ρ^k , mas não necessariamente realizado, e é chamado *centering parameter* e sua interpretação é: se $\beta^k = 1$, o sistema KKT (2.5) define uma direção central, um passo Newton para um ponto no trajeto da barreira. Caso contrário, se $\beta^k = 0$, dá um passo Newton puro, também conhecido como a direção *affine-scaling*. Para compensar os dois objetivos, de reduzir μ^k e melhorar a direção central, β^k é escolhido dinamicamente como foi proposto em [13], $\beta^{k+1} = \max\{0, 95\beta^k; 0, 1\}$, com $\beta^0 = 0,2$.

2.2.4 Critério de Convergência

O processo do método de Newton é repetido até que os critérios de convergência, apresentados na expressão (2.13) sejam atingidos.

- Factibilidade Primal

$$\max\{\|\mathbf{g}(\mathbf{x}^k)\|_2\} \leq \epsilon_f$$

- Factibilidade Dual

$$\frac{\|\nabla \mathbf{f}(\mathbf{x}^k) + \mathbf{J}_g(\mathbf{x}^k)^T \boldsymbol{\lambda}^k + \mathbf{J}_h(\mathbf{x}^k)^T \boldsymbol{\pi}^k\|_\infty}{1 + \|\mathbf{x}^k\|_2} \leq \epsilon_f$$

- Condição de Optimalidade

$$\frac{\rho^k}{1 + \|\mathbf{x}^k\|_2} \leq \epsilon_o \tag{2.13}$$

$$\mu^k \leq \epsilon_\mu$$

- Desvio da Função Objetivo

$$\frac{|f(\mathbf{x}^k) - f(\mathbf{x}^{k-1})|}{1 + |f(\mathbf{x}^k)|} \leq \epsilon_f$$

em que ϵ_f , ϵ_o e ϵ_μ são os erros das *condições de factibilidade, otimalidade e parâmetro de barreira*, respectivamente. Tipicamente $\epsilon_f = \epsilon_o = 10^{-5}$ e $\epsilon_\mu = 10^{-8}$.

2.2.5 Algoritmo Geral

O Método de Pontos Interiores Primal - Dual pode ser sintetizado no seguinte algoritmo :

1. Inicializar μ^0 , β^0 e obter um ponto inicial $(\mathbf{x}^0, \mathbf{s}^0, \boldsymbol{\lambda}^0, \boldsymbol{\pi}^0)$ que satisfaça as condições de não-negatividade $(\mathbf{s}^k, \boldsymbol{\pi}^k \geq 0)$.
2. Calcular $\mathbf{g}(\mathbf{x}^0)$, $\mathbf{h}(\mathbf{x}^0)$, $\mathbf{J}_g(\mathbf{x}^0)$ e $\mathbf{J}_h(\mathbf{x}^0)$.
3. Calcular o vetor da parte direita da expressão (2.6).
4. Fazer $k = 0$.

Repetir

6. Calcular a matriz \mathbf{W} usando (2.8).
7. Montar e resolver no ponto atual o sistema (2.6).
8. Obter o máximo tamanho de passo usando (2.10).
9. Atualizar as variáveis usando (2.9).
10. Calcular $\mathbf{g}(\mathbf{x}^{k+1})$, $\mathbf{h}(\mathbf{x}^{k+1})$, $\mathbf{Jg}(\mathbf{x}^{k+1})$ e $\mathbf{Jh}(\mathbf{x}^{k+1})$.
11. Calcular μ^{k+1} (2.12), ρ^{k+1} (2.11) e β^{k+1} .
12. Calcular o vetor da parte direita da expressão (2.6).
13. Fazer $k = k + 1$.

Até se obter a convergência.

2.2.6 Ponto Inicial

A solução do problema de otimização deve encontrar-se no interior ou na fronteira da região de convergência. O ponto inicial e a trajetória de convergência não possuem essa restrição. De fato, as condições de otimalidade são referidas ao ponto candidato ao ótimo, não sendo necessariamente satisfeitas durante o processo iterativo. A resolução iterativa determina uma seqüência de pontos que aproximam o ponto inicial à solução. Em particular, as equações que determinam a inclusão do ponto na região de convergência serão em geral satisfeitas apenas no final do processo iterativo. Esta é uma importante vantagem na utilização das versões dos MPI para PNL. Os algoritmos dos pontos interiores realizam uma trajetória interior à região formada pelas restrições de desigualdade. Por conseguinte, esses algoritmos necessitam da determinação de um ponto inicial interior a esta região [38]. Embora o ponto inicial somente precise manter as condições de não negatividade, o processo de convergência é sensível ao ponto inicial e o desempenho dos MPI pode melhorar se alguma iniciação heurística for usada [13, 39, 40]. Uma iniciação heurística é dada a seguir.

1. A estimação do \mathbf{x}^0 é dado usando o ponto médio entre os limites superior e inferior para as variáveis com limites.
2. As variáveis de folga primais são inicializadas como

$$s_i^0 = \begin{cases} h_i(\mathbf{x}^0) & \text{se } h_i(\mathbf{x}^0) \geq 1 \\ 1 & \text{caso contrário} \end{cases}$$

3. As variáveis duais λ_i^0 podem ser inicializadas com -1 , 0 ou 1 , de acordo com o problema a ser resolvido.
4. As variáveis de folga duais são inicializadas como:

$$\boldsymbol{\pi}^0 = \mu^0 (\mathbf{S}^0)^{-1} \mathbf{e}$$

em que \mathbf{S}^0 é uma matriz diagonal definida pelos valores de \mathbf{s}^0 . Tipicamente $\mu^0 = \{10; 1 \text{ ou } 0, 1\}$.

2.2.7 Tempo de Execução

O tempo de execução dos métodos de pontos interiores depende de vários fatores, como por exemplo: as características do computador utilizado, ou o esquema de pré-processamento e ordenamento utilizado. O número de iterações do MPI para conseguir a convergência depende principalmente de: a) o tipo de método de pontos interiores utilizado; b) da natureza não-linear do problema por resolver; c) do ponto e parâmetro de barreira inicial; d) da taxa de decrescimento esperado do parâmetro de barreira; e) do tamanho de passo na direção de busca; f) do critério de parada. Em [41] são analisados os fatores que influem na velocidade e convergência do MPI. O tempo de cálculo numa iteração do método de pontos interiores depende principalmente de: a) o método numérico utilizado para resolver (2.6); e b) da estrutura esparsa do problema.

Capítulo 3

O Problema de Fluxo de Potência Ótimo Reativo (FPOR)

O Problema de Fluxo de Potência Ótimo tem sido muito estudado ao longo dos anos, pois é uma ferramenta de fundamental importância no mercado elétrico atual devido às características de competitividade que este apresenta. Tanto no planejamento como na operação dos sistemas de potência, a segurança e a confiabilidade são avaliadas usando uma certa quantidade de programas, entre os quais está incluso o FPO.

O número mínimo de restrições de igualdade do FPO é determinado pelas equações da rede de transmissão, como no problema de fluxo de potência. Adicionalmente, podem ser incluídas no problema de otimização restrições de igualdade que modelam características particulares da operação do sistema de potência (valores fixos de algumas variáveis ou uma combinação de variáveis do sistema).

As restrições de desigualdade são os limites impostos a uma variável ou conjunto de variáveis do sistema. Em relação à sua função, podem ser classificadas em três grandes grupos: a) Restrições Físicas, são as restrições impostas pelos limites da capacidade dos componentes do sistema. Exemplos: limites máximo e mínimo de geração de potência ativa e reativa das unidades geradoras, limites nos valores dos taps dos OLTC, limites de transmissão de potência aparente nas linhas de transmissão, limites da capacidade de geração de potência reativa dos compensadores shunt, etc; b) Restrições Operacionais, a operação do sistema impõe limites que devem ser considerados no modelo. Exemplos: limites máximo e mínimo da magnitude da tensão nas barras, diferença angular máxima entre barras, etc; e c) Restrições de Segurança, as restrições de segurança representam um grupo de restrições relacionadas a um conjunto de contingências determinadas pela análise de segurança em tempo real. O resultado da análise de segurança é uma operação confiável do sistema.

A função objetivo representa a variável ou conjunto de variáveis que se deseja otimizar. Existe uma grande variedade de funções objetivo possíveis de utilizar na operação da rede elétrica, relacionadas ao objetivo do estudo. No FPO, algumas das mais utilizadas são: a) Mínimo Custo de Potência Ativa; b) Minimização das Perdas de Potência Ativa; c) Mínimo Desvio de uma Distribuição

3.1 FPOR Utilizando Formulação de Potência com Tensões em Coordenadas Retangulares 12

de Potência Ativa Pré-especificada; d) Mínimo Desvio de um Perfil de Tensões Pré-especificado; e) Mínimo Corte de Carga; e d) Mínima Ação de Controle. As funções objetivo anteriores podem ser combinadas em um único problema de otimização, através de um problema multi-objetivo.

O problema de fluxo de potência ótimo reativo (FPOR) é um caso especial de problema de FPO e é um dos principais assuntos abordados nos estudos da operação de sistemas de potência. Em alguns casos o FPOR é considerado um problema independente de despacho econômico de potência ativa, e seu objetivo é manter o perfil de tensão em uma escala aceitável, de forma a minimizar a perda total de energia na transmissão, usando como variáveis de controle os valores das magnitudes de tensões dos geradores, os taps dos transformadores com tap variáveis, a potência das fontes de compensação reativa VAR (bancos de capacitores e compensadores síncronos) avaliadas nos sistemas de potência. As restrições incluem os limites de VAR e magnitudes de tensões dos geradores, os limites das magnitudes de tensões das barras de carga, os limites dos taps dos transformadores com tap variável, os limites das fontes de VAR, as restrições de segurança e de balanço de potência nas barras [42]. Existem várias técnicas clássicas para resolver o problema de FPOR (método do gradiente, programação linear e quadrática sucessivas e métodos do tipo Newton), entre as técnicas mais recentes encontram-se o método de pontos interiores [11, 12, 13, 17, 43] e o método de estratégia de região de garantia [25, 26, 27, 28, 29]. Soluções do FPOR por aplicações diretas de alguns programas de otimização não-linear podem ser encontradas em [19, 44, 45, 46].

3.1 FPOR Utilizando Formulação de Potência com Tensões em Coordenadas Retangulares

O problema de minimização de perdas de potência ativa utilizando equações e tensões em coordenadas retangulares é formulado como um problema de PNL [13] definido por:

$$\begin{aligned}
 \min \quad & l = \mathbf{G}_{sw}:(e_{sw}\mathbf{e} + f_{sw}\mathbf{f}) + \mathbf{B}_{sw}:(f_{sw}\mathbf{e} - e_{sw}\mathbf{f}) & (3.1) \\
 \text{s.a.} \quad & \mathbf{G}_k:(e_k\mathbf{e} + f_k\mathbf{f}) + \mathbf{B}_k:(f_k\mathbf{e} - e_k\mathbf{f}) - P_k^{SP} = 0, \quad \forall k \in \tilde{\mathcal{B}} \\
 & \mathbf{G}_k:(f_k\mathbf{e} - e_k\mathbf{f}) - \mathbf{B}_k:(e_k\mathbf{e} + f_k\mathbf{f}) - Q_k^{SP} = 0, \quad \forall k \in \mathcal{L} \\
 & \underline{Q}_k \leq \mathbf{G}_k:(f_k\mathbf{e} - e_k\mathbf{f}) - \mathbf{B}_k:(e_k\mathbf{e} + f_k\mathbf{f}) \leq \overline{Q}_k, \quad \forall k \in \mathcal{G} \\
 & \underline{Q}_k \leq \mathbf{G}_k:(f_k\mathbf{e} - e_k\mathbf{f}) - \mathbf{B}_k:(e_k\mathbf{e} + f_k\mathbf{f}) \leq \overline{Q}_k, \quad \forall k \in \mathcal{S} \\
 & \underline{V}_k^2 \leq e_k^2 + f_k^2 \leq \overline{V}_k^2, \quad \forall k \in \mathcal{B} \\
 & \underline{tap}_m \leq tap_m \leq \overline{tap}_m, \quad \forall m \in \mathcal{T}
 \end{aligned}$$

Onde sw é a barra de referência, \mathbf{G}_k : e \mathbf{B}_k : são as k -ésimas colunas das matrizes de condutâncias \mathbf{G} e susceptâncias \mathbf{B} de barras, respectivamente. e_k e f_k são as componentes das k ésimas posições do vetor das tensões (complexas) real \mathbf{e} e imaginária \mathbf{f} de barras, respectivamente. $P_k^{SP} = P_k^G - P_k^L$ e $Q_k^{SP} = Q_k^G - Q_k^L$ são as potências ativa e reativa especificadas, respectivamente, na barra k ; P_k^G e Q_k^G são as potências ativa e reativa geradas, respectivamente, na barra k ; P_k^L e Q_k^L são as potências

ativa e reativa demandadas, respectivamente, na barra k . $\overline{Q}_k = \overline{Q}_i^G - Q_i^L$ e $\underline{Q}_k = \underline{Q}_i^G - Q_i^L$ são os limites máximo e mínimo de potências reativas na barra k , onde \overline{Q}_k^G e \underline{Q}_k^G são os limites máximos e mínimos da potência gerada ou de compensação shunt na barra k . $\overline{V}_k, \underline{V}_k$ são os limites máximo e mínimo das magnitudes de tensão na barra k ; $\overline{tap}_m, tap_m, \underline{tap}_m$ são os valores máximo, corrente e mínimo dos taps dos transformadores em derivação do ramo m . \mathcal{B} é o conjunto de todas as barras, $\tilde{\mathcal{B}}$ é o conjunto de todas as barras menos a barra de referência, \mathcal{L} é o conjunto de barras com fonte de VAR fixa, \mathcal{G} é o conjunto de todos os geradores, \mathcal{S} é o conjunto de todos os compensadores shunt nas barras, \mathcal{T} é o conjunto de todos os transformadores com LTC. As variáveis de decisão do problema (3.1) são: as componentes real e imaginária da tensão (complexa) das barras e os taps dos transformadores com tap variável. A componente imaginária da tensão (complexa) da barra de referência é fixa com um valor igual a zero ($f_{sw} = 0$).

Se a tensão complexa em cada barra é conhecida, é possível calcular qualquer outra variável da rede. É por esta razão, que a tensão complexa é considerada como variável de otimização. Os taps dos transformadores com tap variável, também devem ser considerados como variáveis de otimização do sistema. A este conjunto mínimo de variáveis de otimização é possível adicionar outras variáveis, dependendo do objetivo do estudo e do método de otimização utilizado. Algumas variáveis que podem ser consideradas como variáveis de otimização são: Potências ativa e reativa de geração, limite máximo de transmissão em uma linha, transações contratadas entre barras, etc.

No problema de fluxo de potência ótimo reativo a geração de potência ativa do conjunto de barras $\tilde{\mathcal{B}}$ é considerada como conhecida. Normalmente, a geração de potência ativa tem sido calculado por meio de um despacho econômico ativo.

3.2 Formulação de Injeção de Corrente com Tensões em Coordenadas Retangulares para um Problema de Fluxo de Potência

Seguindo as principais idéias contidas em [47] e [48], será apresentada uma formulação para o fluxo de potência utilizando o método de injeção de corrente modelado para cargas do tipo potência constante. Esta formulação provou ser eficiente para resolver este tipo de problema.

A equação básica de potência injetada em uma determinada barra k de um sistema é dado por:

$$S_k = P_k + jQ_k = V_k I_k^* \quad (3.2)$$

Assim, tem-se para a corrente injetada nesta mesma barra que:

$$I_k = \frac{P_k + jQ_k}{V_k^*} = \sum_{m \in \Omega_k} (Y_{km} V_m) \quad (3.3)$$

Logo, pode-se escrever que:

$$\sum_{m \in \Omega_k} (Y_{km} V_m) - \frac{P_k + jQ_k}{V_k^*} = 0 \quad (3.4)$$

A equação (3.4) representa o balanço de corrente na barra k , e pode ser separada em duas equações referentes às componentes real e imaginária das correntes injetadas nas barras, escritas em função das partes real e imaginárias das tensão na barra k . Desta forma, as equações de corrente imaginária e real a serem resolvidas no problema de fluxo de potência, denotadas por $\Delta I r_k$ e $\Delta I i_k$ respectivamente, são dadas por:

$$\Delta I i_k = (\mathbf{G}_k : \mathbf{f} + \mathbf{B}_k : \mathbf{e}) - \frac{P_k^{SP} f_k - Q_k^{SP} e_k}{e_k^2 + f_k^2} = 0 \quad (3.5)$$

$$\Delta I r_k = (\mathbf{G}_k : \mathbf{e} - \mathbf{B}_k : \mathbf{f}) - \frac{P_k^{SP} e_k + Q_k^{SP} f_k}{e_k^2 + f_k^2} = 0 \quad (3.6)$$

O sistema de equações não-lineares dado por (3.5) e (3.6) é resolvido utilizando o método de Newton-Raphson:

$$\begin{pmatrix} \mathbf{B}_1 & \mathbf{G}_1 \\ \mathbf{G}_2 & \mathbf{B}_2 \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} \Delta \mathbf{I}_i \\ \Delta \mathbf{I}_r \end{pmatrix} \quad (3.7)$$

Onde $\Delta \mathbf{I}_i$ e $\Delta \mathbf{I}_r$ são os vetores das partes imaginaria e real dos erros de corrente, respectivamente, dados por (3.5) e (3.6). As submatrizes \mathbf{B}_1 , \mathbf{G}_1 , \mathbf{B}_2 e \mathbf{G}_2 para um ramo genérico km são dadas por:

$$\mathbf{B}_1 = \frac{\partial \Delta I i_k}{\partial e_k} = \begin{pmatrix} B_{kk} - a_k & B_{km} \\ B_{mk} & B_{mm} - a_m \end{pmatrix} \quad (3.8)$$

$$\mathbf{G}_1 = \frac{\partial \Delta I i_k}{\partial f_k} = \begin{pmatrix} G_{kk} - b_k & G_{km} \\ G_{mk} & G_{mm} - b_m \end{pmatrix} \quad (3.9)$$

$$\mathbf{G}_2 = \frac{\partial \Delta I r_k}{\partial e_k} = \begin{pmatrix} G_{kk} + b_k & G_{km} \\ G_{mk} & G_{mm} + b_m \end{pmatrix} \quad (3.10)$$

$$\mathbf{B}_2 = \frac{\partial \Delta I r_k}{\partial f_k} = \begin{pmatrix} -B_{kk} - a_k & -B_{km} \\ -B_{mk} & -B_{mm} - a_m \end{pmatrix} \quad (3.11)$$

onde a_m e b_m são dados pelas Eq.s (3.12)-(3.13).

$$a_k = \frac{-2e_k f_k (P_k^{SP}) + (Q_k^{SP})(e_k^2 - f_k^2)}{(e_k^2 + f_k^2)^2} \quad (3.12)$$

$$b_k = \frac{2e_k f_k (Q_k^{SP}) + (P_k^{SP})(e_k^2 - f_k^2)}{(e_k^2 + f_k^2)^2} \quad (3.13)$$

Os elementos fora da diagonal principal das submatrizes da Eq. (3.7), referentes às derivadas parciais de (3.5) e (3.6) em relação a e_k e f_k , são iguais aos elementos da matriz de admitâncias

nodais e são constantes durante todo o processo iterativo, cada uma dessas submatrizes será um bloco de ordem $(nb \times nb)$. Pode-se notar nas Eqs. (3.12)-(3.13) que nas quatro derivadas parciais em relação às tensões complexas, os termos a_k e b_k aparecem duas vezes cada um, o que significa que neste caso os elementos da diagonal da matriz Jacobiana são facilmente atualizados e de forma rápida, pois somente é necessário o cálculo de dois termos varáveis, resultando em uma matriz Hessiana das equações Eqs. (3.5) e (3.6) altamente esparsa.

Também se pode observar que para uma barra de transferência, barras cujas injeções de potência ativa e reativa iguais a zero, conseqüentemente, as potências ativa P_k^{SP} e reativa Q_k^{SP} especificadas são iguais a zero, tornando as Eqs. (3.6) e (3.5) lineares. Esta característica pode ser explorada pelo MPI para resolver um problema de FPOR usando a formulação de equações de corrente.

A matriz Hessiana das equações (3.5) e (3.6) possui os seguintes elementos:

$$\frac{\partial^2 \Delta I i_k}{\partial e_k^2} = \frac{Q_k^{SP}(2e_k^3 - 6e_k f_k^2) + P_k^{SP}(2f_k^3 - 6e_k^2 f_k)}{(e_k^2 + f_k^2)^3} \quad (3.14)$$

$$\frac{\partial^2 \Delta I i_k}{\partial e_k \partial f_k} = \frac{-Q_k^{SP}(2f_k^3 - 6e_k^2 f_k) + P_k^{SP}(2e_k^3 - 6e_k f_k^2)}{(e_k^2 + f_k^2)^3} \quad (3.15)$$

$$\frac{\partial^2 \Delta I i_k}{\partial f_k \partial e_k} = \frac{\partial^2 \Delta I i_k}{\partial e_k \partial f_k} \quad (3.16)$$

$$\frac{\partial^2 \Delta I i_k}{\partial f_k^2} = -\frac{\partial^2 \Delta I i_k}{\partial e_k^2} \quad (3.17)$$

$$\frac{\partial^2 \Delta I r_k}{\partial e_k^2} = -\frac{\partial^2 \Delta I i_k}{\partial e_k \partial f_k} \quad (3.18)$$

$$\frac{\partial^2 \Delta I r_k}{\partial f_k^2} = \frac{\partial^2 \Delta I i_k}{\partial e_k \partial f_k} \quad (3.19)$$

$$\frac{\partial^2 \Delta I r_k}{\partial e_k \partial f_k} = \frac{\partial^2 \Delta I r_k}{\partial f_k \partial e_k} = \frac{\partial^2 \Delta I i_k}{\partial e_k^2} \quad (3.20)$$

Pode ser observado aqui que somente é necessário calcular as derivadas das Eqs. (3.14) e (3.15) para montar a matriz Hessiana pois os outros elementos são equivalentes a essas duas equações. Este fato é importante quando se trata de economia do tempo computacional. Também pode ser observado que cada restrição da Eq. (3.14) (ou da Eq. (3.15)) tem matriz Hessiana formada por somente quatro elementos diferentes de zero.

3.3 FPOR Utilizando Formulação de Injeção de Corrente com Tensões em Coordenadas Retangulares

O problema de minimização das perdas de potência ativa de um sistema elétrico de potência, usando tensões em coordenadas retangulares, é formulado como um problema de PNL e é definido por:

$$\begin{aligned}
 & \min P_{sw}^G & (3.21) \\
 \text{s.a. } & (\mathbf{G}_k:\mathbf{f}+\mathbf{B}_k:\mathbf{e})-\frac{P_k^{SP}f_k-Q_k^{SP}e_k}{e_k^2+f_k^2}=0, \quad \forall k \in \mathcal{B} \\
 & (\mathbf{G}_k:\mathbf{e}-\mathbf{B}_k:\mathbf{f})-\frac{P_k^{SP}e_k+Q_k^{SP}f_k}{e_k^2+f_k^2}=0, \quad \forall k \in \mathcal{B} \\
 & \underline{Q}_k^G \leq Q_k^G \leq \overline{Q}_k^G, \quad \forall k \in \mathcal{G} \\
 & \underline{Q}_k^G \leq Q_k^G \leq \overline{Q}_k^G, \quad \forall k \in \mathcal{S} \\
 & \underline{V}_k^2 \leq e_k^2+f_k^2 \leq \overline{V}_k^2, \quad \forall k \in \mathcal{B} \\
 & \underline{tap}_m \leq tap_m \leq \overline{tap}_m, \quad \forall m \in \mathcal{T}
 \end{aligned}$$

Para poder utilizar as equações de corrente no ORPF foi necessário considerar de forma explícita as variáveis de geração de potência ativa na barra de referência e a geração de potência reativa dos geradores e dos compensadores shunt. Estas variáveis aparecem de forma implícita no modelo (3.1). Este aumento do tamanho das variáveis primais do problema de PNL não piora o desempenho computacional de (3.21) e permite ainda, explorar os elementos constantes que aparecem na matriz Jacobiana e a alta esparsidade das matrizes Hessianas das restrições de igualdade.

As variáveis de decisão do problema (3.21) são: as componentes (complexas) real e imaginária da tensão de barra, taps dos transformadores com tap variável, potência ativa de geração da barra de referência e potência reativa de geração dos geradores e compensadores shunt. A componente imaginária da tensão (complexa) da barra de referência é fixa com um valor igual a zero ($f_{sw} = 0$).

A grande motivação para a utilização da formulação do FPOR utilizando equações de injeção de correntes foi a alta esparsidade da matriz Hessiana do problema da Eq. (3.21), já que o maior esforço computacional para resolver um PNL, quando utilizado o método de pontos interiores, está em inverter uma matriz de Newton. Sendo assim, se o modelo utilizado proporciona uma matriz de Newton mais esparsa, o ganho de tempo computacional a cada iteração é considerável. Outro aspecto importante a ser considerado, é o número de elementos constantes e a serem calculados para a montagem das matrizes Jacobianas e Hessianas.

3.4 Estrutura da Matriz Hessiana (\mathbf{W}) para as Formulações de Potência e Corrente com Tensões em Coordenadas Retangulares

Nesta sessão será mostrada a estrutura da matriz Hessiana dos problemas da Eq. (3.1) e da Eq. (3.21) utilizando o exemplo dado pela Figura 3.1 para a análise de dois cenários: com e sem restrições de potência aparente nas linhas de transmissão.

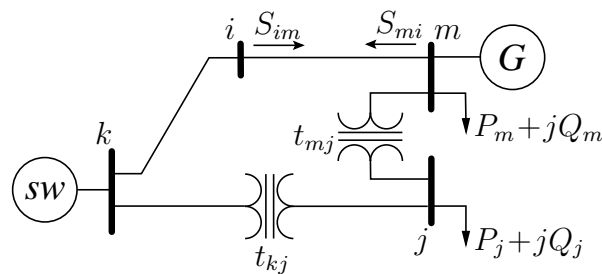


Fig. 3.1: Sistema Ilustrativo

Exemplo 1: Sem restrições de fluxo de potência nas linhas

Nos modelos (3.1) e (3.21) pode-se observar que cada barra contribui com três matrizes hessianas na formação da matriz \mathbf{W} , uma matriz hessiana dada pela restrição da magnitude de tensão e as outras duas dadas pelas equações de injeção de potência ativa e reativa, para o modelo (3.1), e as equações de injeção de corrente real e imaginária, para o modelo (3.21).

As Figuras (3.2) e (3.3) contém símbolos que representam os elementos que compõem as matrizes Hessianas \mathbf{W} . Os quadros sombreados representam a contribuição das matrizes hessianas das restrições de magnitude de tensão, e os símbolos “○”, “×”, “□” e “+” representam as contribuições das matrizes hessianas das equações de injeção de potências ativa e reativa (Fig. (3.2)) e das equações de injeção de corrente real e imaginária (Fig. (3.3)), nas barras k , j , i , m , respectivamente.

A Fig. (3.2) representa a matriz \mathbf{W} para o modelo (3.1). Cabe destacar que somente as matrizes Hessianas das equações de potência da barra i são constantes, as outras matrizes dependem do valor dos taps ou das tensões complexas. Para formar a matriz \mathbf{W} cada matriz Hessiana tem que ser multiplicada por seu respectivo multiplicador de lagrange a cada iteração. Na Fig. (3.2) existem 47 elementos diferentes de zero, com um grau de esparsidade de 41.98%. Fig. (3.3) representa a matriz \mathbf{W} para o modelo (3.21), onde existem 35 elementos diferentes de zero, com um grau de esparsidade de 75.69%. É possível observar na Fig. (3.3) a grande esparsidade da matriz \mathbf{W} quando comparada com a Fig. (3.2). Neste modelo a barra com injeção de potência igual a zero (barra i da Fig. (3.1)) forma duas Eqs. lineares, gerando uma contribuição nula na construção da matriz \mathbf{W} ,

	e_k	e_j	e_i	e_m	f_j	f_i	f_m	t_{mj}	t_{kj}
e_k	○	⊗	□		⊗	□			⊗
e_j	⊗	×		*			*	*	⊗
e_i	□		■	⊕		•	⊕		
e_m		*	⊕	⊕	*	⊕	•	*	
f_j	⊗			*	×		*	*	⊗
f_i	□		•	⊕		■	⊕		
f_m		*	⊕	•	*	⊕	⊕	*	
t_{mj}		*		*	*		*	+	
t_{kj}	⊗	⊗			⊗				○

Fig. 3.2: Matriz \mathbf{W} do problema (3.1) para o sistema ilustrativo da Fig. 3.1

	e_k	e_j	e_i	e_m	f_j	f_i	f_m	t_{mj}	t_{kj}	P_k	Q_k^G	Q_m^G
e_k	○								⊗	○	○	
e_j		×			×			*	⊗			
e_i			•	•		•	•					
e_m			•	⊕		•	⊕	*				+
f_j		×			×			*	⊗			
f_i			•	•		•	•					
f_m			•	⊕		•	⊕	*				+
t_{mj}		*		*	*		*	+				
t_{kj}	⊗	⊗			⊗				○			
P_k	○											
Q_k^G	○											
Q_m^G				+			+					

Fig. 3.3: Matriz \mathbf{W} do problema (3.21) para o sistema ilustrativo da Fig. 3.1

este fato pode ser observado pela falta do símbolo “□” na Fig. (3.3). A Fig. (3.3) mostra também que as derivadas de segunda ordem das restrições de igualdade e desigualdade do modelo (3.21) em relação às tensões preenchem somente as diagonais principais da matriz \mathbf{W} .

Exemplo 2: Com restrição de fluxo de potência aparente na linha

O segundo exemplo considera a inclusão das restrições de fluxo de potência aparente na linha $i - m$ (Eqs. (3.22) e (3.23)) para ambos os modelos. O objetivo desta inclusão é mostrar a mudança da matriz \mathbf{W} quando é adicionada uma restrição de fluxo (ver S_{im} e S_{mi} na Fig. (3.1)).

$$\begin{aligned}
 S_{im}^2 &= P_{im}^2 + Q_{im}^2 \leq \overline{S_{im}}^2 \\
 P_{im} &= (e_i^2 + f_i^2)g_{im} - g_{im}(e_i e_m + f_i f_m) + \\
 &\quad b_{im}(e_i f_m - e_m f_i) \\
 Q_{im} &= -(e_i^2 + f_i^2)(b_{im}^{sh} + b_{im}) + g_{im}(e_i f_m - e_m f_i) + \\
 &\quad b_{im}(e_i e_m + f_i f_m)
 \end{aligned} \tag{3.22}$$

$$\begin{aligned}
 S_{mi}^2 &= P_{mi}^2 + Q_{mi}^2 \leq \overline{S_{im}}^2 \\
 P_{mi} &= (e_m^2 + f_m^2)g_{im} - g_{im}(e_i e_m + f_i f_m) - \\
 &\quad b_{im}(e_i f_m - e_m f_i) \\
 Q_{mi} &= -(e_m^2 + f_m^2)(b_{im}^{sh} + b_{im}) - g_{im}(e_i f_m - e_m f_i) + \\
 &\quad b_{im}(e_i e_m + f_i f_m)
 \end{aligned} \tag{3.23}$$

Onde g_{im} , b_{im} , b_{im}^{sh} e $\overline{S_{im}}$ são a condutância, susceptância, susceptância shunt e limite de fluxo de potência aparente da linha de transmissão $i - m$.

Esta restrição gera uma contribuição na formação da matriz Hessiana \mathbf{W} que pode ser observada pelo símbolo “•” nas Fig. (3.2) e (3.3). Esta restrição ocasiona uma diminuição da esparsidade da matrix \mathbf{W} para ambos os modelos, porém cria um maior prejuízo ao modelo (3.21), não permitindo aproveitar totalmente as vantagens apresentadas pelas CI equations, pois a inclusão desta restrição gera um maior número de elementos a serem inclusos nas matrizes e conseqüentemente a serem calculados e fatorados a cada iteração.

Nesta seção foram apresentados dois exemplos da estrutura da matriz \mathbf{W} para o problema de FPOR considerando as equações de injeções de corrente e de injeção de potência. O Exemplo 1 mostra claramente que o número de elementos diferentes de zero da matriz \mathbf{W} do modelo (3.21) é menor que do modelo (3.1), mesmo sendo essa última de menor dimensão. O Exemplo 2 mostra que se for incluída ao modelo (3.21) uma restrição de fluxo de potência aparente na linha $i - m$ da Fig. (3.1), este ganha um número maior de elementos diferentes de zero para cada restrição de fluxo de linha do sistema do que quando é incluído o mesmo tipo de restrição ao modelo (3.1), logo, quando é considerado fluxo em todas as linhas do sistema, o modelo de injeção de corrente perde suas características de esparsidade perante o modelo de injeção de potência.

Capítulo 4

Comparação de Programas de Otimização Não-Lineares para Resolver o problema de FPOR

Os novos e poderosos recursos computacionais a um custo computacional relativamente baixo e técnicas matemáticas eficientemente consolidadas para a resolução de problemas de PNL de grande porte dão esperanças aos pesquisadores da área de otimização de sistemas de potência, de poder resolver de forma mais rápida e exata o problema de FPOR. Os programas de otimização não-linear: IPOPT [49] (versão 3.3.3), KNITRO [50] (versão 5.2.0), LOQO [51] (versão 6.06), MINOS [52], e SNOPT [53] representam o estado da arte no que se refere a software de otimização para resolver eficientemente os problemas de PNL de grande porte. Cada programa de otimização usa uma técnica de otimização diferente e tem sua própria característica de convergência. Estes e outros programas modernos estão disponíveis no servidor NEOS [54] (localizado no Laboratório Nacional de Argonne). Para avaliar a eficiência de cada técnica de otimização (tempo total de processamento, tempo de avaliação das funções, número de iterações, etc) é necessário analisar as características próprias do problema de PNL resolvido (características de não-linearidades da função objetivo e restrições, esparsidade das matrizes, etc).

Nesta parte do relatório o problema de FPOR utilizando formulação de potência com tensões em coordenadas retangulares foi modelada como um problema de PNL padrão, escrito em AMPL [55] e testado em cada programa de otimização. Foram utilizados cinco sistemas testes e três sistemas reais para criar casos a serem analisados. O objetivo é identificar o programa de otimização que possui a melhor característica de convergência. Para comparar o desempenho computacional de cada programa de otimização (número de iterações, tempo total de processamento e tempo de avaliação das funções) foi usado o perfil de execução proposto em [56]. Também estão relatadas uma comparação dos casos de não convergência.

4.1 Programas de Otimização Não-Linear

Alguns comentários sobre os programas de otimização não-linear serão apresentados a seguir.

4.1.1 IPOPT (Interior Point OPTimizer)

O IPOPT utiliza um algoritmo de pontos interiores primal-dual com um método de filtro de busca linear para assegurar a convergência global. O problema resolvido pelo IPOPT tem a forma do PNL dado por (4.1).

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.a.} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \end{aligned} \quad (4.1)$$

Onde \mathbf{x} são as variáveis de otimização (com possibilidade de possuir limites inferior \mathbf{x}^l e superior \mathbf{x}^u). A função objetivo $f(\mathbf{x})$ e a restrição de igualdade $\mathbf{h}(\mathbf{x})$ é assumida como sendo duas vezes diferenciável. Note que uma restrição de desigualdade não-linear também pode ser incluída à formulação acima usando variáveis de folga. O IPOPT tem como objetivo, encontrar uma solução local de (4.1). Em [49] pode-se encontrar uma descrição mais detalhada do algoritmo, incluindo a fase de restauração da factibilidade para o método do filtro, correção de segunda ordem, e correção de inércia da matriz de KKT. Também são consideradas neste algoritmo, heurísticas que permitem uma execução mais rápida do algoritmo. Em [57] e [58], são analisadas as propriedades de convergências global e local, respectivamente. O IPOPT é um software aberto escrito em C++, C, Fortran e MATLAB. (<http://www.coin-or.org/Ipopt/>)

4.1.2 KNITRO (Nonlinear Interior-point Trust Region Optimizer)

O KNITRO fornece três algoritmos para resolver o problema de PNL: a) algoritmo de pontos interiores direto, b) algoritmo de pontos interiores e gradiente conjugado, e c) algoritmo de conjunto ativo. O problema resolvido pelo KNITRO tem a forma do PNL dado por (4.2).

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.a.} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \quad (4.2)$$

Assume-se que $f(\mathbf{x})$, $\mathbf{h}(\mathbf{x})$ e $\mathbf{g}(\mathbf{x})$ são duas vezes diferenciáveis. O KNITRO possibilita utilizar os algoritmos de forma independente ou utilizar um procedimento de seleção automática implementado internamente que permite cruzar os três algoritmos durante o processo de solução. O primeiro algoritmo implementado pode ser encontrado na referência [50]. O algoritmo de pontos interiores direto aplica técnicas de barreira e fatoração direta da matriz de KKT de um sistema

não-linear. Este algoritmo executa melhor os problemas mal condicionados. Em [33] é descrito um algoritmo de pontos interiores e gradiente conjugado, onde são aplicadas técnicas de barreira usando o método do gradiente conjugado para resolver o sub-problema de KKT, a base teórica deste algoritmo pode ser encontrada em [59]. O algoritmo de conjuntos ativos implementa um método de programação linear-quadrática seqüencial descrita em [35]. Os três algoritmos têm diferenças fundamentais que conduzem a um comportamento diferente de cada problema de otimização não-linear. Ao mesmo tempo, fornecem uma série de diferentes caminhos para atacar as dificuldades dos problemas. É um software comercializado por "Ziena Optimization" e é avaliado em C++, C, Fortran. (<http://www.ziena.com/knitro.html>)

4.1.3 LOQO

O LOQO é baseado em um método de pontos interiores primal-dual infactível que soluciona problemas de otimização convexos e não-convexos, e problemas de otimização não lineares (4.3).

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.a.} \quad & \mathbf{h}^l \leq \mathbf{h}(\mathbf{x}) \leq \mathbf{h}^u \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \end{aligned} \quad (4.3)$$

Se o problema é convexo, então o LOQO encontra uma solução ótima global, caso contrário, encontra uma solução ótima local aproximada para um dado ponto inicial. Note que uma restrição de igualdade pode ser formulada assumindo os valores correspondentes de \mathbf{h}^l e \mathbf{h}^u para um mesmo valor. O LOQO é escrito em Fortran 77 e mais informações sobre ele, podem ser encontradas em [51]. Este software é aberto mas requer um arquivo de licença antes de ser usado. (<http://www.princeton.edu/~rvdb/>)

4.1.4 MINOS (Modular In-core Nonlinear Optimization System)

O MINOS usa uma implementação estável do método simplex primal para resolver problemas lineares. Para problemas com restrições lineares e uma função objetivo não-linear, o método do gradiente reduzido é empregado com uma aproximação quasi-Newton para a Hessiana reduzida. Para problemas de programação não-linear, o MINOS resolve uma seqüência de sub-problemas em que as restrições são linearizadas e a função objetivo é uma função Lagrangeana aumentada, o controle do tamanho de passo é heurístico mas a convergência superlinear é sempre alcançada [52]. O problema resolvido pelo MINOS tem a forma de um problema de PNL (4.4)

$$\begin{aligned} \min \quad & f(\mathbf{x}) + \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \\ \text{s.a.} \quad & \mathbf{h}(\mathbf{x}) + \mathbf{A}_1 \mathbf{y} = \mathbf{b}_1 \\ & \mathbf{A}_2 \mathbf{x} + \mathbf{A}_3 \mathbf{y} = \mathbf{b}_2 \end{aligned} \quad (4.4)$$

$$\mathbf{l} \leq \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \leq \mathbf{u}$$

Em que os vetores \mathbf{c} , \mathbf{d} , \mathbf{b}_1 , \mathbf{b}_2 , \mathbf{l} , \mathbf{u} e as matrizes \mathbf{A}_1 , \mathbf{A}_2 , \mathbf{A}_3 são constantes. O MINOS é um software vendido pela Stanford University Office of Technology Licensing, e é escrito em Fortran 77. (<http://www.sbsi-sol-optimize.com>)

4.1.5 SNOPT (Sparse Nonlinear OPTimizer)

O SNOPT utiliza um algoritmo de programação quadrática seqüencial. As direções de busca são obtidas dos sub-problemas de programação quadrática que minimiza um modelo quadrático da função Lagrangeana sujeito a restrições linearizadas. A função de mérito da Lagrangeana aumentada é reduzida ao longo de cada direção de busca para assegurar a convergência de qualquer ponto inicial. O problema resolvido pelo SNOPT tem a forma de um problema de PNL dado por (4.5).

$$\begin{aligned} \min & f(\mathbf{x}) \\ \text{s.a. } & \mathbf{l} \leq \begin{pmatrix} \mathbf{x} \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{A}\mathbf{x} \end{pmatrix} \leq \mathbf{u} \end{aligned} \quad (4.5)$$

Em que \mathbf{l} e \mathbf{u} são valores constantes, e representam os limites mínimo e máximo e \mathbf{A} é uma matriz esparsa. Informações sobre o SNOPT podem ser encontradas em [53] e é um software vendido pela Stanford University Office of Technology Licensing, escrito em Fortran. (<http://www.sbsi-sol-optimize.com>)

4.2 Perfil de Desempenho

Os perfis de desempenho fornecem meios eficazes de comparar de forma rápida o desempenho computacional dos programas de otimização aqui apresentados. Em [56] define-se o perfil de desempenho de um método como uma função de distribuição cumulativa para um desempenho métrico. Supondo resolver um conjunto de problemas P com diferentes métodos. Denotando um problema particular por p e um método particular por s . A idéia é comparar o desempenho computacional do método s resolvendo o problema p com o melhor desempenho computacional dado por algum método resolvendo este mesmo problema particular. A *taxa de desempenho* é definido como:

$$\rho_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : 1 \leq s \leq n_s\}}$$

Em que $t_{p,s}$ é o tempo total de processamento do programa de otimização s gasto no problema p . n_s é o número de programas de otimização. Se um programa de otimização não resolve um

determinado problema, o resultado de seu desempenho é representado por um valor infinito. Por outro lado, para obter uma avaliação total de um programa no conjunto problemas, é definida uma função de distribuição cumulativa $p_s(\tau)$ como:

$$p_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : \rho_{p,s} \leq \tau\}$$

em que $p_s(\tau)$ é a probabilidade da taxa de desempenho $\rho_{p,s}$ estar dentro de um fator de τ da melhor taxa de desempenho possível. n_p é o número de problemas. Em geral, $p_s(\tau)$ para um programa de otimização particular s fornece informações sobre a porcentagem de problemas que o programa de otimização solucionará se para cada problema, o programa de otimização pode ter um recurso de tempo máximo de τ vezes o tempo mínimo do tempo total de processamento. Para $\tau = 1$ a probabilidade $p_s(1)$ de um programa de otimização particular s é a probabilidade de um programa ganhar sobre todos os outros. Por isso, se estamos interessados somente em conhecer os atuais melhores programas de otimização, necessitamos somente considerar o valor de $p_s(1)$ de todos os programas de otimização. Para valores maiores de τ , a função de probabilidade $p_s(\tau)$ fornece informação se um programa de otimização consegue atualmente resolver um problema. Por isso, se estamos interessados somente na probabilidade que um programa de otimização consiga resolver um problema de forma satisfatória, deveríamos considerar $p_s(\tau)$ para todos os programas de otimização com um valor de τ maior.

Como os diferentes programas buscam somente uma solução local para vários problemas $p \in P$, e é muito provável que cada programa de otimização s encontre uma solução local diferente. De forma a fazer uma comparação correta, foram excluídos da comparação os problemas cujos valores finais da função objetivo não estão próximos ao critério (4.6).

$$\frac{\max\{l^1, \dots, l^{n_s}\} - \min\{l^1, \dots, l^{n_s}\}}{1 + \max\{|\max\{l^1, \dots, l^{n_s}\}|, |\min\{l^1, \dots, l^{n_s}\}|\}} > 10^{-1} \quad (4.6)$$

em que l^s é o valor da função objetivo do problema (3.1) obtida pelo programa s . O tempo total de processamento foi usado como uma medida do desempenho computacional; embora, a idéia acima possa ser utilizada com outra medida do desempenho computacional. Por exemplo, se agora o número de iterações é a medida do desempenho de interesse, ajuste $t_{p,s}$ corretamente.

4.3 Provas e Resultados

Os sistemas teste IEEE 30, 57, 118, e 300 barras, o sistema teste England 39 barras e três sistemas reais, o sistema peruano de 460 barras, um sistema real de 662 barras obtido em [60] e o sistema brasileiro de 2256 barras, foram utilizados para criar os casos usados para resolver o problema de FPOR. Para cada sistema elétrico de potência (SEP), foi calculado um conjunto de casos de contingências factíveis (para saídas de linhas de transmissão somente) cuja solução não resulta

Tab. 4.1: Características dos Sistemas Testados

Data	nb	ng	nsh	nl	nt	nzi	P	Q	nc
IEEE30	30	6	2	41	4	2	283.4	126.2	35
ENGL39	39	10	2	46	0	12	6097.1	1408.9	34
IEEE57	57	7	3	80	15	4	1250.8	336.4	59
IEEE118	118	54	8	186	9	2	3668.0	1438.0	167
IEEE300	300	69	10	411	50	28	23247.0	7788.0	248
PERU460	460	50	25	542	134	119	2959.2	1033.3	176
REAL662	662	73	44	1017	77	103	25827.0	8363.9	875
BRA2256	2256	201	86	3508	1	614	47079.8	15017.6	2807
TOTAL									4401

em um FPOR inactível. As contingências foram obtidas utilizando os critérios apresentados em [16]. A contingência é considerada factível se ela não gera corte de carga. O caso base, assim como o caso de contingências factíveis, foram usados para avaliar os programas de otimização.

A Tabela 4.1 mostra algumas informações relevantes dos sistemas usados nos testes. Onde nb é o número de barras, ng é o número de geradores, nsh é o número de elementos shunt com potência reativa variável, nl é o número de linhas de transmissão, nt é o número de transformadores com controle de posição de tap, nzi é o número de barras de passagem (barras sem injeção de potência ativa e reativa e sem conexão de transformador com tap variável), (P) e (Q) são as potência ativa e reativa total das cargas em MW e MVar, respectivamente, e nc é o número de casos de contingências factíveis.

Os resultados numéricos foram obtidos usando um SunFire V20Z com dois processadores AMD Opteron com 2.46Hz e 8Gb de memória RAM. Os seis software foram rodados com suas opções de “default” sobre 4401 problemas de PNL, todos na mesma máquina. O ponto inicial para as componentes (complexas) real e imaginária da magnitude de tensão nas barras são 1.0 pu e 0.0 pu, respectivamente e para os taps dos transformadores com taps variáveis é 1.0. Estes pontos iniciais foram utilizados por todos os problemas de PNL solucionados aqui. Como critério de parada foi imposto um limite máximo de 1000 iterações ao contador de iterações e um erro com tolerância mínima de 10^{-4} (erro do problema de PNL escalado [49]).

Dos 4401 problemas, 1528 foram excluídos usando o critério (4.6). A maioria de estes casos excluídos são casos factíveis do sistema Brasileiro de 2256. Dos 2873 casos restantes, o IPOPT não convergiu em um dos casos, o LOQO em 4 casos e o MINOS e o SNOPT em 6 casos, respectivamente. O KNITRO convergiu para todos os casos analisados. A Fig. 4.1 apresenta o desempenho computacional dos programas de otimização em relação ao número de iterações, e a

Fig. 4.2 compara os tempos computacional gastos para avaliar as funções do programa, considerando os tempos de: avaliação da função objetivo, avaliação das restrições, avaliação do gradiente da função objetivo, avaliação da matriz Jacobiana e avaliação da matriz Hessiana da função Lagrangeana. “KNITRO-ACA” representa o desempenho computacional do algoritmo de conjunto ativo implementado no KNITRO (algoritmo 3). Em termos de número de iterações e de tempo da avaliação das funções, o IPOPT e o KNITRO são os programas de otimização que mostram ser mais eficiente quando comparado ao LOQO, MINOS, SNOPT e KNITRO-ACA. Lembrando que o KNITRO usa algumas iterações do método de região de garantia, o que lhe custa um maior tempo computacional para convergir, e também que o KNITRO converge para todos os casos.

A Fig. 4.3 apresenta o perfil de desempenho do tempo total de processamento. Neste caso o IPOPT e o KNITRO também mostra um melhor tempo computacional quando comparado aos outros programas de otimização. Em geral os programas de otimização IPOPT, KNITRO e o LOQO apresentam um melhor desempenho computacional para resolver o problema de FPOR, comparados com os programas de otimização MINOS, SNOPT e KNITRO-ACA. Também pode-se observar que o LOQO apresentou um melhor desempenho computacional em relação ao tempo total de processamento para os três primeiros sistemas teste IEEE 14, 30 e 57 barras (sistemas pequenos). Para os sistemas reais (sistemas grandes) o IPOPT e o KNITRO foram os que apresentaram melhores resultados.

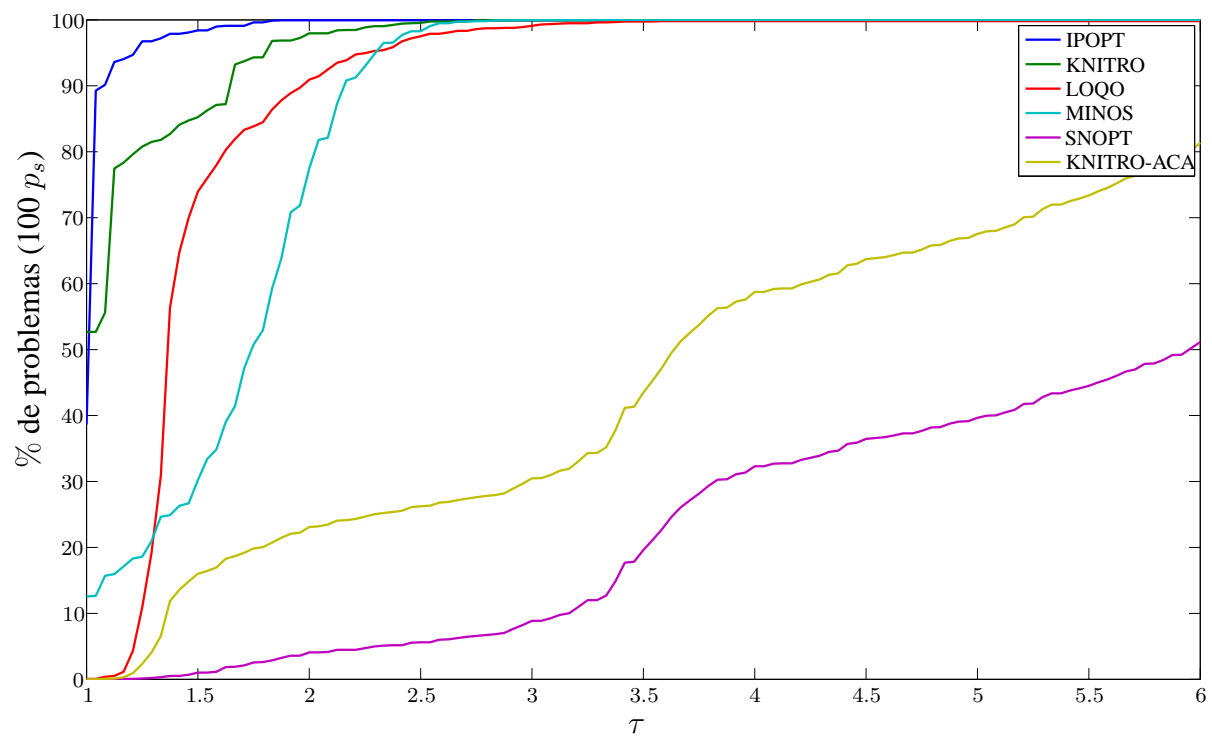


Fig. 4.1: Perfil de Desempenho para o Numero de Iterações

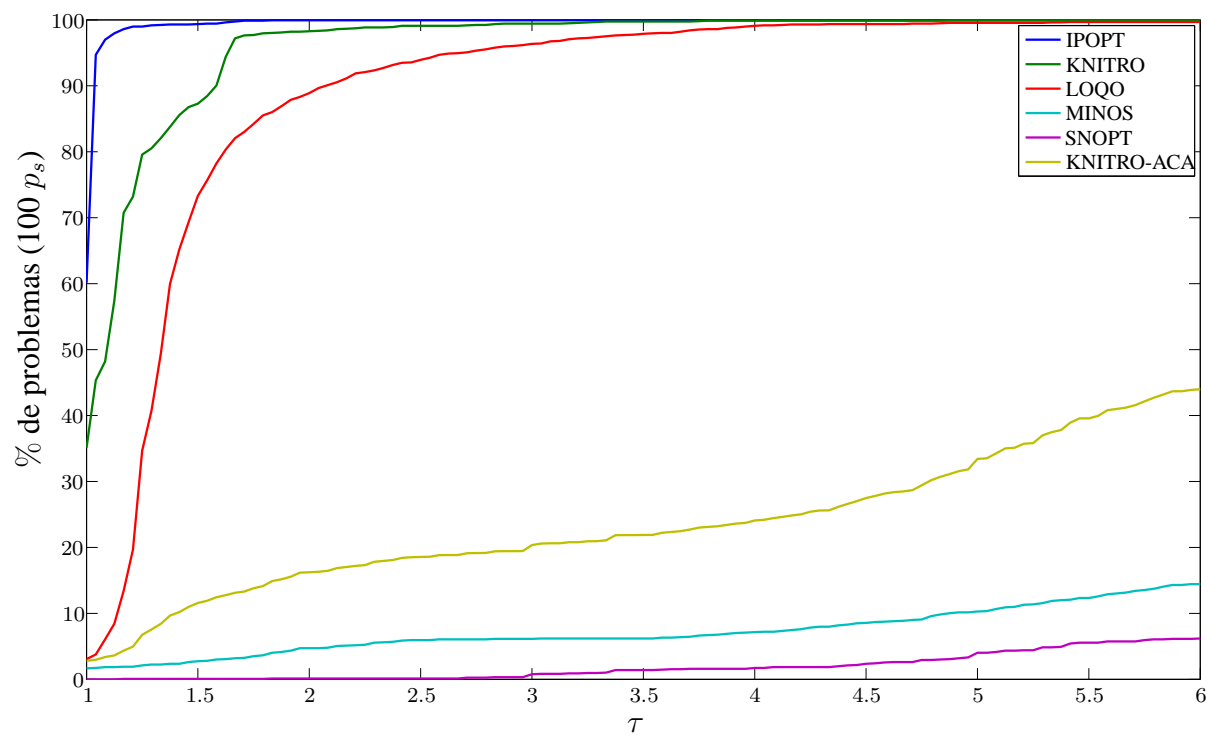


Fig. 4.2: Perfil de Desempenho para o Tempo de Avaliação de Funções

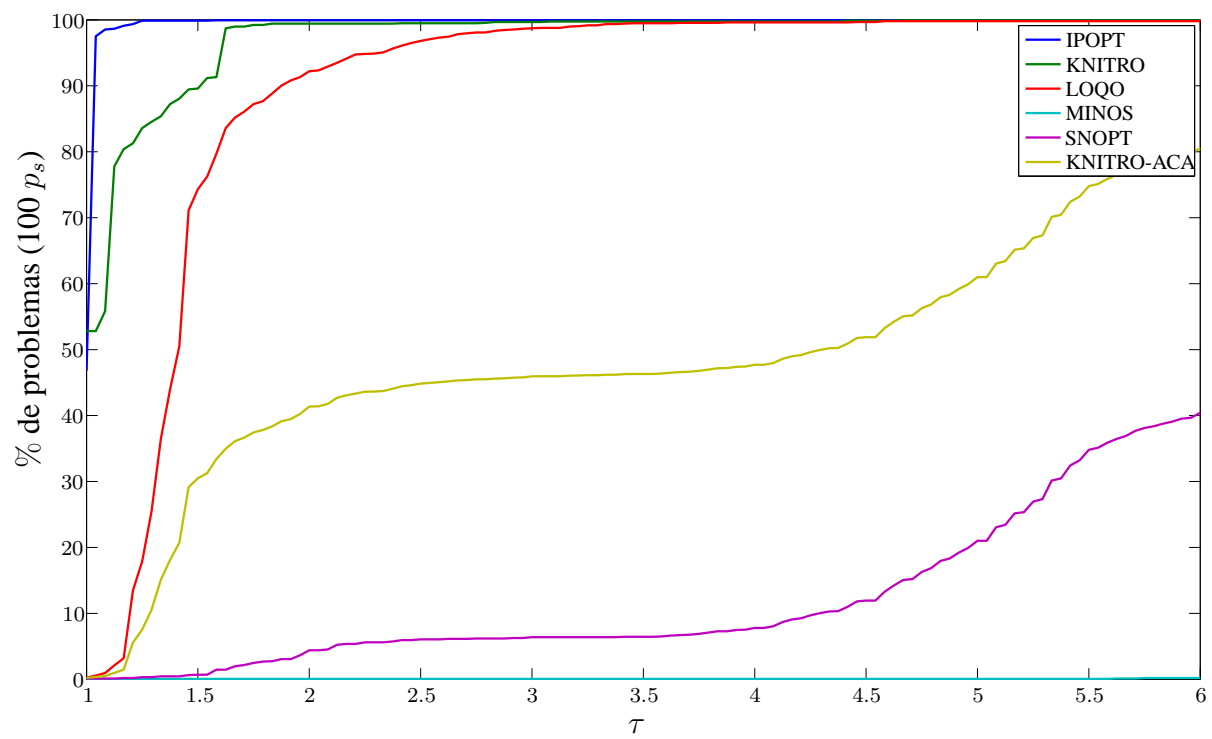


Fig. 4.3: Perfil de Desempenho para o Tempo Total de Processamento

Capítulo 5

Comparação Computacional entre as formulações do FPOR Utilizando Equações de Potência e Injeção de Corrente

Neste capítulo é mostrada uma comparação computacional entre as formulações do FPOR utilizando as equações de potência e injeção de corrente com tensões em coordenadas retangulares. Ambas formulações foram mostradas no Cap. 2. O programa de otimização IPOPT foi utilizado para fazer as comparações. Os cinco sistemas teste e os três sistemas reais, assim como todas as contingências factíveis apresentados no Cap. 4 foram utilizados para avaliar as duas formulações.

Como foi apresentado no Cap. 1, o maior esforço computacional realizado pelo método de pontos interiores é formar, fatorar e resolver, em cada iteração, o sistema de equações lineares dado por (2.6) (ver o passo 7 do algoritmo geral). Para melhorar o desempenho computacional do MPI é necessário analisar a influência de ambos os modelos ((3.1) e (3.21)) na estrutura da matriz \mathbf{W} . Já foi visto no capítulo 2 que usando o modelo (3.21) obtemos elementos constantes na matriz Jacobiana das restrições de igualdade \mathbf{J}_g e uma matriz Hessiana da Lagrangeana altamente esparsa.

Os problemas de FPOR mostrados em (3.1) e (3.21) foram modelados como um problema de PNL padrão, escritos em AMPL [55] e testado em cada um dos sistemas apresentados no Cap. 4. Os resultados numéricos foram obtidos utilizando o mesmo computador apresentado no Capítulo 4. O software IPOPT foi rodado com as mesmas opções de “default” nos 4401 problemas de PNL e na mesma máquina para ambos os modelos. O ponto inicial para as componentes real e imaginária das tensões de barra são 1.0 pu e 0.0 pu, respectivamente, para os taps dos transformadores com tap os valores são iguais a 1.0 pu, para a potência ativa gerada na barra de referência é 0.0 pu e para as potências reativas de geração dos geradores e compensadores shunt é 0.0 pu. Este ponto inicial foi utilizado para todos os problemas de PNL resolvidos. O limite imposto ao contador de iterações foi de 1000 iterações e o erro de tolerância máximo foi 10^{-4} (erro do problema de PNL escalado [49]).

A Tabela 5.1 mostra um resumo das características de ambos os modelos para o FPOR. Onde

Tab. 5.1: Resumo das Características de Otimização

Data		nx	$nxlu$	ngl	$ngnl$	nh	nJ	nH
IEEE30	FPOR-IC	72	12	4	56	30	497	128
	FPOR-EP	63	4	0	51	38	512	269
ENGL39	FPOR-IC	88	10	24	54	39	602	134
	FPOR-EP	77	0	0	67	49	593	297
IEEE57	FPOR-IC	139	25	8	106	57	965	262
	FPOR-EP	128	15	0	103	67	1006	550
IEEE118	FPOR-IC	307	71	4	232	118	2250	575
	FPOR-EP	244	9	0	173	180	2146	1100
IEEE300	FPOR-IC	729	129	56	544	300	5160	1333
	FPOR-EP	649	50	0	520	379	5263	2790
PERU460	FPOR-IC	1129	209	238	682	460	7167	2096
	FPOR-EP	1053	134	0	844	535	7263	4086
REAL662	FPOR-IC	1517	193	206	1118	662	11744	2596
	FPOR-EP	1400	77	0	1207	778	11580	6022
BRA2256	FPOR-IC	4800	288	1228	3284	2256	33732	7018
	FPOR-EP	4512	1	0	4224	2543	36395	18201

nx é o número de variáveis primais, $nxlu$ é o número de variáveis primais com limites mínimos e máximos, ngl é o número de restrições de igualdade lineares, $ngnl$ é o número de restrições de igualdade não lineares, nh é o número de restrições de desigualdade, nJ é o número de elementos diferentes de zero da matriz Jacobiana (J_g) e nH é o número de elementos diferentes de zero da diagonal superior da Matriz \mathbf{W} (sendo \mathbf{W} uma matriz simétrica, o valor nH é usado como seu número de elementos diferentes de zero). Note que o número de restrições de igualdade lineares (ngl) que aparecem no modelo (3.21) é igual a duas vezes o número de barras de passagem (nzi), e também que a diferença do número de elementos diferentes de zero entre as matrizes Hessianas dos dois modelos é de aproximadamente 50%. Em alguns casos o número de elementos diferentes de zero da matriz Jacobiana para o modelo (3.21) é maior do que para o modelo (3.1), porém o número total de elementos diferentes de zero da matriz \mathbf{W} para o modelo (3.21) é de aproximadamente 19% menor que o modelo (3.1).

A tabela 5.2 mostra o tempo total, o tempo de avaliação das funções e o número de iterações (para ambos os modelos) para o caso base de todos os sistemas testados, além do valor do ganho percentual entre os dois modelos. Onde PRC é dado pela Eq. (5.1), em que t_{CI} e t_{EP} é o tempo

Tab. 5.2: Tempo total e de avaliação das funções do problema para o caso base

Systems	Total CPU Time			Evaluation CPU Time			Iterations	
	t_{CI} (sec)	t_{EP} (sec)	PRC (%)	t_{CI} (sec)	t_{EP} (sec)	PRC (%)	FPOR-IC	FPOR-EP
IEEE30	0.016	0.036	55.56	0.001	0.013	92.31	10	10
ENGL39	0.030	0.050	40.00	0.004	0.024	83.33	10	11
IEEE57	0.032	0.073	56.16	0.007	0.039	82.05	11	11
IEEE118	0.096	0.132	27.27	0.033	0.095	65.26	16	12
IEEE300	0.196	0.455	56.92	0.079	0.347	77.23	20	17
PERU460	0.439	0.461	4.77	0.260	0.345	24.64	28	15
REAL662	0.409	0.920	55.54	0.185	0.701	73.61	21	16
BRA2256	1.912	3.196	40.18	0.848	2.316	63.39	36	23

utilizado pela formulação do FPOR-IC e FPOR-EP, respectivamente. Note que o tempo total de processamento para o modelo (3.21) é menor que para o modelo (3.1), principalmente para o tempo gasto na avaliação das funções do problema.

$$\text{PRC} = \frac{(a - b)}{a} 100\% \begin{cases} \text{se } t_{CI} \geq t_{EP} \text{ então} \\ a = t_{CI} \text{ e } b = t_{EP}; \\ \text{caso contrário} \\ b = t_{EP} \text{ e } a = t_{CI}. \end{cases} \quad (5.1)$$

A tabela 5.3 mostra o tempo total de processamento gasto para fazer a análise de contingências para cada sistema testado, e o tempo total de processamento de todas as 4401 contingências simuladas para ambos os modelos apresentados. Os resultados mostram que existe um considerável ganho de tempo total de processamento quando o modelo (3.21) é comparado com o modelo (3.1). A Fig. (5.2) e (5.1) mostra o valor do PRC para o tempo de processamento total e o tempo de avaliação de funções, respectivamente, entre ambos modelos e para cada contingência. Os valores positivos representam a porcentagem de tempo ganho pelo modelo (3.21) em relação ao modelo (3.1). As barras com valores negativos representam a porcentagem de tempo ganho pelo modelo (3.1) em relação ao modelo (3.21). Observa-se que o tempo de processamento gasto pelo modelo (3.21) para cada contingência é consideravelmente menor que o tempo de processamento gasto pelo modelo (3.1), tanto para o tempo total de processamento quanto para o tempo de avaliação das funções do problema. Observa-se também que, o modelo (3.1) somente apresentou ganho de tempo total de processamento em relação ao modelo (3.21) para apenas sete contingências do Sistema Peruano.

Tab. 5.3: Tempo computacional total gasto pela análise das contingências

Method	t_{CI} (sec)	t_{EP} (sec)	PRC (%)
IEEE30	0.603	1.065	43.43
ENGL39	0.695	1.395	50.19
IEEE57	2.003	3.983	49.72
IEEE118	12.591	21.555	41.59
IEEE300	47.410	88.559	46.36
PERU460	53.801	80.755	33.37
REAL662	332.563	749.440	55.63
BRA2256	5354.376	8897.707	39.82
Total	5804.042	9844.459	41.04

Os resultados mostram que o modelo (3.21) gasta um tempo de processamento total e de avaliação de suas funções menor que o modelo (3.1) na grande maioria dos 4401 casos, obtendo um ganho computacional total de aproximadamente 41%. Também, outro fato a considerar é que os modelos (3.1) e (3.21) possuem características de convergência diferentes, podendo o modelo (3.21) gastar um número maior de iterações para convergir em relação ao modelo (3.1), ver Tabela 5.2. Esta característica também faz com que o modelo (3.21) convirja para todos os 4401 casos, enquanto que o modelo (3.1) não converge para um dos caso. A seção 3.4 mostrou que o modelo (3.21) requer um menor cálculo de elementos para a matriz (2.6) tornando a construção desta menos complexa que a da mesma matriz para o modelo (3.1).

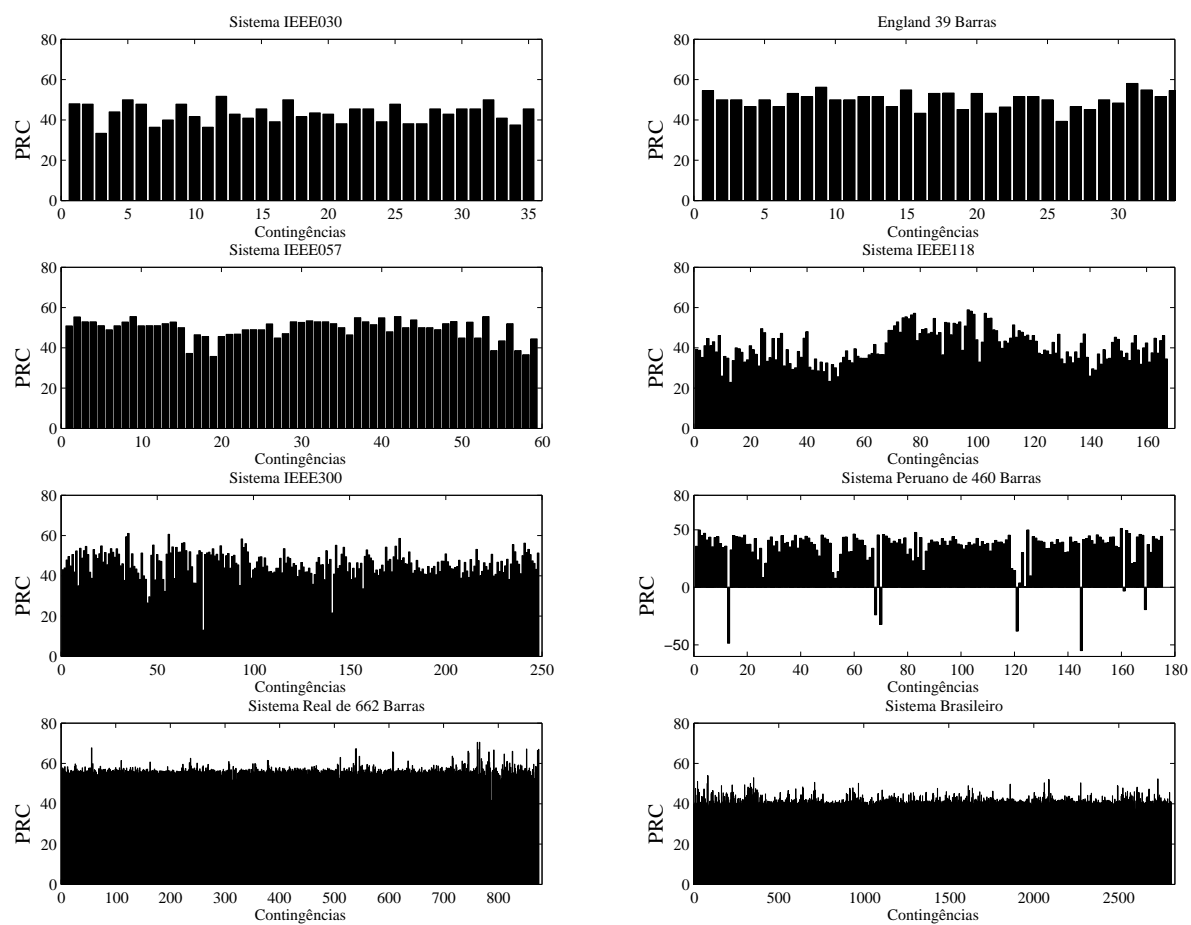


Fig. 5.1: Tempo computacional total gasto para cada contingência

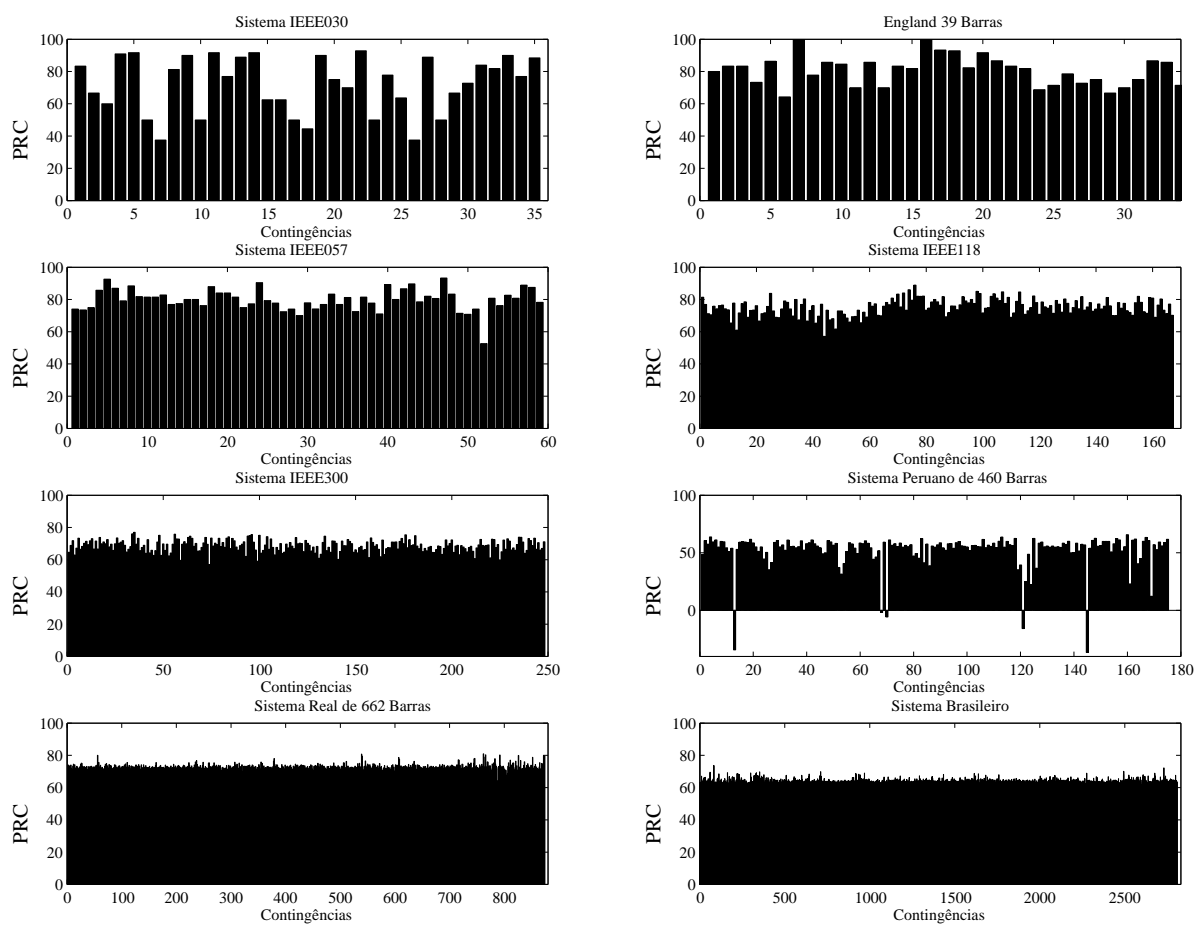


Fig. 5.2: Tempo de avaliação das funções do problema para cada contingência

Capítulo 6

Método de Pontos Interiores com Técnicas de Região de Garantia

Neste capítulo será apresentado um método de pontos interiores que associa técnicas de região de garantia para encontrar um passo ótimo a cada iteração do algoritmo. Este método é baseado no programa de otimização KNITRO (apresentado no Cap. 4) e foi testado utilizando a formulação do FPOR-PE apresentada na seção 3.1.

A técnica de região de garantia é uma classe de algoritmos de otimização para resolver problemas não convexos e não-lineares. A idéia principal do método está baseada no fato de que a utilização do método de região de garantia sozinho apresenta a vantagem de não ter que fatorar a matriz Hessiana da função Lagrangeana a cada iteração, o que é eficiente quando se resolve um problema de grande porte, porém, apresenta a desvantagem de se torna um método com alto custo computacional quando a matriz Hessiana da função Lagrangeana é mal condicionada [61].

Foi então desenvolvido um algoritmo que calcula um passo ótimo utilizando um método de pontos interiores sempre que a qualidade desses passos possa ser assegurada, caso contrário o passo é calculado utilizando a região de garantia. Isto pode ser implementado em um só algoritmo desde que o método de pontos interiores e a técnica de região de garantia sejam utilizadas independentes uma da outra.

Este algoritmo resolve um PNL do tipo mostrado pela Eq. (4.2). A direção de busca é calculada neste algoritmo pela fatoração de um sistema primal-dual. No intuito de se conseguir uma robustez matemática, esta a busca linear levanta duas questões importantes: a) Como definir a direção de busca quando o modelo usado pelo algoritmo é não convexo; e b) Como tratar o rank deficiente da matriz Hessiana da Lagrangeana e da Jacobiana das restrições.

Em [61] é descrito também, um mecanismo para estabilizar a busca linear de cada iteração, que consiste em voltar, sob determinadas condições, em um passo da região de garantia que seja confiável, tal que este caminhe na direção da factibilidade e da otimalidade. Um desafio é projetar o algoritmo de modo que haja uma transição direta entre o método de pontos interiores e as etapas da região de garantia. E afirma que o algoritmo apresentado não possui um custo computacional mais elevado do que outros métodos, tendo propriedades de convergência favoráveis, e boa eficiência

computacional para os problemas testados por este.

6.1 Formulação do Problema

O problema de PNL testado com o método de pontos interiores com técnicas de região de garantia é o problema dado pela Eq. (3.1). A função objetivo do problema adicionando o fator de penalização das variáveis de folga não-negativas:

$$f_\mu(\mathbf{x}) = \mathbf{G}_{sw}:(e_{sw}\mathbf{e} + f_{sw}\mathbf{f}) + \mathbf{B}_{sw}:(f_{sw}\mathbf{e} - e_{sw}\mathbf{f}) - \mu \sum_{i=1}^{nh} \ln \mathbf{s}_i \quad (6.1)$$

A restrição de desigualdade é:

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{G}_k:(e_k\mathbf{e} + f_k\mathbf{f}) + \mathbf{B}_k:(f_k\mathbf{e} - e_k\mathbf{f}) - P_k^{SP} \\ \mathbf{G}_k:(f_k\mathbf{e} - e_k\mathbf{f}) - \mathbf{B}_k:(e_k\mathbf{e} + f_k\mathbf{f}) - Q_k^{SP} \end{bmatrix} \quad (6.2)$$

A variável canalizada existente no problema (3.1) é considerada como uma restrição de desigualdade:

$$\mathbf{h}'(\mathbf{x}) = \begin{bmatrix} \mathbf{G}_k:(f_k\mathbf{e} - e_k\mathbf{f}) - \mathbf{B}_k:(e_k\mathbf{e} + f_k\mathbf{f}) \\ e_k^2 + f_k^2 \\ \underline{tap}_m \end{bmatrix} \quad (6.3)$$

A restrição de desigualdade $\mathbf{h}(\mathbf{x})$ da Eq. 2.3 é:

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \mathbf{h}'(\mathbf{x}) - \mathbf{h}^u(\mathbf{x}) \\ \mathbf{h}^l(\mathbf{x}) - \mathbf{h}'(\mathbf{x}) \end{bmatrix} \quad (6.4)$$

em que

$$\mathbf{h}^u(\mathbf{x}) = \begin{bmatrix} \overline{Q}_k \\ \overline{V}_k^2 \\ \overline{tap}_m \end{bmatrix} \quad (6.5)$$

e

$$\mathbf{h}^l(\mathbf{x}) = \begin{bmatrix} \underline{Q}_k \\ \underline{V}_k^2 \\ \underline{tap}_m \end{bmatrix} \quad (6.6)$$

onde $\mathbf{h}^u(\mathbf{x})$ e $\mathbf{h}^l(\mathbf{x})$ são os vetores que contém os valores dos limites mínimo e máximo, respectivamente, das potências reativas, tensões e taps de um sistema.

Assim pode-se adaptar o problema dado por (3.1) ao problema dado por (2.3), sendo a função Lagrangeana, as condições de otimalidade de KKT e o sistema de Newton dados pelas Eqs. (2.4), (2.5) e (2.6), respectivamente.

6.2 Função de Mérito

A função de mérito é dada por:

$$\phi_v(\mathbf{z}) = f_\mu(\mathbf{z}) + v\|\mathbf{c}(\mathbf{z})\| \quad (6.7)$$

onde $f_\mu(\mathbf{z})$ é a função de barreira definida pela Eq. (6.1), $\mathbf{z} = (\mathbf{x}, \mathbf{s})$, e

$$\mathbf{c}(\mathbf{z}) = \begin{bmatrix} \mathbf{g}(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) + \mathbf{s} \end{bmatrix} \quad (6.8)$$

e $v > 0$ é um parâmetro de penalidade que é atualizado a cada iteração, tal que a direção de busca \mathbf{d}_z seja uma direção de descida para ϕ_v .

$$\mathbf{d}_z = \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{s} \end{bmatrix} \quad (6.9)$$

A atualização do parâmetro v é baseada no método de região de garantia. Ao invés de exigir somente que a derivada direcional de ϕ_v seja negativa, como normalmente é feito, foi escolhido um parâmetro v baseado no decrescimento do modelo quadrático/linear da função de mérito calculado em função do passo \mathbf{d}_z , e é dado por:

$$v^+ = \begin{cases} v & \text{se } v \geq v_T \\ v_T + 1 & \text{caso contrário} \end{cases} \quad (6.10)$$

sendo

$$v \geq \frac{\nabla f_\mu(\mathbf{z})^T \mathbf{d}_z + \frac{\sigma}{2} \mathbf{d}_z^T \mathbf{W}_z \mathbf{d}_z}{(1 - \rho)\|\mathbf{c}(\mathbf{z})\|} \equiv v_T \quad (6.11)$$

o parâmetro $\rho \in (0, 1)$. Neste trabalho foi utilizado $\rho = 1$.

$$\sigma = \begin{cases} 1 & \text{se } \mathbf{d}_z^T \mathbf{W}_z \mathbf{d}_z > 0 \\ 0 & \text{caso contrário} \end{cases} \quad (6.12)$$

e

$$\mathbf{W}_z = \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1}\mathbf{\Pi} \end{bmatrix} \quad (6.13)$$

Para mostrar que a escolha de v garante que \mathbf{d}_z é uma direção de descida de ϕ_v , notou-se [61] que (6.10) e a definição de σ implicam que:

$$\nabla \mathbf{f}_\mu(\mathbf{z})^T \mathbf{d}_z - v \|\mathbf{c}(\mathbf{z})\| \leq -\rho v \|\mathbf{c}(\mathbf{z})\| \quad (6.14)$$

Desde que a derivada direcional de ϕ_v em relação a \mathbf{d}_z seja dada por:

$$D\phi_v(\mathbf{z}; \mathbf{d}_z) = \nabla \mathbf{f}_\mu^T(\mathbf{z}) \mathbf{d}_z - v \|\mathbf{c}(\mathbf{z})\| \quad (6.15)$$

6.3 Critério de Convergência

6.3.1 Critério de Convergência Geral

Idealmente, o critério de parada deveria ser independente da escala das variáveis, da função objetivo e das restrições do problema. Porém, na prática, é difícil de conseguir sempre uma escala independente, e em casos onde certas quantidades se aproximam de zero, esta escala independente não é desejável. Desta forma usou-se uma técnica que alcança um equilíbrio entre a praticidade e a independência. Baseado na Eq. (2.13), porém utilizando um fator de escala que é calculado ao final de cada iteração geral como mostram as Eqs. (6.16), (6.17) e (6.18).

- Factibilidade Primal

$$\|(\mathbf{g}(\mathbf{x}^k), \mathbf{h}(\mathbf{x}^k)^+)\|_\infty \leq \max\{1, \|\mathbf{g}(\mathbf{x}^0), \mathbf{h}(\mathbf{x}^0)^+\|_\infty\} \epsilon^f \quad (6.16)$$

- Factibilidade Dual

$$\frac{\|\nabla \mathbf{f}(\mathbf{x}^k) + \mathbf{J}_g(\mathbf{x}^k)^T \boldsymbol{\lambda}^k + \mathbf{J}_h(\mathbf{x}^k)^T \boldsymbol{\pi}^k\|_\infty}{1 + \|\mathbf{x}^k\|_\infty} \leq \max\{1, \|\nabla \mathbf{f}(\mathbf{x}^k)\|_\infty\} \epsilon^o \quad (6.17)$$

- Condição de Optimalidade

$$\|\mathbf{S}\boldsymbol{\pi}\|_\infty \leq \max\{1, \|\nabla \mathbf{f}(\mathbf{x}^k)\|_\infty\} \epsilon^o \quad (6.18)$$

Onde $\mathbf{h}(\mathbf{x}^0)^+ = \max\{0, \mathbf{h}(\mathbf{x}^k)\}$ e \mathbf{x}^0 é o valor inicial da variável \mathbf{x} . Quando o valor máximo não é igual a 1, o fator de escala da Eq. (6.17) faz com que este teste de convergência seja independente para qualquer escala de $f(\mathbf{x}^k)$, $\mathbf{c}(\mathbf{x}^k)$ e de qualquer mudança linear da variável x^k . O fator 1 é necessário para assegurar o teste de convergência de factibilidade dual quando $\|\nabla \mathbf{f}(\mathbf{x}^k)\|_\infty$ é igual a zero ou possui um valor muito pequeno.

6.3.2 Teste de Convergência do Problema de Barreira

Como dito no Cap. 3, a sequência de parâmetros de barreira $\{\mu\}$ deve decrescer monotonicamente até zero, para isto foi utilizado uma estratégia simples para a atualização de μ que é dado da seguinte forma:

$$\mu^{k+1} = \begin{cases} \frac{\mu^k}{100} & \text{se } k \leq 3 \\ \frac{\mu^k}{5} & \text{caso contrário.} \end{cases} \quad (6.19)$$

Para o problema de barreira logaritmica o teste de convergência é feito em função do parâmetro de barreira μ . As tolerâncias são calculadas proporcionalmente ao decrescimento do valor de μ e é dado por:

- Factibilidade Primal

$$\|(\mathbf{g}(\mathbf{x}^k), \mathbf{h}(\mathbf{x}^k) + \mathbf{s}^k)\|_\infty \leq \max\{1, \|\mathbf{g}(\mathbf{x}^0), \mathbf{h}(\mathbf{x}^0)^+\|_\infty\} \epsilon_\mu^f \quad (6.20)$$

- Factibilidade Dual

$$\frac{\|\nabla \mathbf{f}(\mathbf{x}^k) + \mathbf{J}_g(\mathbf{x}^k)^T \boldsymbol{\lambda}^k + \mathbf{J}_h(\mathbf{x}^k)^T \boldsymbol{\pi}^k\|_\infty}{1 + \|\mathbf{x}^k\|_\infty} \leq \max\{1, \|\nabla \mathbf{f}(\mathbf{x}^k)\|\} \epsilon_\mu^o \quad (6.21)$$

- Condição de Otimalidade

$$\|\mathbf{S}\boldsymbol{\pi} - \mu \mathbf{e}\|_\infty \leq \max\{1, \|\nabla \mathbf{f}(\mathbf{x}^k)\|\} \epsilon_\mu^o \quad (6.22)$$

Note que os fatores de escala calculados nos critérios de convergência geral e de barreira são iguais, o que muda nos dois critérios é o lado esquerdo das equações que se diferem pelas somas de $-\mu \mathbf{e}$ na Eq. (6.22) e de $\mathbf{g}(\mathbf{x}) + \mathbf{s}$ na Eq. (6.20), em que:

$$\begin{aligned} \epsilon_\mu^o &= \max\{\theta\mu, \epsilon^o - \mu\} \\ \epsilon_\mu^f &= \max\{\theta\mu, \epsilon^f\} \end{aligned} \quad (6.23)$$

Isto assegura que a escolha da tolerância não cause um número muito grande de soluções do subproblema de barreira, porém não assegura um limite mínimo para μ . Embora um valor muito pequeno de μ não causar um excessivo número de soluções do subproblema de barreira, ainda sim é necessário prevenir um valor muito pequeno de μ , pois isto pode causar falha na iteração. Baseado na tolerância do problema geral foi forçado um valor mínimo do parâmetro de barreira dado por:

$$\mu_{\min} = \frac{\min\{\epsilon^o, \epsilon^f\}}{100} \quad (6.24)$$

6.4 Transição Entre o Método de Pontos Interiores e de Região de Garantia

Se o método de pontos interiores não calcula um passo para o subproblema de barreira, o algoritmo de região de garantia é então utilizado. Para a utilização deste algoritmo é desejável que se dê a ele uma informação do tamanho do raio da região de garantia, tal que este reflita as informações da iteração corrente. Este fato é particularmente importante quando dois passos de região de garantia são separados por uma grande sequência de passos de busca linear. Quando a busca linear calcula uma direção d_z^k , o raio da região de garantia é calculado da seguinte maneira:

$$\Delta^{k+1} = 2\alpha_z^k \|d_z^k\| \quad (6.25)$$

Por outro lado, se a direção de descida mais recente foi calculada por um passo da região de garantia, ou um passo de busca linear foi rejeitado, o raio da região de garantia é atualizado de acordo com a regra padrão de região de garantia.

6.5 Algoritmo Geral

O método de pontos interiores primal – dual com técnicas de região de garantia pode ser sintetizado no seguinte algoritmo:

1. Inicializar $\mu^0, \eta > 0, 0 < \delta < 1, imax$, obter um ponto inicial $(\mathbf{x}^0, \mathbf{s}^0, \boldsymbol{\lambda}^0, \boldsymbol{\pi}^0)$ de acordo com a seção 2.2.6, tal que sejam satisfeitas as condições de não-negatividade e fazer $k = 0$.

Repetir

2. Até que os critérios de convergência sejam satisfeitos
3. Calcular $\mathbf{g}(\mathbf{x}^0), \mathbf{h}(\mathbf{x}^0), \mathbf{Jg}(\mathbf{x}^0)$ e $\mathbf{Jh}(\mathbf{x}^0)$.
4. Calcular o vetor da parte direita da expressão (2.6).

Início do Problema de Barreira Logarítmica

Repetir

5. Fazer *busca linear* = falso.
6. Calcular a matriz \mathbf{A} usando (2.8).
7. Verificar se \mathbf{W}_z é definida positiva (ou seja, $\mathbf{d}_z^T \mathbf{W}_z \mathbf{d}_z > 0$), e ir ao passo 8, senão ir ao passo 16.
8. Montar e resolver no ponto atual o sistema (2.6).

9. Obter o máximo tamanho de passo usando (2.10) e chamar $\alpha_z^{max} = \alpha_p$ e $\alpha_\lambda^{max} = \alpha_d$.
10. Verificar se $\alpha_k > \delta$, e ir ao passo 11, senão ir ao passo 16.

Repetir

12. Enquanto ($j \leq imax$), ($\alpha_T > \delta$) e *busca linear = falso*,
 13. Verificar se $\phi_v(\mathbf{z}_k + \alpha_T \alpha_z^{max} \mathbf{d}_z) \leq \phi_v(\mathbf{z}_k) + \eta \alpha_T \alpha_z^{max} D\phi_v(\mathbf{z}_k; \mathbf{d}_z)$, e ir ao passo 14, senão, ir ao passo 15.
 14. Fazer $\alpha_z = \alpha_T \alpha_z^{max}$, $\alpha_\lambda = \alpha_T \alpha_\lambda^{max}$, atualizar as variáveis utilizando a Eq. (2.9), calcular Δ_{k+1} e fazer *busca linear = verdadeiro* e voltar ao passo 12.
 15. Fazer $j = j + 1$ e $\alpha_T = \frac{\alpha_T}{2}$

Até que uma das condições seja violada.

16. Verificar se *busca linear = falso*, e ir ao passo 17, senão passe ao passo 18.
17. Calcular \mathbf{x}^{k+1} , \mathbf{s}^{k+1} , $\boldsymbol{\lambda}^{k+1}$ e $\boldsymbol{\pi}^{k+1}$ e atualizar Δ^{k+1} utilizando o método de região de garantia.
18. Fazer $\mu^{k+1} = \mu^k$ e $k = k + 1$.

Até se obter a convergência do problema de barreira

19. Atualizar μ^k (2.12),

Até se obter a convergência do problema geral

Os valores dos parâmetro utilizados nas simulações foram, $\mu^0 = 1$, $\eta = 10^{-8}$, $\delta = 10^{-5}$, $imax = 3$.

6.6 Método de Região de Garantia

O método de região de garantia utilizado, descrito em [33], foi desenvolvido para resolver os problemas de PNL de grande porte, onde um MPI-TRG é utilizado para encontrar um passo ótimo para o subproblema de barreira logaritmica, quando a busca linear falha. O TRG calcula separadamente, as variáveis primais e duais, diferente de quando é utilizada o método de pontos interiores primal – dual.

Os multiplicadores de Lagrange para o TRG são calculados utilizando uma aproximação por mínimos quadrados. Isto é utilizado em alguns problemas de PQS existentes na literatura, que calcula uma estimação por mínimos quadrados baseado nas condições de estacionaridade da iteração corrente, assim, são escolhidos os multiplicadores de Lagrange que minimizam a norma euclidiana da primeira e última equações do conjunto de equações (2.5), e são dados por:

$$\begin{bmatrix} \boldsymbol{\lambda}^k \\ \boldsymbol{\pi}^k \end{bmatrix} = [(\hat{\mathbf{J}}^k)^T \hat{\mathbf{J}}^k]^{-1} (\hat{\mathbf{J}}^k)^T \begin{bmatrix} -\nabla \mathbf{f}(\mathbf{x}^k) \\ \boldsymbol{\mu}^k \mathbf{e} \end{bmatrix} \quad (6.26)$$

Onde

$$\hat{\mathbf{J}} = \begin{bmatrix} \mathbf{J}_{\mathbf{g}}(\mathbf{x}) & \mathbf{J}_{\mathbf{h}}(\mathbf{x}) \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \quad (6.27)$$

O maior esforço computacional para calcular este sistema aumentado é a inversão da matrix $(\hat{\mathbf{J}}^k)^T \hat{\mathbf{J}}^k$. Porém, deste modo não se pode garantir, que o valor de π^k será sempre positivo. Para garantir que π^k seja positivo, foi adotado uma forma na qual não se pretende forçar que π^k seja positivo, mas assegurar que o modelo quadrático permaneça convexo nas variáveis de folga. Dessa forma Σ^k que no problema original é dado por:

$$\Sigma_i^k = (\mathbf{S}^k)^{-1} \Pi \quad (6.28)$$

agora será uma matriz diagonal composta por:

$$\sigma_i^k = \begin{cases} \pi_i^k / s_i & \text{se } \pi_i^k > 0 \\ \mu^k / s_i^2 & \text{caso contrário} \end{cases} \quad (6.29)$$

6.7 Função de Mérito para a Região de Garantia

A função de mérito $\phi(\mathbf{x}, \mathbf{s}, \nu)$, definida por (6.1), é utilizada aqui, para determinar se o passo total $\mathbf{d} = \mathbf{v} + \boldsymbol{\omega}$ é aceitável, e também dar informação sobre como atualizar o raio da região de garantia Δ . O parâmetro de penalidade ν (para não ser confundido com o parâmetro de barreira μ) equilibra a relativa contribuição da função objetivo e das restrições, e necessita ser atualizada a cada iteração tal que o passo \mathbf{d} e a função de mérito ϕ sejam compatíveis. Isto significa que se a região de garantia é suficientemente pequena, então o passo \mathbf{d} deveria causar uma redução em ϕ .

Aproxima-se a mudança da função de mérito devido ao passo \mathbf{d} pela *Redução Prevista* definida como:

$$pred^k(\mathbf{d}) = -q(\tilde{\mathbf{v}} + \tilde{\mathbf{w}}) + \nu^k v pred^k \quad (6.30)$$

Onde $v pred^k$ é a redução prevista pelo passo normal, e q é a função objetivo do passo horizontal ou tangencial do subproblema quadrático que é inerente à formulação de região de garantia e que é discutido em [33], q é dado por:

$$q(\tilde{\mathbf{v}} + \tilde{\mathbf{w}}) = (\nabla \mathbf{f}^T - \mu \mathbf{e}^T)(\tilde{\mathbf{v}} + \tilde{\mathbf{w}}) + \frac{1}{2}(\tilde{\mathbf{v}} + \tilde{\mathbf{w}})^T \mathbf{G}(\tilde{\mathbf{v}} + \tilde{\mathbf{w}}) \quad (6.31)$$

Onde \mathbf{e} é um vetor de tamanho apropriado formado por elementos iguais a 1 e a matriz \mathbf{G} é dada por:

$$\mathbf{G} = \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}\Sigma\mathbf{S} \end{bmatrix} \quad (6.32)$$

E $\tilde{\mathbf{v}} = (\mathbf{v}_x, \mathbf{S}^{-1}\mathbf{v}_s)$, \mathbf{v} é chamado de passo normal e $\tilde{\mathbf{w}}$ é a tangente do gradiente das restrições, o cálculo detalhado é dado em [33].

$$vpred^k(\mathbf{d}) = \left\| \begin{bmatrix} \mathbf{g}(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) + \mathbf{s} \end{bmatrix} \right\| - \left\| \begin{bmatrix} \mathbf{g}(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) + \mathbf{s} \end{bmatrix} + \hat{\mathbf{J}}^T \tilde{\mathbf{v}} \right\| \quad (6.33)$$

A definição de (6.30) é baseada na análise apresentada por [59]. Neste problema é necessário que ν^k seja grande o bastante para que o $pred^k(\mathbf{d})$ seja positivo e proporcional à redução $\sqrt{-m(\tilde{\mathbf{v}})}$ fornecida pelo passo vertical ou passo normal, e é dado por:

$$m(\tilde{\mathbf{v}}) = 2 \begin{bmatrix} \mathbf{g}^T & (\mathbf{h} + \mathbf{s})^T \end{bmatrix} \hat{\mathbf{J}}^T \begin{bmatrix} \mathbf{v}_x \\ \tilde{\mathbf{v}}_s \end{bmatrix} + \begin{bmatrix} \mathbf{v}_x^T & \tilde{\mathbf{v}}_s^T \end{bmatrix} \hat{\mathbf{J}}\hat{\mathbf{J}}^T \begin{bmatrix} \mathbf{v}_x \\ \tilde{\mathbf{v}}_s \end{bmatrix} \quad (6.34)$$

Assim,

$$pred^k(\mathbf{d}) \geq \rho \nu^k vpred \quad (6.35)$$

Onde $0 < \rho < 1$, para este trabalho foi escolhido um valor de $\rho = 0.3$. Pode-se observar em (6.30) que, a inequação (6.35) pode ser forçada pela escolha de ν tal que:

$$\nu \geq \frac{q(\tilde{\mathbf{v}} + \tilde{\mathbf{w}})}{(1 - \rho)vpred} \quad (6.36)$$

Como mostrado em [59], se $m(\tilde{\mathbf{v}}) = 0$, então $\tilde{\mathbf{v}} = 0$, que implica em $q(\tilde{\mathbf{v}} + \tilde{\mathbf{w}}) \leq 0$, e então (6.35) é satisfeito para qualquer valor de ν . Neste caso ν pode ser definido como seu valor da iteração anterior, que será chamado aqui de ν^- . Assim a atualização de ν é feita como mostrado a seguir:

$$\text{se } m(\tilde{\mathbf{v}}) = 0 \text{ então } \nu = \nu^- \quad (6.37)$$

Senão

$$\nu = \max\left\{\nu^-, \frac{m(\tilde{\mathbf{V}})}{(1-\rho)vpred}\right\}$$

Este procedimento é aplicado enquanto o parâmetro de barreira μ é fixo. Desta forma, para um valor fixo do parâmetro de barreira o parâmetro de penalidade ν aumenta monotonicamente à medida que as iterações avançam, que é uma propriedade importante para a análise da convergência

global do algoritmo. Se o valor do parâmetro de barreira for trocado exatamente no início da iteração corrente, o valor ν^- que será utilizado no procedimento de escolha do parâmetro ν volta a ser igual ao seu valor próprio inicial. Sendo que a função de mérito já foi especificada, esta será utilizada para determinar se o passo d^k da iteração corrente será aceito ou não. Nos método de região de garantia é necessário calcular a *Redução Atual* na função de mérito, que é dada por:

$$ared(\mathbf{d}) = \phi(\mathbf{x}, \mathbf{s}, \nu) - \phi(\mathbf{x} + \mathbf{d}_x, \mathbf{s} + \mathbf{d}_s, \nu) \quad (6.38)$$

Os valores de \mathbf{d} somente são aceitos se,

$$\gamma \equiv \frac{ared(\mathbf{d})}{pred(\mathbf{d})} \geq \eta \quad (6.39)$$

Se o passo é aceito, então o raio da região de garantia é aumentado da seguinte forma:

$$\Delta^+ = \begin{cases} \max\{7\|\mathbf{d}\|, \Delta\} & \text{se } \gamma \geq 0,9 \\ \max\{2\|\mathbf{d}\|, \Delta\} & \text{se } 0,3 \leq \gamma < 0,9 \\ \Delta & \text{se } \eta \leq \gamma < 0,3 \end{cases} \quad (6.40)$$

Quando o passo é rejeitado, o novo raio da região de garantia é no máximo a metade do valor do último raio, mas não é menor do que um décimo do valor do mesmo. Quando o parâmetro de barreira μ é reduzido Δ é ajustado pela regra $\Delta \leftarrow \max\{5\Delta, 1\}$.

6.8 Algoritmo para o Método de Região de Garantia

Esta seção irá descrever somente o passo 17 da seção 6.5 e os procedimentos iniciais necessários para os cálculos do problema. Incluir no passo 1 do algoritmo apresentado na seção 6.5, a escolha dos parâmetros constantes do problema tal que, $\tau \in (0, 1)$, $\theta \in (0, 1)$ e $\zeta \in (0, 1)$.

1. Calcular o passo normal $\mathbf{v}^k = (\mathbf{v}_x^k, \mathbf{v}_s^k)$. (como mostra [33])
2. Calcular os multiplicadores de Lagrange dados pela Eq. (6.26);
3. Calcular a matriz W com os novos valores de λ^k e π^k , e Σ_k , usando (6.29);
4. Calcular o passo tangencial ω_k utilizando o método dado em [33];
5. Calcular o passo total $\mathbf{d}^k = \mathbf{v}^k + \omega^k$;
6. Atualizar o parâmetro de penalidade ν , como dado por (6.37);
7. Calcular $pred^k(\mathbf{d}_k)$ utilizando (6.30) e $ared^k(\mathbf{d}^k)$ por (6.38);
8. Verificar se $ared^k(\mathbf{d}^k) \geq \eta pred^k(\mathbf{d}_k)$, e ir ao passo 9, senão ir ao passo 10

9. Atualizar as variáveis primais $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}_x^k$ e $\mathbf{s}^{k+1} = \mathbf{s}^k + \mathbf{d}_s^k$ e o raio da região de garantia Δ^{k+1} .
10. Fazer $\mathbf{x}^{k+1} = \mathbf{x}^k$ e $\mathbf{s}^{k+1} = \mathbf{s}^k$ e atualizar o raio da região de garantia tal que $\Delta^{k+1} \in [0.1\Delta^k, 0.5\Delta^k]$.

Os valores dos parâmetro utilizados nas simulações foram, $\tau = 0.995$, $\theta = 0.2$ e $\zeta = 0.8$.

6.9 Testes e Resultados

O problema mostrado, foi testado utilizando vários sistemas teste. Serão mostrados aqui os testes feitos com os sistemas teste IEEE 14, 30, 118 e 300 barras, o sistema teste England 39 barras e um sistema real, o sistema peruano de 460 barras. O ponto inicial para as componentes (complexas) real e imaginária da magnitude de tensão nas barras são 1.0 pu e 0.0 pu, respectivamente e para os taps dos transformadores com taps variáveis é 1.0 pu. Estes pontos iniciais foram utilizados por todos os problemas de PNL solucionados aqui. Como critério de parada, foi imposto um limite máximo de 50 iterações ao contador de iterações, para o problema geral e 50 iterações para o subproblema de barreira e uma tolerância mínima de 10^{-4} (erro do problema de PNL escalado [49]). Os resultados numéricos foram obtidos utilizando o mesmo computador apresentado no Capítulo 4.

A Tabela 6.1 mostra um resumo dos principais dados do problema. Onde nx é o número de todas as variáveis do problema e $nxlu$ é o número de variáveis canalizadas do problema, ng é o número de restrições de igualdade, nh é o número de restrições de desigualdade. Também é mostrado o número de iterações do problema geral (Geral), do subproblema de barreira logaritmica (SBL) e da técnica de região de garantia (TRG). Como a técnica de região de garantia faz parte do subproblema de barreira logaritmica, o número de iterações de TRG tem o intuito de mostrar quantos passos de região de garantia foram dados dentro do subproblema de barreira logaritmica.

Tab. 6.1: Resumo das Características de Otimização

Data	nx	$nxlu$	ng	nh	número de iterações		
					Geral	SBL	TRG
IEEE14	33	5	19	54	4	11	0
IEEE30	64	4	49	88	3	10	0
ENGL39	78	0	67	98	4	10	1
IEEE118	245	9	169	386	4	11	2
IEEE300	650	50	518	862	4	26	1
PERU460	1054	134	244	1338	4	18	2

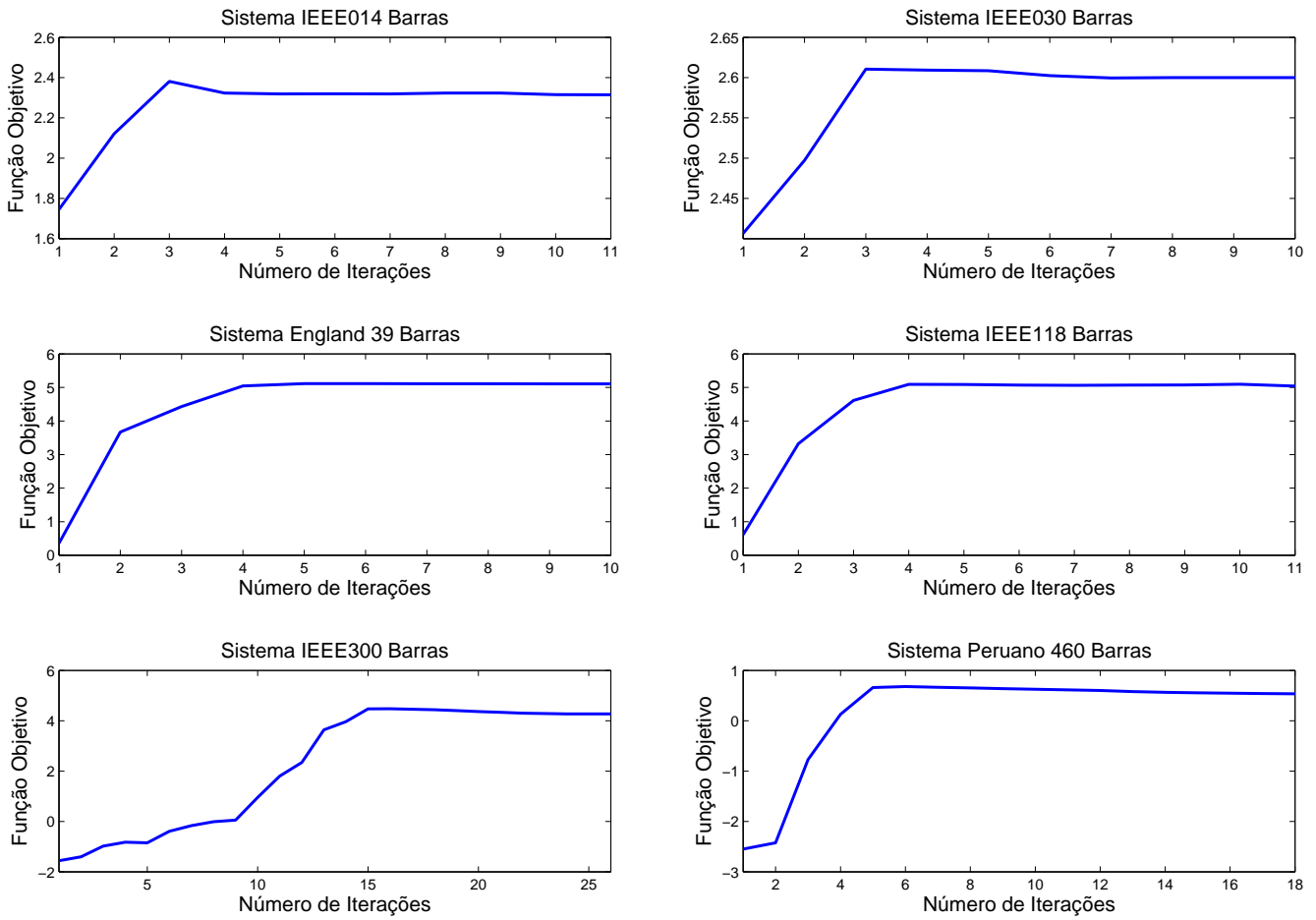


Fig. 6.1: Trajetória de Convergência da Função Objetivo

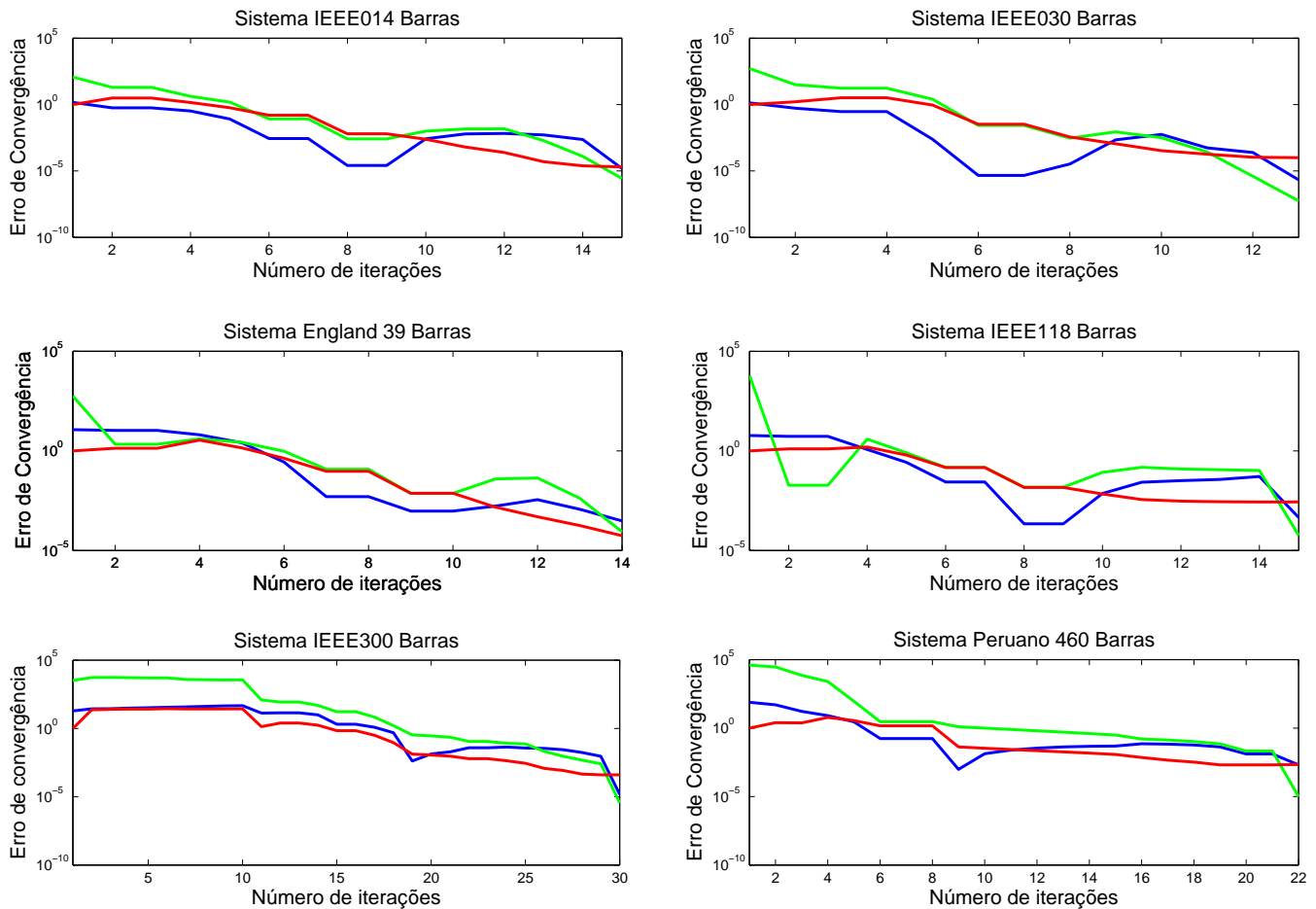


Fig. 6.2: Trajetória de Convergência

As Figuras 6.1 e 6.2 mostram a trajetória da função objetivo e da convergência de cada sistema testado, respectivamente. A Fig. 6.2 mostra a evolução dos critérios de convergências utilizados, como os erros primal e dual e o gap de complementaridade.

O algoritmo apresentado neste capítulo se mostrou eficiente para resolver o problema de FPOR e apesar de possuir uma estrutura mais complexa que um algoritmo de MPI primal-dual, não apresenta um custo computacional elevado além de assegurar uma convergência global do problema, isto também pode ser observado nos resultados do Capítulo 4. Este algoritmo pode ser facilmente implementado pois é uma extensão do algoritmo de MPI primal-dual.

Capítulo 7

Conclusão

Neste relatório técnico foi apresentado: no Capítulo 3 uma nova formulação para resolver o problema de fluxo de potência ótimo reativo utilizando equações de injeção de corrente; no Capítulo 4 uma comparação entre cinco diferentes softwares de programação não linear para resolver o problema de fluxo de potência ótimo reativo utilizando equações de potência; no Capítulo 5 uma comparação do esforço computacional entre a formulação do fluxo de potência ótimo reativo utilizando equações de injeção de corrente e a formulação do fluxo de potência ótimo reativo utilizando equações de potência; e no Capítulo 6 um algoritmo de método de pontos interiores com técnicas de região de garantia para resolver a formulação do fluxo de potência ótimo reativo utilizando equações de potência.

A nova formulação do FPOR-IC apresentou uma matriz Jacobiana das restrições de igualdade cuja estrutura está composta por vários elementos constantes, em consequência disso, apresentou uma matriz Hessiana altamente esparsa. Estas características ajudam a diminuir o esforço computacional para resolver o problema de FPOR-IC, especialmente quando é usado um método de ponto interiores. Cinco sistemas teste e três sistemas reais foram usados para comparar o esforço computacional entre o FPOR-IC e o FPOR-EP. O programa de otimização IPOPT foi utilizado para fazer as comparações. Um conjunto total de 4401 casos (caso base e contingências factíveis) foram utilizados para avaliar as duas formulações. Os resultados mostraram que o tempo de processamento do FPOR-IC é de aproximadamente 41% menor que do FPOR-PE.

Cinco diferentes softwares de programação não linear foram comparados, no intuito de mostrar quais dos softwares apresentam melhor desempenho para solucionar o problema de FPOR-EP. Para avaliar a eficiência de cada técnica de otimização, perfis de desempenho foram utilizados. Comparações de tempo computacional total, tempo computacional de avaliação das funções, número de iterações e casos de não convergência foram apresentados. Os programas de otimização IPOPT, KNITRO e o LOQO apresentam um bom desempenho computacional para resolver o problema de FPOR-EP, também pode-se observar que o LOQO apresentou um melhor desempenho computacional em relação ao tempo total de processamento para os sistemas de pequeno porte. Entretanto, para os sistemas de grande porte o IPOPT e o KNITRO foram os que apresentaram melhores resultados.

O MPI-TRG foi testado para resolver o problema de FPOR-PE para vários sistemas teste. Este algoritmo inclui a) um subproblema de barreira logaritmica; b) uma busca linear para encontrar um passo ótimo; e c) uma escolha entre o método de pontos interiores e a técnica de região de garantia dependendo da convexidade do problema em cada iteração. Seis sistemas teste foram usados para avaliar o desempenho computacional do algoritmo. Este algoritmo tem como objetivo ser rápido e assegurar uma convergência global do problema.

Referências Bibliográficas

- [1] J. Carpentier, “Contribution to the Economic Dispatch Problem” (in French), *Bulletin Society Française Electriciens*, vol 3, no. 8, pp. 431-447, Aug. 1962.
- [2] M.E. El-Hawary, “Optimal Power Flow: Solution Techiques, Requirements and Challenges”, *In IEEE Tutorial Course*, New York, 1996.
- [3] H. W. Dommel and W. F. Tinney, “Optimal Power Flow Solutions”, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, no. 10, pp. 1866-1876, Oct. 1968.
- [4] T. C. Giras and S. N. Talukdar, “Quasi-Newton Method for Optimal Power Flows”, *International Journal of Electrical Power and Energy Systems*, vol. 3, no. 2, Apr. 1981.
- [5] R. C. Burchett, H. H. Happ and D. R. Vierath, “Quadratically Convergent Optimal Power Flow”, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 11, pp. 3267-3275, Nov. 1984.
- [6] D. I. Sun, B. Ashley, B. Brewer, A. Hughes and W. F. Tinney, “Optimal Power Flow by Newton Approach”, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 10, pp. 2864-2880, Oct. 1984.
- [7] B. Stott and J. L. Marinho, “Linear Programming for Power-System Network Security Applications”, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, no. 3, pp. 837-848, May/June 1979.
- [8] O. Alsac, J. Bright, M. Prais and B. Stott, “Further Developments in LP-Based Optimal Power Flow”, *IEEE Transactions on Power Systems*, vol. 5, no. 3, pp. 697-711, Aug. 1990.
- [9] V. H. Quintana, G. L. Torres and J. Medina-Palomo, “Interior-point methods and their applications to power systems: A classification of publications and software codes,” *IEEE Trans. on Power Syst.*, vol. 15, no. 1, pp. 170-176, Feb. 2000.
- [10] N. Karmarkar, “A New Polynomial Time Algorithm for Linear Programming”, *Combinatorica*, vol 4, pp. 373-395.

- [11] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Trans. on Power Syst.*, vol. 9, no. 1, pp. 136-146, Feb. 1994.
- [12] Y. C. Wu, A. S. Debs and R. E. Marsten, "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows," *IEEE Trans. on Power Syst.*, vol. 9, no. 2, pp. 876-883, May 1994.
- [13] G. L. Torres and V. H. Quintana, "An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates," *IEEE Trans. on Power Syst.*, vol. 13, no. 4, pp. 1211-1218, Nov. 1998.
- [14] H. Wei, H. Sasaki, J. Kubokawa and R. Yohoyama, "An interior point methods for power systems weighted nonlinear L1 norm static state estimation", *IEEE Transactions on Power Systems*, vol. 13, no. 2, pp. 617-623, May 1998.
- [15] G. D. Irisarri, X. Wang, J. Tong and S. Mokhtari, "Maximum loadability of power systems using interior point method nonlinear optimization", *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 162-172, Feb. 1997.
- [16] S. Granville, J. C. O. Mello and A. C. G. Melo, "Application of interior point methods to power flow unsolvability," *IEEE Transactions on Power Systems*, vol. 11, no. 2, pp. 1096-1103, May 1996.
- [17] M. J. Rider, C. A. Castro, M. F. Bedriñana and A. V. Garcia, "Towards a fast and robust interior point method for power system applications," *Proc. IEE Gen., Transm., Distrib.*, vol. 151, no. 5, pp. 575-581, Sep. 2004.
- [18] X. Wang, G. C. Ejebe, J. Tong and J. G. Waight, "Preventive/corrective control for voltage stability using direct interior point method", *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 878-883, Aug. 1998.
- [19] J. Medina, V. H. Quintana, A. J. Conejo and F. P. Thoden, "A comparison of interior-point codes for medium-term hydro-thermal coordination," *IEEE Trans. on Power Syst.*, vol. 13, no. 3, pp. 836-843, Aug. 1998.
- [20] X. Yan and V. H. Quintana, "An efficient predictor - corrector interior point algorithm for security-constrained economic dispatch", *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 803-810, May 1997.
- [21] R. A. Jabr, A. H. Coonick and B. J. Cory, "A primal-dual interior point method for optimal power flow dispatching", *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 654 - 662, Aug. 2002.

- [22] M. J. Rider, A. V. Garcia and R. Romero, "Transmission System Expansion Planning by a Branch-and-Bound Algorithm", *IEE Proceedings of Generation, Transmission and Distribution*, vol. 2, no. 1, pp. 90-99, Jan. 2008.
- [23] A. R. L. Oliveira, S. Soares and L. Nepomuceno, "Optimal active power dispatch combining network flow and interior point approaches", *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1235 - 1240, Nov. 2003.
- [24] D. Youjie, J. D. McCalley and V. Vittal, "Simplification, expansion and enhancement of direct interior point algorithm for power system maximum loadability", *IEEE Transactions on Power Systems*, vol. 15, no. 3, pp. 1014 - 1021, Aug. 2000.
- [25] A. A. Sousa and G. L. Torres, "Globally convergent optimal power flow by trust-region interior-point methods," in *Proc. of the International Conference 2007 IEEE Switzerland PowerTech*, pp. 01-06, Lausanne, Switzerland, Jul. 01-05, 2007.
- [26] W. Hongye and R. J. Thomas, "Towards Reliable Computation of Large-Scale Market-Based Optimal Power Flow", in *40th Annual Hawaii International Conference on System Sciences (HICSS 2007)*, pp. 1 - 10, Jan. 2007.
- [27] M. Yongfei and W. Xianmin, "Application of sequential quadratic programming algorithm based on region search method in reactive power optimization," in *Proc. of the 3rd International Conference on Electrical and Electronics Engineering*, pp. 01-05, Veracruz, Mexico, Sep. 06-08, 2006.
- [28] Z. Renjun, Z. Yanping and L. Jianhua, "Reactive power optimization algorithm based on trust-region global SQP," in *Proc. of the 2006 PES General Meeting*, pp. 01-05, Montreal, Canada, Jun. 18-22, 2006.
- [29] R. Zhou, Y. Zhang and H. Yang, "A trust-region algorithm based on global SQP for reactive power optimization," in *Proc. of the 2nd International Conference on Electrical and Electronics Engineering*, pp. 292-295, Mexico City, Mexico, Sep. 07-09, 2005.
- [30] S. Pajic and K. A. Clements, "Power system state estimation via globally convergent methods", *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1683 - 1689, Nov. 2005.
- [31] M. A. Gonzalez and J. R. Vega, "An economic dispatch algorithm for cogeneration systems", *IEEE 2004 Power Engineering Society General Meeting*, pp. 989 - 994, 6-10 Jun. 2004.
- [32] Z. Daoyuan, P. B. Luh and Z. Yuanhui, "A bundle method for hidrothermal scheduling", *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1355 - 1361, Nov. 1999.
- [33] R. Byrd, M. E. Hribar and J. Nocedal, "An interior point method for large scale nonlinear programming," *SIAM J. Optim.*, vol. 9, no. 4, pp. 877-900, 1999.

- [34] R. Byrd, G. Liu and J. Nocedal, "On the local behavior of an interior point method for Non-linear Programming", in *Numerical Analysis 1997*, pp. 37 - 56, 1998, Addison Wesley Longman.
- [35] R. H. Byrd, N. I. M. Gould, J. Nocedal and R. A. Waltz, "An algorithm for nonlinear optimization using linear programming and equality constrained subproblems", *Mathematical Programming, Series B*, vol. 100, no. 1, pp. 27-48, 2004.
- [36] R. Byrd, N. Gould, J. Nocedal and R. Waltz, "On the convergente of successive linear programming algorithms". *Technical Report OTC 2002*, 2002.
- [37] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, 1968.
- [38] E. D. Castronuovo, "Aplicação de métodos de pontos interiores no fluxo de potência ótimo não-linear com utilização de processamento de alto desempenho", *Departamento de Engenharia Elétrica - Centro Tecnológico, UFSC*, Tese de Doutorado, Maio, 2001.
- [39] Y. Wu and A. S. Debs, "Initialisation, decoupling, hot start, and warm start in direct nonlinear interior point algorithm for optimal power flows", *IEE Proceedings of Generation, Transmission and Distribution*, vol. 148, no. 1, pp. 67-75, January, 2001.
- [40] J. L. M. Ramos and A. G. Exposito and V. H. Quintana, "Transmission power loss reduction by interior-point methods: implementation issues and practical experience", *IEE Proceedings of Generation, Transmission and Distribution*, vol. 152, no. 1, pp. 90-98, January, 2005.
- [41] Y. Dai and J. D. McCalley and V. Vittal, "Simplification, expansion and enhancement of direct interior point algorithm for power system maximum loadability", *IEEE Transactions on Power Systems*, vol. 15, no. 3, pp. 1014-1021, August, 2000.
- [42] S. S. Sharif, J. H. Taylor, E. F. Hill, B. Scott and D. Daley, "Real-time implementation of optimal reactive power flow," *IEEE Power Engineering Review*, vol. 20, no. 8, pp. 47-51, Aug. 2000.
- [43] W. Yan, J. Yu, D. C. Yu and K. Bhattarai, "A new optimal reactive power flow model in rectangular form and its solution by predictor corrector primal dual interior point method," *IEEE Trans. on Power Syst.*, vol. 21, no. 1, pp. 61-67, Feb. 2006.
- [44] W. Rosehart, C. Roman and A. Schellenberg, "Optimal power flow with complementarity constraints," *IEEE Trans. on Power Syst.*, vol. 20, no. 2, pp. 813-822, May 2005.
- [45] T. Orfanogianni and R. Bacher, "Increased OPF code development efficiency by integration of general purpose optimization and derivative omputation tools," *IEEE Trans. on Power Syst.*, vol. 15, no. 3, pp. 987-993, Aug. 2000.

- [46] M. T. Vasconcelos, R. J. Vanderbei and A. Santos Junior, “New Optimal Power Flow Teaching Using LOQO Interior Point solver and AMPL Modeling,” in *Proc. of the International Conference on Engineering and Computer Education (ICECE/IEEE)*, Rio de Janeiro, 1999.
- [47] V. M. da Costa, N. Martins and J. L. R. Pereira “Developments in the Newton Raphson Power Flow Formulation Based on Current Injections”. *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp 1320 - 1326, November 1999.
- [48] V. M. da Costa, M. Lavorato and M. R. Guedes. “Developments in the Analysis of Unbalanced Three-Phase Power Flow Solutions”. *Electrical Power and Energy Systems*, vol. 29, no. 2, pp. 175-182, February. 2007.
- [49] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Program.*, vol. 106, no. 1, pp. 25-57, 2006.
- [50] R. H. Byrd, J. Nocedal and R. A. Waltz, “KNITRO: An integrated package for nonlinear optimization,” *Large-Scale Nonlinear Optimization*, pp. 35-59, New York, Springer Verlag, 2006.
- [51] R. J. Vanderbei, “LOQO: An interior point code for quadratic programming,” *Optimization Methods and Software*, vol. 12, pp. 451-484, 1999.
- [52] B. A. Murtagh and M. A. Saunders, “MINOS 5.5 User’s Guide, Report SOL 83-20R,” *Systems Optimization Laboratory, Stanford University*, revised July 1998.
- [53] P. E. Gill, W. Murray and M. A. Saunders, “SNOPT: An SQP algorithm for large-scale constrained optimization,” *SIAM J. Optim.*, vol. 12, pp. 979-1006, 2002.
- [54] ‘NEOS server for optimization’, disponível em: <http://neos.mcs.anl.gov/neos/>, ultimo acesso October 2008.
- [55] R. Fourer, D. M. Gay and B. W. Kernighan, *AMPL: A modeling language for mathematical programming*. CA: Brooks/Cole-Thomson Learning, Pacific Grove, 2nd Ed., 2003.
- [56] E. D. Dolan and J. J. Moré, “Benchmarking optimization software with performance profiles,” *Mathematical Programming, Series A*, vol. 91, no. 2, pp. 201-213, 2002.
- [57] A. Wächter and L. T. Biegler, “Line search filter methods for nonlinear programming: Motivation and global convergence,” *SIAM J. Optim.*, vol. 16, no. 1, pp. 1-31, 2005.
- [58] A. Wächter and L. T. Biegler, “Line search filter methods for nonlinear programming: Local convergence,” *SIAM J. Optim.*, vol. 16, no. 1, pp. 32-48, 2005.

-
- [59] R. Byrd, J. C. Gilbert, and J. Nocedal, “A trust region method based on interior point techniques for nonlinear programming,” *Mathematical Programming*, vol. 89, no. 1, pp. 149-185, 2000.
- [60] ‘Same AMPL examples’, available at: <http://www.orfe.princeton.edu/~rvdb/ampl/nlmodels/>, accessed October 2008.
- [61] R. A. Waltz, J. L. Morales, J. Nocedal, D. Orban, “An interior algorithm for nonlinear optimization that combines line search and trust region steps”, *Mathematical Programming, Series A and B*, vol. 107, no. 3, pp. 391-408, July, 2006.

Apêndice A

Artigo submetido

Apêndice B

Avaliações dos Professores Brasileiro e Estrangeiro

Apêndice C

Bilhetes de Passagem Aérea

Apêndice D

Confirmação de endereço e Telefones de Contato no Brasil

D.1 Endereço Residencial

Rua Hermantino Coelho, 255, apto. 31, bloco 2
Bairro: Mansões Santo Antônio
CEP: 13087-500, Campinas, SP
Telefones: (19) 3326 3804 (19) 8112 7445

D.2 Endereço da Universidade

UNICAMP/FEEC/DSEE
Av. Albert Einstein, 400
CEP: 13083-852, Campinas, SP
Telefone: (19) 3521 0260