

CARLOS EDUARDO FERREIRA SOARES

**ANÁLISE DE FERRAMENTA DE FLUXO DE POTÊNCIA
ÓTIMO NA OTIMIZAÇÃO DE SISTEMA REAL DE
SUBTRANSMISSÃO**

Recife - Pernambuco - Brasil

Abril de 2010



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ANÁLISE DE FERRAMENTA DE FLUXO DE POTÊNCIA
ÓTIMO NA OTIMIZAÇÃO DE SISTEMA REAL DE
SUBTRANSMISSÃO**

por

CARLOS EDUARDO FERREIRA SOARES

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da
Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de
Mestre em Engenharia Elétrica

ORIENTADOR: PROF. GERALDO LEITE TORRES, PhD

Recife, abril de 2010.

© Carlos Eduardo Ferreira Soares, 2010

S676a

Soares, Carlos Eduardo Ferreira.

Análise de Ferramenta de Fluxo de Potência Ótimo na Otimização de Sistema Real de Subtransmissão / Carlos Eduardo Ferreira Soares. – Recife: O Autor, 2010. xix, 129 folhas, il., gráfs., tabs.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2010.

Orientador: Prof. Geraldo Leite Torres

Inclui referências bibliográficas e anexos.

1. Engenharia Elétrica. 2. Fluxo de Potência Ótimo. 3. Otimização da Operação. 4. Pontos Interiores – Método.
I. Título.

621.3 CDD (22.ed.)

UFPE

BCTG/2010-128



Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE
DISSERTAÇÃO DO MESTRADO ACADÊMICO DE

CARLOS EDUARDO FERREIRA SOARES

TÍTULO

**“ANÁLISE DE FERRAMENTA DE FLUXO DE POTÊNCIA ÓTIMO NA
OTIMIZAÇÃO DE SISTEMA REAL DE SUBTRANSMISSÃO”**

A comissão examinadora composta pelos professores: GERALDO LEITE TORRES, DEESP/UFPE, MANOEL AFONSO DE CARVALHO JÚNIOR, DEESP/UFPE, e ANTÔNIO CARLOS ZAMBRONI DE SOUZA, DEE/UNIFEI sob a presidência do primeiro, consideram o candidato **CARLOS EDUARDO FERREIRA SOARES APROVADO.**

Recife, 29 de abril de 2010.

RAFAEL DUEIRE LINS
Coordenador do PPGE

GERALDO LEITE TORRES
Orientador e Membro Titular Externo

ANTÔNIO CARLOS ZAMBRONI DE SOUZA
Membro Titular Externo

MANOEL AFONSO DE CARVALHO JÚNIOR
Membro Titular Externo

Aos meus pais, Roberto Carlos e Edirany,
aos meus irmãos, Carlos Roberto e Mariana,
a minha esposa, Roberta,
e ao meu filho, Leonardo,
pela atenção, dedicação, apoio e carinho,
DEDICO.

Agradecimentos

Ao meu orientador, Geraldo Leite Torres, pela orientação ao longo do desenvolvimento deste trabalho.

Aos colegas do LOASP, Vicente e Andréa, pela atenção e apoio em todos os momentos de dificuldade.

Aos colegas Francisco Nilton e Manuelle Cavalcanti pelo incentivo ao longo de todo o trabalho.

Aos amigos e familiares pelo apoio e incentivo.

Agradecimentos Especiais

Ao meu pai, Roberto Carlos, que com uma conduta impecável, sempre foi uma grande referência pessoal e profissional.

A minha mãe, Edirany, pela exaustiva dedicação durante toda minha formação acadêmica. Seu acompanhamento e incentivo nos primeiros anos de estudo foram fundamentais nesta caminhada.

Aos meus irmãos, Carlos Roberto e Mariana, pelo amor e carinho durante toda vida.

A minha esposa, Roberta, pelo amor e incentivo sempre.

Ao meu filho, Leonardo, pelo novo sentido que sua chegada trouxe a minha vida.

Aos avós, Maria Zilda, Espedito (in memoriam) e Lourdes, pela dedicação e o amor durante nosso convívio.

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

ANÁLISE DE FERRAMENTA DE FLUXO DE POTÊNCIA ÓTIMO NA OTIMIZAÇÃO DE SISTEMA REAL DE SUBTRANSMISSÃO

Carlos Eduardo Ferreira Soares

Abril de 2010

Orientador: Prof. Geraldo Leite Torres, PhD

Área de Concentração: Processamento da Energia

Palavras-chave: Fluxo de Potência Ótimo, Otimização da Operação, Métodos de Pontos-Interiores.

Número de Páginas: 129

Com o aumento da dimensão e da complexidade dos sistemas elétricos de potência, o nível de automação da operação vem crescendo continuamente, de forma que os centros de operação das empresas de energia elétrica necessitam cada vez mais de ferramentas computacionais que possam analisar a segurança do sistema e otimizar um objetivo da operação em tempo-real. Nesse contexto, as ferramentas de Fluxo de Potência Ótimo (FPO) exercem um papel fundamental. O objetivo de um programa de FPO é calcular ajustes ótimos para os controles do sistema visando a sua operação num nível de segurança desejado, enquanto otimiza uma função objetivo tal como a minimização de perdas elétricas. Os ajustes ótimos calculados podem conduzir a operação do sistema de um nível de segurança para outro ou restaurar a otimalidade de um nível de segurança já alcançado. No entanto, existem diversas dificuldades no uso prático de programas de FPO. A principal delas é que o problema de FPO no mundo real é matematicamente e computacionalmente muito diferente da formulação clássica. Essa Dissertação apresenta uma análise dos principais requisitos para o uso prático de ferramentas de FPO, bem como discute desenvolvimentos recentes para o tratamento de alguns dos requisitos críticos. Mais especificamente, é feita uma análise crítica da aplicação de um programa de FPO recentemente desenvolvido para otimização da operação de um sistema real de subtransmissão. São apresentados e discutidos resultados numéricos da aplicação da ferramenta de FPO na otimização de sistema real de subtransmissão.

Abstract of Dissertation presented to UFPE as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

ANALYSIS OF OPTIMAL POWER FLOW TOOL IN THE OPTIMIZATION OF AN ACTUAL SUBTRANSMISSION SYSTEM

Carlos Eduardo Ferreira Soares

April, 2010

Supervisor: Prof. Geraldo Leite Torres, PhD

Area of Concentration: Energy processing

Keywords: Optimal Power Flow, Operation Optimization, Interior-Point Methods.

Number of Pages: 129

With the increase in dimension and complexity of electrical power systems, the level of power systems operation automation is continuously increasing, so that Energy Management Systems need computational tools capable to assess system security and optimize an operation objective in real time. In this context, Optimal Power Flow (OPF) tools play a fundamental role. The objective of an OPF program is to compute optimal adjustments to the system controls aiming at its operation with a desired security level, while optimizing an objective function such as active losses minimization. The optimal control adjustments computed can drive system operation from a security level to another, or restore the optimality of an already secure operation. However, there exists several difficulties in the practical use of OPF tools. The main one is that the OPF problems in the real world are mathematically e computationally very different from the classical formulation. This Dissertation presents an analysis of the main requirements for practical usage of OPF tools, as well as discusses recent developments for the treatment of critical requirements. More specifically, it presents a critical analysis of the application of a recently developed OPF program to the optimization a an actual sub-transmission system. Numerical results are presented and discussed.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Algoritmos	xv
Lista de Símbolos e Abreviaturas	xvi
1. Introdução	1
1.1. Objetivos da Pesquisa	4
1.2. Composição da Dissertação	5
2. Modelos de Fluxo de Potência Ótimo	7
2.1. Forma Geral do Problema	8
2.2. Equações Básicas de Fluxo de Potência	11

2.2.1.	Fluxos de Correntes e Potências nos Circuitos	11
2.2.2.	Injeções de Potências nos Nós	15
2.2.3.	Equações de Balanço de Potência	17
2.3.	Modelos de Fluxo de Potência Ótimo	18
2.3.1.	Minimização de Perdas Ativas na Transmissão	19
2.3.2.	Minimização do Corte de Carga	21
2.3.3.	Maximização do Carregamento do Sistema	23
2.3.4.	Maximização da Capacidade de Transferência Simultânea	25
2.4.	As Técnicas Usuais de Solução	25
3.	Métodos de Pontos-Interiores	29
3.1.	O Método Primal-Dual Simples	30
3.1.1.	Cálculo das Direções de Busca	32
3.1.2.	Atualização das Variáveis	33
3.1.3.	Redução do Parâmetro de Barreira	34
3.1.4.	Testes de Convergência	35
3.2.	O Método Preditor-Corretor	36
3.2.1.	O Passo Preditor	38
3.2.2.	O Passo Corretor	39
3.3.	O Método Preditor-Corretor Múltiplo	40
3.4.	O Método de Múltiplas Correções de Centralidade	41
3.4.1.	As Correções de Centralidade	42
3.4.2.	Quantas Correções Realizar?	44
4.	Estudo dos Requisitos para Utilização Prática de Programas de FPO	45
4.1.	Otimização por Áreas	47

4.2. Restrições de Contingências	48
4.3. Objetivos Operacionais	49
4.4. Supressão de Ajustes Ineficazes	51
4.5. Inviabilidades na Solução do FPO	53
4.6. Discretização de Variáveis	56
4.7. Interface Gráfica com o Usuário	57
4.8. Análise de Sensibilidade	59
5. Aplicação de Programa de FPO em Sistema Real de Subtransmissão	62
5.1. Características do Sistema de Subtransmissão	63
5.2. Os Sistemas Testes	64
5.3. Os Dados Necessários	66
5.4. Resumo de Análises de Fluxo de Potência	69
5.5. Otimização dos Regionais	71
5.5.1. Solução com Parâmetros Padrões	71
5.5.2. Influência do Ponto Inicial	72
5.5.3. Influência da Regra do Comprimento de Passo	73
5.5.4. Influência do Parâmetro de Barreira Inicial	75
5.6. Observações Finais	76
6. Conclusões	77
6.1. Sugestões de Trabalhos Futuros	80
Referências Bibliográficas	82
A. Implementação dos Algoritmos de Pontos-Interiores	88
A.1. Inicialização das Variáveis	88

A.2. Montagem da Matriz $\nabla g(x)$	89
A.3. Montagem das Matrizes Hessianas	90
B. Estrutura do Programa OOTrans	93
B.1. Utilizando o Programa OOTrans	94
B.1.1. Habilitação do Usuário	95
B.1.2. Informando os Arquivos de Dados	96
B.1.3. Configurando a Solução de Fluxo de Potência	99
B.1.4. Configurando a Solução de Minimização de Perdas	100
B.1.5. Ajustando o Modelo de Otimização	102
B.1.6. Selecionando Relatórios para Impressão	106
B.1.7. Executando a Otimização	107
B.1.8. Visualizando Resultados	108
B.1.9. Analisando a Solução	109
B.2. As Rotinas Computacionais	110

Lista de Figuras

2.1. Representação geral de linhas de transmissão.	12
2.2. Representação geral de transformadores.	13
4.1. Diagrama unifilar com animação para exibição de resultados no programa PowerWorld.	60
5.1. Localização dos regionais do sistema elétrico CELPE.	63
B.1. Janela inicial do programa OOTrans	94
B.2. Habilitação do usuário.	95
B.3. Barra de ferramentas após habilitação do usuário.	95
B.4. Primeira pasta de configuração do programa OOTrans	96
B.5. Janela de seleção de arquivos da rede.	97
B.6. Seleção do formato ANAREDE para os dados da rede.	97
B.7. Seleção do arquivo de dados Angelim.pwf	98

B.8. Janela de trabalho Fluxo de Potência	100
B.9. Janela de trabalho Otimização	101
B.10. Painel para escolha do ponto inicial.	103
B.11. Exemplo de ajuste nas variáveis de controle tape.	105
B.12. Exemplo da exclusão da variável de controle tape do circuito 40.	106
B.13. Janela de trabalho para seleção de relatórios.	107
B.14. Exemplo de pesquisa da barra 1 do regional Angelim.	110

Lista de Tabelas

5.1. Número de barras, circuitos e transformadores do sistema CELPE.	64
5.2. Quantidades e modulações das susceptâncias <i>shunts</i> e tapes de LTC's.	65
5.3. Montante de perdas por regional na solução não-otimizada.	66
5.4. Número de variáveis n , número de restrições de igualdades m , número de limites simples sobre as variáveis p , e número de soluções discretas possíveis, por regional.	67
5.5. Valores padrões dos parâmetros dos algoritmos.	68
5.7. Quadro síntese da análise de redes do sistema CELPE (extraído do Relatório III do Projeto de P&D).	70
5.8. Desempenho dos algoritmos de otimização utilizando os parâmetros padrões.	72
5.9. Desempenho dos algoritmos com as variáveis x iniciadas pela regra do ponto médio (opção <i>Ponto Médio</i> no Popup Menu).	73
5.10. Desempenho dos algoritmos quando podem utilizar comprimentos de passos distintos: $\alpha_k^P \neq \alpha_k^D$	74

5.11. Desempenho dos algoritmos com $\mu_0 = 0.1$	75
5.12. Desempenho dos algoritmos com $\mu_0 = 1$	76

Lista de Algoritmos

3.1. Método primal-dual de pontos-interiores para resolver (2.1).	32
3.2. Método preditor-corretor de pontos interiores para resolver (2.1).	40

Lista de Símbolos e Abreviaturas

Abreviaturas

FPO	: Fluxo de Potência Ótimo
KKT	: condições de Karush-Kuhn-Tucker
LTC	: Load Tap Changer
MCC	: Múltiplas Correções de Centralidade
PDS	: Primal-Dual de Pontos-Interiores
PCS	: Primal-Dual Preditor-Corretor
PI	: Pontos Interiores
PL	: Programação Linear
PLS	: Programação Linear Sucessiva
PNL	: Programação Não-Linear
PQ	: Programação Quadrática
PQS	: Programação Quadrática Sucessiva

Símbolos

\mathcal{B}	: conjunto de pares de barras terminais dos ramos do sistema
B_{ij}	: elemento ij da matriz susceptância de barra, B
b_i	: susceptância do ramo conectando as barras i e j
b_{ij}^{sh}	: susceptância do carregamento <i>shunt</i> do ramo ij
ΔP	: balanço de potência ativa
ΔQ	: balanço de potência reativa
\mathcal{E}	: conjunto das barras de carga elegíveis ao controle de reativos
$f(x)$: função objetivo
\mathcal{F}	: conjunto das barras com fontes fixas de reativos <i>shunt</i>
F_{ij}	: fluxo de potência no ramo ij
$g(x)$: vetor das funções de restrições de igualdade
g_{ij}	: condutância do ramo conectando as barras i e j
\mathcal{G}	: conjunto de todas as barras de geração
G_{ij}	: elemento ij da matriz condutância G
H_k	: Hessiana da função de Lagrange
\vec{I}	: corrente complexa
$J_{PF}(V, \theta)$: Jacobiana do problema de fluxo de potência não otimizado
λ	: multiplicadores de Lagrange das restrições de igualdade
μ	: parâmetro de barreira
\mathcal{N}	: conjunto de todas as barras do sistema
$\tilde{\mathcal{N}}$: conjunto de todas as barras exceto a barra de folga
P_i^G	: potência ativa gerada na barra i
P_i^D	: potência ativa demandada na barra i
P_L	: perdas ativas
Q_i^G	: potência reativa gerada na barra i

Q_i^D	: potência reativa demandada na barra i
t_{ij}	: tape entre os ramos ij
\mathcal{T}	: conjunto das barras terminais dos transformadores com LTC
θ_i	: ângulo de fase da tensão complexa na barra i
V_i	: magnitude da tensão complexa na barra i
\vec{V}_i	: tensão complexa na barra i : $\vec{V}_i = V_i e^{j\theta_i}$
\vec{y}_{ij}	: admitância complexa do ramo conectando as barras i e j : $\vec{y}_{ij} = g_{ij} + jb_{ij}$

Capítulo 1

Introdução

Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável (...) para aprender a conhecer a influência libertadora da beleza do reino do espírito, para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer.

Albert Einstein

NA OPERAÇÃO diária de um sistema elétrico de potência, o controle freqüente da operação almejando a operação econômica e segura, é uma tarefa extremamente complexa, devido as grandes dimensões e a complexidade dos sistemas elétricos atuais, envolvendo um número elevado de controles. A decisão sobre uma ação de controle ótima pode ser tomada de forma eficiente, tanto em modo de planejamento quanto em tempo real, com o

auxílio de um programa computacional de Fluxo de Potência Ótimo (FPO), o qual é parte integrante de um Sistema de Gerenciamento de Energia.

Com a introdução de novas tecnologias de monitoramento e controle, e a crescente competitividade no mercado de energia elétrica, o número de sistemas elétricos com automação da operação vem crescendo continuamente, de forma que os centros de operação das empresas de energia elétrica necessitam cada vez mais de programas computacionais que possam analisar a segurança do sistema e otimizar um objetivo da operação em tempo-real [1, 2], tal como a minimização de perdas elétricas.

Algumas empresas de energia elétrica atualmente dispõem nos seus centros integrados de operação, de um sistema de monitoramento e controle da operação. Um sistema de monitoramento e controle pode fornecer para um determinado instante de tempo, o estado e a topologia da rede elétrica em operação naquele instante, informações que possibilitam a otimização da operação em tempo-real.

Os programas de segurança da operação existentes nos centros de operação podem ser executados de duas formas: em tempo-real (on-line) ou no modo estudo (off-line).

No modo tempo-real (on-line), o modelo estático do sistema elétrico sob observação é geralmente derivado da saída do estimador de estados, ou seja, do sistema de monitoramento e controle da rede. Se o monitoramento da segurança detecta violações de limites operacionais em tempo real, então cálculos de controles da segurança para implementação imediata são necessários. Portanto, programas para aplicações em tempo-real têm uma necessidade particular por velocidade de processamento e confiabilidade.

No modo estudo (off-line), o modelo estático do sistema elétrico geralmente representa uma condição de operação prevista, gerada automaticamente a partir de padrões históricos armazenados, de informações recentes de tendências, e de conhecimentos específicos ou hipóteses. O principal objetivo dos programas no modo estudo é planejar, para um curto período de tempo, a otimalidade e a segurança futura da operação do sistema de potência. Estados de operação anteriores que foram armazenados também podem ser estudados.

O objetivo de um programa de FPO utilizado em tempo-real é calcular os controles do sistema para a sua operação num nível de segurança desejado, enquanto otimiza uma função objetivo tal como a minimização de perdas elétricas na transmissão. O novo ajuste de controles pode conduzir a operação do sistema de um nível de segurança para outro, ou

pode restaurar a otimalidade de um nível de segurança já alcançado.

Os programas de FPO vêm sendo projetados para uso interativo com o operador do sistema, seja no modo tempo-real ou no modo estudo [1, 3]. No modo tempo-real, os controles calculados, uma vez aceitos, podem ser implementados manualmente ou, onde possível, automaticamente: gerações e intercâmbios são informados ao CAG (controle automático da geração), e os outros controles são tratados pelos mecanismos adequados do controle supervisão.

De forma ideal, o objetivo final de um processo de otimização da operação em tempo-real é ter os ajustes otimizados dos controles, que são calculados pelo programa de FPO, implementados de maneira totalmente automática, sem qualquer intervenção humana. No entanto, para atingir esse objetivo, é necessário que a empresa disponha de meios para acionamento automático de todos os dispositivos de controle da rede, e que haja um elevado grau de confiança na confiabilidade e na precisão dos ajustes de controles que são calculados pelo FPO.

Embora o FPO tenha sido um assunto de intensa pesquisa desde a década de 1960, a última palavra sobre modelagem e solução de problemas de FPO ainda não foi escrita [2]. Em especial, no que diz respeito à satisfação de diversos requisitos importantes para o uso de programas de FPO em tempo real, tal como o manuseio de um grande número de variáveis discretas, garantia de convergência quando o problema é viável (obter a solução sempre que ela existir), tratamento eficiente dos casos inviáveis, minimização do número de controles modificados, etc. Relativamente há pouca literatura sobre o tratamento de requisitos para o uso do FPO em tempo real [4–7]. As pesquisas até então concentraram-se na busca por novos algoritmos de solução do FPO, com foco principalmente na rapidez de processamento e na modelagem de novos problemas de FPO.

Uma deficiência crítica nas formulações clássicas de FPO e nos métodos de solução é que o número de ações de controle obtidas na solução é geralmente muito grande para ser executado na operação real de um sistema, mesmo num controle automático em malha fechada [4]. Uma vez que muitos problemas de FPO estão relacionados com mudanças nos controles com relação a valores referenciais iniciais, o número de ações de controle implementadas deverá ser limitado às práticas reais de operação, ou seja, modificar apenas os controles com efeito significativo. Surpreendentemente, a redução do número de ações de controle é um problema crítico que tem recebido pouca atenção na literatura.

Vários controles cujas ações são limitadas em número são ajustáveis apenas em passos discretos, e métodos para lidar com variáveis discretas tendem a diminuir o número de movimentos dos controles, mantendo os controles discretos de menor importância nos seus valores iniciais [4]. As ações de controle não podem ser ordenadas de maneira simples, porque a importância de uma ação não está necessariamente relacionada à sua magnitude. Adicionalmente, cada controle participa na minimização da função objetivo e no atendimento das restrições, e assim não há uma maneira de separar esses dois efeitos.

É afirmado em [4] que há pouca esperança de se desenvolver uma técnica rápida para solução do problema de minimização do número de controles, uma vez que trata-se de um problema inerentemente combinatorial, e o objetivo deve ser desenvolver um bom método de aproximação da solução ótima que seja rápido o suficiente para uso prático. A avaliação é de que é melhor ter soluções aproximadamente ótimas para problemas realistas do que soluções rigorosamente ótimas para problemas irreais. A metodologia recentemente proposta em [8] para redução do número de ações de controle segue esse raciocínio.

Mesmo que o objetivo primário da otimização de um sistema elétrico seja, por exemplo, minimizar perdas ativas na transmissão, no contexto da aplicação da ferramenta de FPO em tempo-real a necessidade de implementar outras funções objetivos surge naturalmente. Alguns exemplos de objetivos auxiliares são: (i) minimização do desvio do ponto de operação, (ii) minimização do número de controles ajustados, (iii) supressão de ajustes ineficazes, (iv) minimização de violação de restrições, etc.

Por questões de competição comercial, metodologias para o tratamento de requisitos no uso do FPO em tempo-real são incorporadas aos programas comerciais de FPO sem a divulgação das metodologias implementadas na literatura especializada, de forma que há pouca literatura abordando o tratamento de requisitos para uso do FPO em tempo-real.

1.1 Objetivos da Pesquisa

Como comentado inicialmente, há relativamente pouca literatura sobre o tratamento de requisitos para aplicação do FPO em tempo real. Assim, a presente pesquisa de Mestrado trata desse tema complexo e carente de estudos, sendo norteadada pelos seguintes objetivos principais:

- Realizar um estudo das formulações clássicas de FPO e dos algoritmos de otimização atualmente empregados na resolução de problemas de FPO;
- Realizar um estudo dos principais requisitos para aplicações práticas dos programas computacionais de FPO, em especial para a otimização da operação de sistemas elétricos em tempo real;
- A partir dos conhecimentos adquiridos no desenvolvimento das duas etapas anteriores, realizar uma análise detalhada da aplicação de um programa de FPO recentemente desenvolvido para otimização da operação de um sistema elétrico real de subtransmissão.

1.2 Composição da Dissertação

Esta Dissertação encontra-se organizada em 6 capítulos, descritos a seguir:

- Capítulo 1 Apresentam-se a motivação para a pesquisa, a análise de alguns trabalhos relacionados com o tema da Dissertação e os principais objetivos do trabalho proposto.
- Capítulo 2 São apresentadas a formulação geral dos problemas de FPO e algumas formulações clássicas, incluindo a de minimização de perdas ativas na transmissão que é considerada neste trabalho.
- Capítulo 3 Faz-se uma apresentação detalhada e tutorial dos principais algoritmos de pontos-interiores aplicados na solução de problemas de FPO.
- Capítulo 4 Faz-se uma análise dos requisitos básicos para aplicações práticas de programas de FPO.
- Capítulo 5 Apresentam-se os resultados numéricos obtidos com um programa de FPO na linguagem MATLAB aplicado num sistema real de subtransmissão.
- Capítulo 6 Apresentam-se as conclusões e as perspectivas de trabalhos futuros.
- Anexo A Traz detalhes da implementação dos algoritmos de pontos-interiores, em especial do cálculo eficiente de gradientes e matrizes Hessianas.
-

Anexo B Faz-se uma descrição do uso do programa de FPO utilizado, com uma breve descrição das rotinas computacionais do programa.

Capítulo 2

Modelos de Fluxo de Potência Ótimo

Os problemas significativos que enfrentamos não podem ser resolvidos no mesmo nível de pensamento em que estávamos quando os criamos.

Albert Einstein

NA OPERAÇÃO diária de um sistema elétrico de potência as demandas das cargas por potências ativa e reativa modificam-se constantemente e muitas vezes resultam em níveis de tensões que estão bem além dos limites toleráveis, provavelmente violando restrições de operação de equipamentos de consumidores e da própria empresa de energia elétrica. Para corrigir essas condições de operação inaceitáveis, os operadores do sistema são constantemente requisitados para controlarem a produção, a absorção e o fluxo de

potência em todos os níveis do sistema, através do ajuste de diversas variáveis de controle do sistema, tais como a geração de potência ativa e a tensão terminal dos geradores, o tap dos transformadores com dispositivos LTC (comutação de tap sob carga), o ângulo de defasagem dos transformadores defasadores, a susceptância de capacitores e de reatores em paralelo, etc.

Devido ao fato de que os sistemas elétricos recebem injeções de potência de várias unidades de geração, e de que eles suprem potência para um grande número de cargas que são dispersas em áreas geográficas de grandes dimensões, a tarefa de manter as tensões dentro dos limites requeridos pode ser bastante complexa. O controle de tensão é largamente reconhecido como sendo fortemente relacionado ao controle da potência reativa [9]. Porém, face ao elevado número de variáveis de controle que podem ser manipuladas, associado com o elevado número de restrições que são impostas sobre a operação do sistema, a seleção apropriada e a coordenação dos equipamentos para exercer esse controle está entre os maiores desafios da engenharia de potência.

No entanto, essa tarefa pode ser eficientemente executada por programas de Fluxo de Potência Ótimo (FPO) existentes nos centros de controle do sistema elétrico, seja de forma totalmente automática ou como ferramenta de auxílio a tomada de decisão pelo operador. O FPO é uma sofisticada ferramenta computacional que se utiliza de técnicas avançadas de otimização na determinação do estado operativo ótimo do sistema, minimizando ou maximizando um determinado índice de desempenho do sistema enquanto satisfaz um conjunto de restrições impostas sobre a operação.

2.1 Forma Geral do Problema

Vários problemas de FPO podem ser expressos na seguinte forma geral de um problema de programação não-linear:

$$\text{Minimize } f(x) \tag{2.1a}$$

$$\text{sujeito a } g(x) = 0 \tag{2.1b}$$

$$l \leq x \leq u \tag{2.1c}$$

em que:

- $x \in \mathbb{R}^n$ é um vetor com as variáveis de decisão explícitas, incluindo as variáveis de controle (magnitude das tensões das barras de geração, taps dos transformadores com LTC, compensação de reativos em paralelo, potência ativa dos geradores, fator de carregamento, etc) e as variáveis dependentes que não são funções (ângulo de fase das tensões, magnitude das tensões das barras de carga, potência reativa dos geradores, etc);
- $f : \mathbb{R}^n \mapsto \mathbb{R}$ é a função escalar que representa um dado objetivo de otimização da operação ou do planejamento do SEP, tal como o custo da geração, perdas de potência no sistema de transmissão, corte de carga para tornar operativo um sistema não operativo, etc;
- $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ é um vetor não-linear que contém as equações usuais de balanço de potência nas barras, ocasionalmente aumentado por algumas restrições especiais de igualdades, tal como o controle do fluxo de potência entre sistemas numa operação em *pool*, ou fluxos que são estabelecidos em um determinado valor, etc;
- $l \in \mathbb{R}^n$ e $u \in \mathbb{R}^n$ são vetores de limites mínimos e máximos sobre as variáveis x , correspondentes a limites físicos de equipamentos e limites operacionais do sistema.

Em vez de minimizar, o objetivo pode ser de maximizar uma função. Dentre os objetivos usualmente utilizados, encontram-se:

- *Minimização de custos de geração*: minimiza o custo da geração da potência ativa para a configuração base da rede elétrica enquanto assegura a viabilidade nas configurações de contingências;
 - *Minimização de perdas ativas*: minimiza as perdas ativas na configuração base enquanto assegura a viabilidade nas configurações de contingências;
 - *Minimização de corte de carga*: minimiza o corte de carga para corrigir violações de restrições operacionais tais como sobrecargas em circuitos, problemas de tensão, etc, no caso base e nas configurações de contingências;
 - *Minimização do movimento de variáveis de controle*: determina o menor número de dispositivos de controle a serem ajustados de forma a corrigir violações de restrições operacionais;
-

- *Maximização do fluxo de potência ativa em um conjunto de circuitos:* maximiza o fluxo de potência ativa através de um conjunto de circuitos na configuração base enquanto assegura a viabilidade nas configurações de contingências;
- *Maximização da carga em um conjunto de barras:* maximiza a carga num conjunto de barras, mantendo o mesmo fator de potência da carga e a viabilidade no caso base e configurações de contingências;
- *Maximização da potência transferida entre duas barras:* maximiza a potência transferida entre duas barras, mantendo a viabilidade no caso base e configurações de contingências.

Para otimizar (minimizar ou maximizar) a função objetivo especificada, as seguintes variáveis de controle podem ser utilizadas:

- Potência reativa de bancos de capacitores e indutores chaveáveis, de compensadores síncronos, de compensadores estáticos, etc;
- Tapes dos transformadores com dispositivo LTC;
- Tensão terminal dos geradores;
- Potência ativa dos geradores;
- Ângulo dos defasadores controlando fluxo de potência ativa;
- Corte de carga, etc.

O conjunto de restrições geralmente é constituído por:

- Limites (máximos e mínimos) sobre as tensões das barras;
 - Limites sobre os fluxos nos circuitos (MVA, MW e/ou MVAr);
 - Limites sobre os tapes dos transformadores com dispositivo LTC;
 - Limites sobre os ângulos dos defasadores;
 - Limites sobre as gerações de potência ativa e potência reativa;
-

- Limites sobre as injeções de potência reativa de fontes em paralelo controláveis;
- Intercâmbio de potência ativa e reativa entre áreas, etc.

Além do caso base, a solução pode contemplar a operação do sistema sob contingências, tais como:

- Perda de circuito (linha de transmissão ou transformador);
- Perda de barramento;
- Perda de gerador;
- Adição de circuito (reconfiguração), etc.

Idealmente, a formulação de FPO deve permitir a definição de controles e limites diferentes na condição base e nas configurações de contingências.

2.2 Equações Básicas de Fluxo de Potência

De fundamental importância para a formulação matemática de uma ampla variedade de problemas de FPO são as equações básicas do problema de fluxo de potência tradicional, ou seja, o fluxo de potência não-otimizado. Para o desenvolvimento dessas equações são apresentados inicialmente os modelos de dois componentes básicos da rede elétrica: as linhas de transmissão e os transformadores. No caso de transformadores, é apresentado um modelo bastante geral, que modela até os sofisticados transformadores defasadores.

Considerando que a implementação computacional do programa de FPO utilizado nesta Dissertação é feita no ambiente MATLAB, e que as operações em MATLAB são mais eficientemente realizadas em termos de variáveis complexas, em especial, matrizes e vetores complexos, ênfase especial é dada ao desenvolvimento das expressões matemáticas em termos de grandezas complexas.

2.2.1 Fluxos de Correntes e Potências nos Circuitos

Linhas de Transmissão: Nos estudos de fluxo de potência as linhas de transmissão são geralmente modeladas pelo circuito Π ilustrado na Figura 2.1, em que $\vec{y}_{ij} = g_{ij} + jb_{ij}$

representa a admitância série da linha, e $\vec{y}_{ij}^{\text{sh}} = jb_{ij}^{\text{sh}}$ representa o efeito do carregamento capacitivo da linha (metade do carregamento total concentrado em cada terminal).

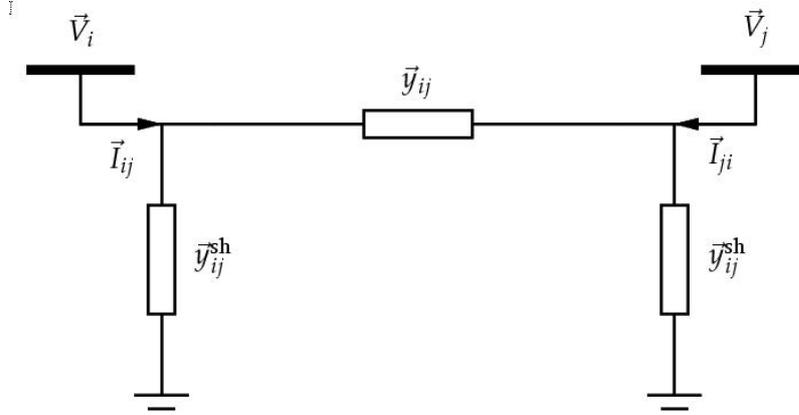


Figura 2.1: Representação geral de linhas de transmissão.

Deduz-se, da Figura 2.1, que a corrente complexa que flui do nó i para o nó j é dada por:

$$\begin{aligned}\vec{I}_{ij} &= \vec{y}_{ij}^{\text{sh}} \vec{V}_i + \vec{y}_{ij} (\vec{V}_i - \vec{V}_j) \\ &= (\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \vec{V}_i - \vec{y}_{ij} \vec{V}_j\end{aligned}\quad (2.2)$$

Analogamente, a corrente complexa que flui do nó j para o nó i é calculada por:

$$\begin{aligned}\vec{I}_{ji} &= \vec{y}_{ij}^{\text{sh}} \vec{V}_j + \vec{y}_{ij} (\vec{V}_j - \vec{V}_i) \\ &= -\vec{y}_{ij} \vec{V}_i + (\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \vec{V}_j\end{aligned}\quad (2.3)$$

Matricialmente, temos a relação entre os fluxos de correntes \vec{I}_{ij} e \vec{I}_{ji} e as tensões de nós \vec{V}_i e \vec{V}_j dada por:

$$\begin{bmatrix} \vec{I}_{ij} \\ \vec{I}_{ji} \end{bmatrix} = \begin{bmatrix} (\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) & -\vec{y}_{ij} \\ -\vec{y}_{ij} & (\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \end{bmatrix} \begin{bmatrix} \vec{V}_i \\ \vec{V}_j \end{bmatrix}.\quad (2.4)$$

A potência complexa que flui do nó i para o nó j é calculada por:

$$\begin{aligned}\vec{S}_{ij} &= \vec{V}_i \vec{I}_{ij}^* \\ &= \vec{V}_i [(\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \vec{V}_i - \vec{y}_{ij} \vec{V}_j]^* \\ &= \vec{V}_i [\vec{y}_{ij} (\vec{V}_i - \vec{V}_j) + \vec{y}_{ij}^{\text{sh}} \vec{V}_i]^*\end{aligned}\quad (2.5)$$

A potência complexa que flui do nó j para o nó i é calculada por:

$$\begin{aligned} \vec{S}_{ji} &= \vec{V}_j \vec{I}_{ji}^* \\ &= \vec{V}_j [-\vec{y}_{ij} \vec{V}_i + (\vec{y}_{ij} + \vec{y}_{ij}^{sh}) \vec{V}_j]^* \\ &= \vec{V}_j [\vec{y}_{ij} (\vec{V}_j - \vec{V}_i) + \vec{y}_{ij}^{sh} \vec{V}_j]^* \end{aligned} \quad (2.6)$$

Matricialmente, temos a relação entre os fluxos de potências complexas e as tensões de nós dada por:

$$\begin{bmatrix} \vec{S}_{ij} \\ \vec{S}_{ji} \end{bmatrix} = \begin{bmatrix} \vec{V}_i & 0 \\ 0 & \vec{V}_j \end{bmatrix} \left(\begin{bmatrix} (\vec{y}_{ij} + \vec{y}_{ij}^{sh}) & -\vec{y}_{ij} \\ -\vec{y}_{ij} & (\vec{y}_{ij} + \vec{y}_{ij}^{sh}) \end{bmatrix} \begin{bmatrix} \vec{V}_i \\ \vec{V}_j \end{bmatrix} \right)^* \quad (2.7)$$

Transformadores em Fase e Defasadores: Uma representação geral de transformadores (em-fase e defasadores) é ilustrada na Figura 2.2, a qual consiste de uma admitância série \vec{y}_{ij} e um auto-transformador ideal com relação de transformação complexa na forma $1 : \vec{t}_{ij}$, ou seja, o tape \vec{t}_{ij} no lado da admitância série. Temos que $\vec{t}_{ij} = t_{ij} e^{j\phi_{ij}}$, sendo $\phi_{ij} = 0$ no caso de transformadores em-fase.

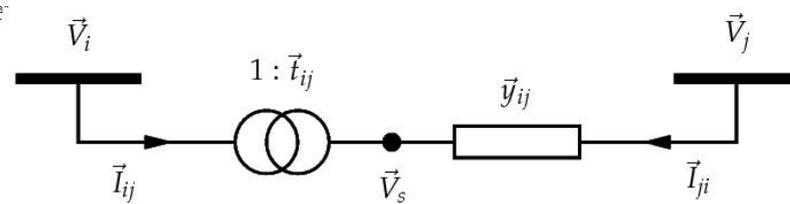


Figura 2.2: Representação geral de transformadores.

Para a representação de transformadores da Figura 2.2, na qual o tape está do lado da admitância série \vec{y}_{ij} , a relação entre as tensões dos nós terminais do transformador ideal é como segue:

$$\frac{\vec{V}_s}{\vec{V}_i} = \vec{t}_{ij} \quad (2.8)$$

Uma vez que as potências complexas na entrada e na saída do transformador ideal devem ser iguais, temos então que

$$\vec{V}_i \vec{I}_{ij}^* + \vec{V}_s \vec{I}_{ji}^* = 0 \quad (2.9)$$

Podemos agora, a partir das relações (2.8) e (2.9), deduzir a relação entre as correntes

complexas \vec{I}_{ij} e \vec{I}_{ji} injetadas nos terminais do transformador, como

$$\begin{aligned}\frac{\vec{I}_{ij}}{\vec{I}_{ji}} &= -\frac{\vec{V}_s^*}{\vec{V}_i^*} \\ &= -\vec{t}_{ij}^* \end{aligned} \quad (2.10)$$

Podemos deduzir, da Figura 2.2, que a corrente complexa que flui do nó j para o nó i é dada por

$$\vec{I}_{ji} = \vec{y}_{ij}(\vec{V}_j - \vec{V}_s) \quad (2.11)$$

Utilizando as expressões (2.10) e (2.11), calculamos a corrente complexa que flui do nó i para o nó j como

$$\begin{aligned}\vec{I}_{ij} &= -\vec{t}_{ij}^* \vec{I}_{ji} \\ &= -\vec{t}_{ij}^* \vec{y}_{ij}(\vec{V}_j - \vec{V}_s) \end{aligned} \quad (2.12)$$

Matricialmente, temos a relação entre os fluxos de correntes e as tensões dos nós terminais na forma

$$\begin{bmatrix} \vec{I}_{ij} \\ \vec{I}_{ji} \end{bmatrix} = \begin{bmatrix} (\vec{t}_{ij} \vec{t}_{ij}^*) \vec{y}_{ij} & -\vec{t}_{ij}^* \vec{y}_{ij} \\ -\vec{t}_{ij} \vec{y}_{ij} & \vec{y}_{ij} \end{bmatrix} \begin{bmatrix} \vec{V}_i \\ \vec{V}_j \end{bmatrix}. \quad (2.13)$$

A potência complexa que flui do nó i para o nó j é calculada por:

$$\begin{aligned}\vec{S}_{ij} &= \vec{V}_i \vec{I}_{ij}^* \\ &= \vec{V}_i (-\vec{t}_{ij}^* \vec{y}_{ij} (\vec{V}_j - \vec{V}_s))^* \\ &= \vec{V}_s (\vec{y}_{ij} (\vec{V}_s - \vec{V}_j))^* \end{aligned} \quad (2.14)$$

A potência complexa que flui do nó j para o nó i é calculada por:

$$\begin{aligned}\vec{S}_{ji} &= \vec{V}_j \vec{I}_{ji}^* \\ &= \vec{V}_j (\vec{y}_{ij} (\vec{V}_j - \vec{V}_s))^* \end{aligned} \quad (2.15)$$

Matricialmente, temos a relação entre os fluxos de potências complexas e as tensões de nós na forma

$$\begin{bmatrix} \vec{S}_{ij} \\ \vec{S}_{ji} \end{bmatrix} = \begin{bmatrix} \vec{V}_i & 0 \\ 0 & \vec{V}_j \end{bmatrix} \left(\begin{bmatrix} (\vec{t}_{ij} \vec{t}_{ij}^*) \vec{y}_{ij} & -\vec{t}_{ij}^* \vec{y}_{ij} \\ -\vec{t}_{ij} \vec{y}_{ij} & \vec{y}_{ij} \end{bmatrix} \begin{bmatrix} \vec{V}_i \\ \vec{V}_j \end{bmatrix} \right)^*. \quad (2.16)$$

Equações Gerais de Fluxos: Comparando-se as equações de transformadores (2.13) e (2.16) com as equações de linhas de transmissão (2.4) e (2.7), podemos deduzir as seguintes equações gerais de fluxos de correntes e de potências complexas:

$$\begin{bmatrix} \vec{I}_{ij} \\ \vec{I}_{ji} \end{bmatrix} = \begin{bmatrix} (\vec{t}_{ij}\vec{t}_{ij}^*\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) & -\vec{t}_{ij}^*\vec{y}_{ij} \\ -\vec{t}_{ij}\vec{y}_{ij} & (\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \end{bmatrix} \begin{bmatrix} \vec{V}_i \\ \vec{V}_j \end{bmatrix} \quad (2.17)$$

$$\begin{bmatrix} \vec{S}_{ij} \\ \vec{S}_{ji} \end{bmatrix} = \begin{bmatrix} \vec{V}_i & 0 \\ 0 & \vec{V}_j \end{bmatrix} \left(\begin{bmatrix} (\vec{t}_{ij}\vec{t}_{ij}^*\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) & -\vec{t}_{ij}^*\vec{y}_{ij} \\ -\vec{t}_{ij}\vec{y}_{ij} & (\vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \end{bmatrix} \begin{bmatrix} \vec{V}_i \\ \vec{V}_j \end{bmatrix} \right)^* \quad (2.18)$$

em que:

- $t_{ij} = 1$ e $\phi_{ij} = 0$, no caso de linhas de transmissão;
- $\vec{y}_{ij}^{\text{sh}} = 0$ e $\phi_{ij} = 0$, no caso de transformadores em fase;
- $\vec{y}_{ij}^{\text{sh}} = 0$ e $t_{ij} = 1$, no caso de defasadores puros; e
- $\vec{y}_{ij}^{\text{sh}} = 0$, no caso de transformadores defasadores.

2.2.2 Injeções de Potências nos Nós

A potência líquida injetada numa barra é definida como sendo a diferença entre a geração total e a carga total naquela barra. Essa potência vem a ser também a soma das potências que fluem nos circuitos que tem aquela barra como um de seus terminais. Ou seja,

$$\begin{aligned} \vec{S}_i &= \vec{S}_i^G - \vec{S}_i^D \\ &= \sum_j \vec{S}_{ji} + \sum_k \vec{S}_{ik} \end{aligned} \quad (2.19)$$

A potência complexa líquida injetada no nó i é calculada também por

$$\vec{S}_i = \vec{V}_i \vec{I}_i^* \quad (2.20)$$

em que \vec{I}_i é a corrente líquida injetada no nó i , ou seja, a i -ésima componente no vetor \vec{I} da equação de rede:

$$\vec{Y}\vec{V} = \vec{I} \quad (2.21)$$

Temos então de (2.20) e (2.21) que as potências complexas líquidas injetadas nos nós i e j são calculadas por

$$\vec{S}_i = \vec{V}_i \left(\sum_{k \in \mathcal{I}} \vec{Y}_{ik} \vec{V}_k \right)^* \quad (2.22)$$

$$\vec{S}_j = \vec{V}_j \left(\sum_{k \in \mathcal{J}} \vec{Y}_{jk} \vec{V}_k \right)^* \quad (2.23)$$

em que \vec{Y}_{jk} e \vec{Y}_{ik} são elementos da matriz admitância de nó \vec{Y} ; \mathcal{I} é o conjunto das barras adjacentes à barra i , incluindo a própria barra i ; e \mathcal{J} é o conjunto das barras adjacentes à barra j , incluindo a própria barra j .

Admitindo que as barras i e j são interligadas, seja por linhas de transmissão ou por transformadores, explicitando em (2.22) e (2.23) os termos das tensões dessas barras, \vec{V}_i e \vec{V}_j , obtemos

$$\vec{S}_i = \vec{V}_i \left(\vec{Y}_{ii} \vec{V}_i + \vec{Y}_{ij} \vec{V}_j + \sum_{(k \neq i, j) \in \mathcal{I}} \vec{Y}_{ik} \vec{V}_k \right)^* \quad (2.24)$$

$$\vec{S}_j = \vec{V}_j \left(\vec{Y}_{ji} \vec{V}_i + \vec{Y}_{jj} \vec{V}_j + \sum_{(k \neq i, j) \in \mathcal{J}} \vec{Y}_{jk} \vec{V}_k \right)^* \quad (2.25)$$

Considerando que \vec{Y}_{ii}^0 , \vec{Y}_{ij}^0 , \vec{Y}_{ji}^0 e \vec{Y}_{jj}^0 , denotam elementos da matriz admitância de nó antes da inclusão do circuito- ij (que pode ser uma linha ou um transformador), temos que a inclusão do circuito- ij (que assumimos ser um transformador) modifica as equações de injeções de potência da forma

$$\vec{S}_i = \vec{V}_i \left((\vec{Y}_{ii}^0 + \vec{t}_{ij} \vec{t}_{ij}^* \vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \vec{V}_i + (\vec{Y}_{ij}^0 - \vec{t}_{ij} \vec{y}_{ij}) \vec{V}_j + \sum_{(k \neq i, j) \in \mathcal{I}} \vec{Y}_{ik} \vec{V}_k \right)^* \quad (2.26)$$

$$\vec{S}_j = \vec{V}_j \left((\vec{Y}_{ji}^0 - \vec{t}_{ij} \vec{y}_{ij}) \vec{V}_i + (\vec{Y}_{jj}^0 + \vec{y}_{ij} + \vec{y}_{ij}^{\text{sh}}) \vec{V}_j + \sum_{(k \neq i, j) \in \mathcal{J}} \vec{Y}_{jk} \vec{V}_k \right)^* \quad (2.27)$$

2.2.3 Equações de Balanço de Potência

As formulações matemáticas dos problemas de FPO fazem uso das equações de balanço de potência ativa e potência reativa nas barras do sistema. Faz-se necessária, portanto, a identificação das componentes real (potência ativa) e imaginária (potência reativa) das injeções de potência complexa nas barras. Uma vez que

$$\vec{S}_i = P_i + jQ_i \quad (2.28)$$

as injeções líquidas de potência ativa ($P_i = \Re\{\vec{S}_i\}$) e de potência reativa ($Q_i = \Im\{\vec{S}_i\}$) são dadas por

$$P_i = V_i \sum_{j \in \mathcal{I}} V_j (G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)) \quad (2.29)$$

$$Q_i = V_i \sum_{j \in \mathcal{I}} V_j (G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j)) \quad (2.30)$$

em que V_i e θ_i são, respectivamente, a magnitude e o ângulo de fase da tensão complexa da barra i , $\widehat{V}_i = V_i e^{j\theta_i}$, G_{ij} é o elemento ij da matriz condutância de barra G , e B_{ij} é o elemento ij da matriz susceptância de barra B . Os tapes de transformadores estão presentes implicitamente em elementos das matrizes G e B .

Uma vez que

$$\vec{S}_{ij} = P_{ij} + jQ_{ij} \quad (2.31)$$

os fluxos de potência ativa, $P_{ij} = \Re\{\vec{S}_{ij}\}$, e de potência reativa, $Q_{ij} = \Im\{\vec{S}_{ij}\}$, no sentido do nó i para o nó j , são calculados por

$$P_{ij} = V_i^2 g_{ij} - V_i V_j g_{ij} \cos(\theta_i - \theta_j) - V_i V_j b_{ij} \sin(\theta_i - \theta_j) \quad (2.32)$$

$$Q_{ij} = -V_i^2 (b_{ij} + b_{ij}^{sh}) + V_i V_j b_{ij} \cos(\theta_i - \theta_j) - V_i V_j g_{ij} \sin(\theta_i - \theta_j) \quad (2.33)$$

e no sentido do nó j para o nó i , por

$$P_{ji} = V_j^2 g_{ij} - V_i V_j g_{ij} \cos(\theta_i - \theta_j) + V_i V_j b_{ij} \sin(\theta_i - \theta_j) \quad (2.34)$$

$$Q_{ji} = -V_j^2 (b_{ij} + b_{ij}^{sh}) + V_i V_j b_{ij} \cos(\theta_i - \theta_j) + V_i V_j g_{ij} \sin(\theta_i - \theta_j) \quad (2.35)$$

em que g_{ij} , b_{ij} e b_{ij}^{sh} são, respectivamente, a condutância série, a susceptância série e a susceptância em paralelo do circuito- ij . A partir dos fluxos de potência ativa (2.32) e (2.34), em sentidos opostos, calcula-se a perda de potência ativa no circuito ij como

$$P_{ij} + P_{ji} = g_{ij}(V_i^2 + V_j^2 - 2V_iV_j \cos(\theta_i - \theta_j)). \quad (2.36)$$

As perdas ativas globais no sistema de transmissão, P_{perdas} , são calculadas como a soma das perdas ativas em todos os circuitos do sistema, ou seja,

$$P_{\text{perdas}} = \sum_{(i,j) \in \mathcal{B}} g_{ij}(V_i^2 + V_j^2 - 2V_iV_j \cos(\theta_i - \theta_j)). \quad (2.37)$$

As equações (2.29) a (2.37) formam a base para o desenvolvimento dos modelos de FPO.

2.3 Modelos de Fluxo de Potência Ótimo

Problemas de FPO podem ser matematicamente formulados de várias maneiras. Nesta seção são apresentadas as formulações em quatro variantes importantes nessa ampla família de problemas de otimização da operação: (a) minimização de perdas ativas no sistema de transmissão, conhecido como despacho ótimo da potência reativa; (b) minimização do corte de carga; (c) maximização da capacidade de carregamento; e (d) maximização da capacidade de transferência simultânea da potência.

Os seguintes conjuntos de índices são utilizados neste capítulo: representa-se por \mathcal{N} o conjunto de todas as barras do sistema, por $\tilde{\mathcal{N}}$, o conjunto de todas as barras exceto a barra de folga, por \mathcal{G} , o conjunto de barras de *geração*, por \mathcal{F} , o conjunto de barras de carga com fontes de reativos em paralelo *fixas* e por \mathcal{C} , o conjunto de barras de carga *candidatas* ao controle da potência reativa. As seguintes relações entre conjuntos são observadas: $\mathcal{N} = \mathcal{G} \cup \mathcal{F} \cup \mathcal{C}$ e $\mathcal{G} \cap \mathcal{F} = \mathcal{G} \cap \mathcal{C} = \mathcal{F} \cap \mathcal{C} = \emptyset$. Representa-se por \mathcal{N}_i o conjunto de todas as barras eletricamente vizinhas a barra i , ou seja, que estão diretamente conectadas a barra i . O conjunto de pares de índices ordenados

$$\mathcal{B} = \{(i, j) \mid i \in \mathcal{N}, j \in \mathcal{N}_i \text{ e } j > i\}$$

é definido como sendo o conjunto dos pares de barras terminais de todos os ramos (linhas

de transmissão e transformadores) do sistema. Define-se $\mathcal{T} \subset \mathcal{B}$ como o conjunto das barras terminais (i, j) dos transformadores com dispositivo LTC.

2.3.1 Minimização de Perdas Ativas na Transmissão

O problema de minimização de perda ativa na transmissão, como formulado nesta seção, apresenta como restrições: as equações básicas de balanço de potência ativa e reativa nas barras; os limites de operação em relação aos níveis de tensões em todas as barras; os limites de operação em relação a geração de potência reativa pelos geradores; e os limites físicos das susceptâncias em paralelo e dos tapes dos transformadores com LTC.

Assume-se que as injeções de potência ativa em todas as barras do sistema, exceto a barra de folga, são conhecidas e permanecem fixas nos valores definidos pelo despacho econômico (DE) de geração, ou seja, as gerações de potência ativa não são consideradas como variáveis do problema. As variáveis de controle (aquelas que podem ser diretamente manipuladas pelo operador) são: as tensões terminais dos geradores, as susceptâncias em paralelo dos capacitores e reatores, e os tapes dos transformadores com dispositivo LTC. As variáveis de estado ou dependentes são: as tensões nas barras de carga, o ângulo de fase das tensões nodais, e a potência reativa dos geradores.

De forma compacta, o problema de minimização de perdas ativas no sistema de transmissão pode ser matematicamente expresso como segue:

$$\begin{aligned}
& \text{Minimize } P_{\text{perdas}}(V, \theta, t) \\
& \text{sujeito a } \begin{aligned}
& P_i(V, \theta, t) + P_{D_i}(V_i) - P_{G_i} = 0, & \text{para } i \in \mathcal{N} \\
& Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} = 0, & \text{para } i \in \mathcal{G} \\
& Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} = 0, & \text{para } i \in \mathcal{F} \\
& Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} - b_i^{sh} V_i^2 = 0, & \text{para } i \in \mathcal{C} \\
& F_{ij}(V, \theta, t) - F_{ij} = 0, & \{(i, j)\} \subseteq \mathcal{B} \\
& V_i^{\min} \leq V_i \leq V_i^{\max}, & \text{para } i \in \mathcal{N} \\
& t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, & \{(i, j)\} \subseteq \mathcal{T} \\
& P_i^{\min} \leq P_{G_i} \leq P_i^{\max}, & \text{para } i \in \mathcal{G} \\
& Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, & \text{para } i \in \mathcal{G} \\
& b_i^{\min} \leq b_i^{sh} \leq b_i^{\max}, & \text{para } i \in \mathcal{C} \\
& F_{ij}^{\min} \leq F_{ij} \leq F_{ij}^{\max}, & \{(i, j)\} \subseteq \mathcal{B}
\end{aligned} \tag{2.38}
\end{aligned}$$

em que $P_{\text{perdas}}(V, \theta, t)$, $P_i(V, \theta, t)$ e $Q_i(V, \theta, t)$ são as funções perdas elétricas, injeção líquida de potência ativa e injeção líquida de potência reativa, calculadas pelas expressões (2.37), (2.29) e (2.30), respectivamente. $P_{D_i}(V_i)$ e $Q_{D_i}(V_i)$ são as demandas de potências ativa e reativa na barra i , respectivamente, as quais admitimos poderem variar com o módulo da tensão da barra, segundo o modelo ZIP de carga [10]:

$$P_{D_i}(V_i) = (a_p + b_p V_i + c_p V_i^2) P_i^{\text{nom}} \quad (2.39)$$

$$Q_{D_i}(V_i) = (a_q + b_q V_i + c_q V_i^2) Q_i^{\text{nom}} \quad (2.40)$$

em que $a + b + c = 1$, e P_i^{nom} e Q_i^{nom} são as demandas na tensão nominal $V_i = 1$ p.u.

A variável F_{ij} representa o fluxo no ramo ij , seja de potência aparente, potência ativa, potência reativa ou de corrente. Observe que, para reduzir o número de restrições do problema, são incluídas restrições de limites de fluxos apenas num subconjunto dos ramos, aqueles considerados mais críticos e com maior possibilidade de violação.

O primeiro grupo de restrições de igualdade impõe o balanço de potência ativa nas barras do sistema, exceto a barra de folga, enquanto o segundo, terceiro e quarto grupos de restrições impõem o balanço de potência reativa nas barras de geração, nas barras de carga com reativo fixo, e nas barras de carga com reativo controlável, respectivamente. A quinta restrição de igualdade juntamente o último conjunto de restrições de desigualdade impõe os limites de carregamento nos circuitos selecionados. Outras restrições de igualdade podem ser incorporadas à formulação, como restrições de controle de intercâmbio entre áreas.

As desigualdades representam os limites físicos sobre as susceptâncias em paralelo e os taps dos transformadores com LTC, e os limites operacionais em relação ao nível das tensões representam a geração de reativos pelos geradores e os fluxos nas linhas e nos transformadores. Observe que as equações básicas do problema de fluxo de potência tradicional (não-otimizado) fazem parte do conjunto de restrições do problema (2.38).

Alternativamente, pode-se definir uma função objetivo que considere as perdas ativas apenas em uma certa área do sistema, como segue:

$$\text{Minimize} \quad \sum_{(i,j) \in \Omega} P_{ij} + P_{ji},$$

em que Ω é o conjunto dos índices das barras terminais dos circuitos que definem a área de

interesse onde as perdas devem ser minimizadas. Alternativamente, pode-se definir a função objetivo como sendo as perdas reativas no sistema ou em uma certa área do sistema. Essa função tem vários efeitos benéficos [1]. Ela quase minimiza as perdas ativas, mantém o perfil de tensões plano e quase maximiza as reservas de potência reativa dos geradores.

2.3.2 Minimização do Corte de Carga

Em várias situações na operação de um sistema de potência, esquemas de cortes de carga são utilizados para reduzir a carga atual do sistema para um nível que possa ser suprido com segurança pela geração e pela topologia da rede disponível. Esse tipo de situação pode ocorrer, por exemplo, quando um sistema fortemente carregado é submetido à perda de um equipamento importante, tal como uma linha de transmissão, um transformador ou um gerador, de maneira tal que ele não pode mais suprir toda a demanda da carga com a geração e a rede de transmissão pós-contingência.

O problema de minimização do corte de carga consiste em determinar o corte mínimo da carga que é necessário para restaurar a viabilidade da operação (atender limites que estão sendo violados), ou mesmo para restaurar a solvabilidade das equações de balanço de potência nas barras as quais, sem uma redução na carga, não teriam solução. Uma formulação possível para esse problema, entre várias possíveis, é como segue:

$$\begin{aligned}
& \text{Minimize } \tau \\
& \text{sujeito a} \quad P_i(V, \theta, t) + P_{D_i}^0(V_i)(1 - \tau) - P_{G_i} = 0, & \text{para } i \in \mathcal{N} \\
& \quad Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 - \tau) - Q_{G_i} = 0, & \text{para } i \in \mathcal{G} \\
& \quad Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 - \tau) - Q_{G_i} = 0, & \text{para } i \in \mathcal{F} \\
& \quad Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 - \tau) - Q_{G_i} - b_i^{sh} V_i^2 = 0, & \text{para } i \in \mathcal{C} \\
& \quad F_{ij}(V, \theta, t) - F_{ij} = 0, & \{(i, j)\} \subseteq \mathcal{B} \\
& \quad V_i^{\min} \leq V_i \leq V_i^{\max}, & \text{para } i \in \mathcal{N} \\
& \quad t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, & \{(i, j)\} \subseteq \mathcal{T} \\
& \quad P_i^{\min} \leq P_{G_i} \leq P_i^{\max}, & \text{para } i \in \mathcal{G} \\
& \quad Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, & \text{para } i \in \mathcal{G} \\
& \quad b_i^{\min} \leq b_i^{sh} \leq b_i^{\max}, & \text{para } i \in \mathcal{C} \\
& \quad F_{ij}^{\min} \leq F_{ij} \leq F_{ij}^{\max}, & \{(i, j)\} \subseteq \mathcal{B}
\end{aligned} \tag{2.41}$$

em que τ é o fator de corte de carga nas barras, e $P_{D_i}^0$ e $Q_{D_i}^0$ são as demandas originais das potências ativa e reativa, ou seja, as demandas anteriores à ocorrência da contingência.

Observe que a formulação do problema de corte de carga é relativamente simples e pouco difere da formulação para minimização de perdas elétricas, principalmente com relação ao conjunto de restrições, o que facilita a sua implementação a partir de um programa existente para minimização de perdas. A diferença básica está na multiplicação da potência da carga pelo fator de corte da carga a ser minimizado. Assim, um novo perfil de carga pode ser determinado de forma que os limites operacionais sejam atendidos com o menor corte de carga possível.

Observe ainda que essa formulação considera que o mesmo percentual de corte de carga é aplicado a todas as barras do sistema, o que pode ser impraticável, uma vez que a área mais crítica do sistema que é a diretamente afetada pela contingência poderá requerer um percentual de carga bem maior do que o necessário em outras barras/áreas do sistema. Além do fator de corte comum, o mesmo está presente em todas as barras do sistema, enquanto na prática ocorre somente em determinados conjuntos de barras.

Para diferenciar os consumidores no evento de um corte de carga, pode-se considerar a formulação:

$$\begin{aligned}
& \text{Minimize} && \sum_{i \in \mathcal{N}} w_i \tau_i \\
& \text{sujeito a} && P_i(V, \theta, t) + P_{D_i}^0(V_i)(1 - \tau_i) - P_{G_i} = 0, && \text{para } i \in \mathcal{N} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 - \tau_i) - Q_{G_i} = 0, && \text{para } i \in \mathcal{G} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 - \tau_i) - Q_{G_i} = 0, && \text{para } i \in \mathcal{F} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 - \tau_i) - Q_{G_i} - b_i^{sh} V_i^2 = 0, && \text{para } i \in \mathcal{C} \\
& && F_{ij}(V, \theta, t) - F_{ij} = 0, && \{(i, j)\} \subseteq \mathcal{B} \\
& && V_i^{\min} \leq V_i \leq V_i^{\max}, && \text{para } i \in \mathcal{N} \\
& && t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{T} \\
& && P_i^{\min} \leq P_{G_i} \leq P_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && b_i^{\min} \leq b_i^{sh} \leq b_i^{\max}, && \text{para } i \in \mathcal{C} \\
& && F_{ij}^{\min} \leq F_{ij} \leq F_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{B}
\end{aligned} \tag{2.42}$$

em que w_i é um coeficiente de ponderação da importância da barra i no corte de carga.

Observa-se nas formulações (2.41) e (2.42) que o mesmo parâmetro de corte de carga, τ ou τ_i , é utilizado para o corte da carga ativa $P_{D_i}^0$ e da carga reativa $Q_{D_i}^0$. Isso significa que o fator de potência das cargas permanecerá constante. Todavia, pode-se facilmente modelar o corte com modificação do fator de potência.

2.3.3 Maximização do Carregamento do Sistema

Para um estado de operação viável qualquer de um sistema, o problema de maximização do carregamento é o de determinar o máximo crescimento da carga que o sistema ainda pode suportar (seja a carga total do sistema, a carga em uma certa área do sistema, ou a carga em uma determinada barra ou conjunto de barras), ainda satisfazendo às restrições de operação da empresa e dos equipamentos dos consumidores. Um programa capaz de calcular o máximo crescimento da carga do sistema, sem contemplar novos investimentos na geração e no sistema de transmissão, é uma ferramenta bastante valiosa em estudos de estabilidade de tensão. Dependendo de como o problema é formulado, sua solução pode ser utilizada para definir contratos com grandes consumidores, definir transações entre sistemas interligados, etc.

Uma formulação particular do problema de maximização do carregamento do sistema é como segue:

$$\begin{aligned}
& \text{Maximize} && \tau \\
& \text{sujeito a} && P_i(V, \theta, t) + P_{D_i}^0(V_i)(1 + \tau) - P_{G_i} = 0, && \text{para } i \in \mathcal{N} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 + \tau) - Q_{G_i} = 0, && \text{para } i \in \mathcal{G} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 + \tau) - Q_{G_i} = 0, && \text{para } i \in \mathcal{F} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 + \tau) - Q_{G_i} - b_i^{sh} V_i^2 = 0, && \text{para } i \in \mathcal{C} \\
& && F_{ij}(V, \theta, t) - F_{ij} = 0, && \{(i, j)\} \subseteq \mathcal{B} \\
& && V_i^{\min} \leq V_i \leq V_i^{\max}, && \text{para } i \in \mathcal{N} \\
& && t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{T} \\
& && P_i^{\min} \leq P_{G_i} \leq P_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && b_i^{\min} \leq b_i^{sh} \leq b_i^{\max}, && \text{para } i \in \mathcal{C} \\
& && F_{ij}^{\min} \leq F_{ij} \leq F_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{B}
\end{aligned} \tag{2.43}$$

em que τ é o fator de crescimento da carga.

Observe que para resolver o problema (2.43) parte-se de um estado de operação viável (ou seja, um estado no qual as restrições em (2.43) podem ser satisfeitas para $\tau = 0$) e então aumenta-se o parâmetro τ até que o conjunto viável de (2.43) no espaço definido por (V, θ, t) deixe de existir, pelo menos localmente, para o novo nível de carga: $(1 + \tau)P_{D_i}^0$ e $(1 + \tau)Q_{D_i}^0$. Portanto, o problema do máximo crescimento da carga trata com condições de operação altamente não-lineares. Observe também que há uma grande semelhança entre a formulação do problema de minimização do corte de carga (2.41) e o de maximização do carregamento do sistema (2.43). A principal diferença diz respeito ao sinal do fator de carregamento τ . As dificuldades numéricas que esses dois problemas impõem sobre a técnica de otimização empregada na solução são as mesmas.

De forma similar ao problema de minimização do corte de carga, podem-se considerar crescimentos de carga distintos nas barras, resolvendo-se o problema:

$$\begin{aligned}
& \text{Maximize} && \sum_{i \in \mathcal{N}} w_i \tau_i \\
& \text{sujeito a} && P_i(V, \theta, t) + P_{D_i}^0(V_i)(1 + \tau_i) - P_{G_i} = 0, && \text{para } i \in \mathcal{N} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 + \tau_i) - Q_{G_i} = 0, && \text{para } i \in \mathcal{G} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 + \tau_i) - Q_{G_i} = 0, && \text{para } i \in \mathcal{F} \\
& && Q_i(V, \theta, t) + Q_{D_i}^0(V_i)(1 + \tau_i) - Q_{G_i} - b_i^{sh} V_i^2 = 0, && \text{para } i \in \mathcal{C} \\
& && F_{ij}(V, \theta, t) - F_{ij} = 0, && \{(i, j)\} \subseteq \mathcal{B} \\
& && V_i^{\min} \leq V_i \leq V_i^{\max}, && \text{para } i \in \mathcal{N} \\
& && t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{T} \\
& && P_i^{\min} \leq P_{G_i} \leq P_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && b_i^{\min} \leq b_i^{sh} \leq b_i^{\max}, && \text{para } i \in \mathcal{C} \\
& && F_{ij}^{\min} \leq F_{ij} \leq F_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{B}
\end{aligned} \tag{2.44}$$

em que w_i é um coeficiente de diferenciação da barra i no crescimento da carga.

2.3.4 Maximização da Capacidade de Transferência Simultânea

Outro problema de grande interesse no planejamento da operação de um SEP é o da determinação da máxima potência que pode ser transferida de um sistema para outro interligado, ou de uma área para outra do mesmo sistema. A solução desse problema é conhecida como máxima capacidade de transferência simultânea, e o problema pode ser formulado da seguinte forma:

$$\begin{aligned}
& \text{Maximize} && \sum_{ij \in \Omega} P_{ij}(V, \theta, t) \\
& \text{sujeito a} && P_i(V, \theta, t) + P_{D_i}(V_i) - P_{G_i} = 0, && \text{para } i \in \mathcal{N} \\
& && Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} = 0, && \text{para } i \in \mathcal{G} \\
& && Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} = 0, && \text{para } i \in \mathcal{F} \\
& && Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} - b_i^{sh} V_i^2 = 0, && \text{para } i \in \mathcal{C} \\
& && F_{ij}(V, \theta, t) - F_{ij} = 0, && \{(i, j)\} \subseteq \mathcal{B} \\
& && V_i^{\min} \leq V_i \leq V_i^{\max}, && \text{para } i \in \mathcal{N} \\
& && t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{T} \\
& && P_i^{\min} \leq P_{G_i} \leq P_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, && \text{para } i \in \mathcal{G} \\
& && b_i^{\min} \leq b_i^{sh} \leq b_i^{\max}, && \text{para } i \in \mathcal{C} \\
& && F_{ij}^{\min} \leq F_{ij} \leq F_{ij}^{\max}, && \{(i, j)\} \subseteq \mathcal{B}
\end{aligned} \tag{2.45}$$

em que Ω é o conjunto dos índices (i, j) das barras terminais dos ramos nos quais se deseja maximizar o fluxo de potência ativa, ao mesmo tempo em que um conjunto de restrições é satisfeito.

2.4 As Técnicas Usuais de Solução

Os problemas de FPO são inevitavelmente problemas de programação não-linear não-convexos e de grande escala, bastante complicados nas aplicações práticas pela presença de um grande número de variáveis discretas, tais como os tapes dos transformadores, as susceptâncias dos capacitores em paralelo, etc.

Dada a sua importância nas atividades de planejamento e operação, o FPO tem sido um

tema de intensa pesquisa há cerca de quatro décadas [1, 11]. Métodos do gradiente foram as primeiras técnicas utilizadas para resolver um problema de FPO [12], o qual foi formulado matematicamente pela primeira vez por Carpentier em 1962.

Desde então, melhorias nas ferramentas de FPO têm sido alcançadas de duas maneiras principais: (a) formulações mais eficientes do problema, e (b) técnicas de otimização mais eficientes, flexíveis e robustas. As principais técnicas para resolver os problemas de FPO incluem métodos do gradiente reduzido, métodos baseados no Lagrangeano aumentado, funções de penalidades exatas e, principalmente, métodos baseados em aproximações locais, tais como Programação Linear Sucessiva (PLS) e Programação Quadrática Sucessiva (PQS).

As técnicas de PLS e PQS têm sido largamente utilizadas na otimização de sistemas de potência. Atualmente, elas podem tirar proveito dos eficientes métodos de Pontos-Interiores (PI) para resolver os subproblemas de Programação Linear (PL) ou Programação Quadrática (PQ) que são gerados a cada iteração PLS ou PQS [13, 14]. Entretanto, o processo de convergência da PLS e PQS é, entre outros fatores, altamente dependente da existência de um bom ponto de operação inicial para a aproximação local das funções não-lineares, o que nem sempre ocorre.

Por outro lado, tem sido crescente a necessidade de resolverem-se problemas de FPO na sua forma não-linear original. Portanto, técnicas de otimização baseadas em aproximações locais sucessivas, como PLS e QLS, tornaram-se menos atrativas. Por outro lado, a solução eficiente de FPOs na forma não-linear pode ser um problema bastante complexo. Várias condições sob as quais um algoritmo de FPO pode falhar na convergência são estudadas em [15].

Um algoritmo que teve grande repercussão na solução não-linear de problemas de FPO foi o proposto por Sun et al. [16]. Ele combina, no mesmo algoritmo, o método de Newton (para otimização sem restrições), um método de multiplicadores de Lagrange (para otimização com restrições de igualdades) e funções de penalidade (para o tratamento de restrições de desigualdades). Uma estrutura de dados bem projetada, a qual possibilita a fatorização em blocos, associada ao uso eficiente de técnicas de esparsidade, tornou esse algoritmo bastante atrativo na ocasião. Entretanto, sua eficiência computacional revelou-se altamente dependente da identificação das restrições ativas, um problema que foi posteriormente estudado em [17].

Recentemente, os problemas de otimização em sistemas de potência [18–23], em especial os problemas de FPO, têm sido resolvidos de forma eficiente por métodos de PI, tanto na forma linear quanto na forma não-linear. Na área de programação matemática, nos últimos vinte anos as pesquisas sobre os métodos de PI experimentaram um avanço impressionante, tanto na teoria quanto na prática computacional.

O primeiro método de PI é atribuído a Frisch [24], o qual é um método de barreira logarítmica que foi posteriormente, nos anos 1960, extensivamente estudado por Fiacco e McCormick [25] para resolver problemas com desigualdades não-lineares. No entanto, foi na área de PL, em 1984, que o extraordinário desempenho computacional de um método de PI foi demonstrado na prática [26]. Desde então, vários métodos de PI foram propostos e implementados.

Os primeiros resultados teóricos para os métodos de PI do tipo primal-dual seguidor de trajetória são devidos a Megiddo [27]. Os métodos primal-dual de PI que incorporam passos de predição e correção, tal como o método preditor-corretor de Mehrotra [28], são atualmente aceitos como os métodos de PI computacionalmente mais eficientes. Melhorias adicionais sobre o método preditor-corretor de Mehrotra foram posteriormente alcançadas com o uso de múltiplos passos de correção [29, 30]. Aplicações dos métodos de PI com múltiplos passos de correção em problemas de FPO foram propostas em [31–33]. Atualmente, variantes do método primal-dual de PI estão sendo estudadas para resolver todos os tipos de problemas: de linear a não-linear, e de convexo a não-convexo.

Otimização de sistemas de potência é uma das áreas onde os métodos de PI estão sendo aplicados extensivamente [34]. Isso porque, devido as dimensões e as características especiais de tais problemas, os métodos de PI têm demonstrado, na prática computacional, serem bastante eficientes no que diz respeito ao tempo de processamento e à robustez de convergência. Entre as diversas aplicações dos métodos de PI em sistemas de potência, citam-se: (a) estimação de estados [18,35], (b) modelos diversos de FPO [13,14,19–23,31,36,37], (c) coordenação hidro-térmica [38], (d) colapso de tensão [39], e (e) controle de reservatórios [40].

Nas aplicações supracitadas, os algoritmos de solução incluem diferentes métodos de PI aplicados a sequências de subproblemas de PL [14,36], a sequências de subproblemas de PQ [13], ou diretamente ao problema não-linear original [18–23,37,39]. Diversos métodos de PI foram considerados, tais como o método *dual affine-scaling* [36,41], variantes do

método primal-dual de barreira logarítmica para PNL [18, 20], e variantes tipo preditor-corretor para PL [14] e para PNL [19, 21, 22].

O bom desempenho computacional das aplicações descritas em [18–20], em termos de robustez de convergência e tempo de processamento, foi responsável pelo crescente interesse nos métodos de PI para resolver problemas de otimização não-linear de grande porte em sistemas de potência. Resultados computacionais nas primeiras aplicações em FPO, baseados em redes elétricas de 2423 barras [19] e 3467 barras [20], mostraram que o número de iterações para convergência dos métodos de PI é insensível ao tamanho do sistema elétrico (número de barras e circuitos), e que tais métodos são numericamente robustos.

Como exemplos da robustez dos métodos de PI, cita-se as soluções dos problemas de minimização de corte de carga [23] e de maximização do carregamento [21]. Esses problemas são variantes altamente não-lineares do problema de FPO, que dificilmente são resolvidas por técnicas baseadas em PL. Uma característica interessante do método primal-dual de PI é que a viabilidade é geralmente alcançada durante o processo iterativo, na busca pela otimalidade. Isto significa que as equações de balanço de potência não precisam ser satisfeitas desde o ponto inicial. Essa característica é particularmente interessante na solução do problema de mínimo corte de carga, situação em que a não solvabilidade das equações de fluxo de carga é o problema em questão.

Capítulo 3

Métodos de Pontos-Interiores

Nós, cientistas, acreditamos que o que nós e nossos semelhantes fizermos ou deixarmos de fazer nos próximos anos determinará o destino de nossa civilização. E consideramos nossa tarefa explicar incansavelmente essa verdade, ajudar as pessoas a perceber tudo o que está em jogo, e trabalhar, não para contemporizar, mas para aumentar o entendimento e conseguir, finalmente, a harmonia entre os povos e nações de diferentes pontos de vista.

Albert Einstein

HÁ uma grande variedade de métodos de otimização para a solução de problemas restritos como o FPO formulado em (2.1) (ver [42–44]). No que se refere à resolução de problemas de FPO, ultimamente tem-se utilizado uma classe de métodos chamados

de métodos de Pontos-Interiores (PI). Este capítulo descreve quatro algoritmos de PI de grande sucesso computacional na solução de problemas de FPO: (i) método primal-dual simples, (ii) método primal-dual preditor-corretor, (iii) método preditor-corretor múltiplo, e (iv) método de múltiplas correções de centralidade.

O texto apresentado no presente capítulo segue proximamente o texto de relatórios de pesquisa produzidos no Laboratório de Otimização Aplicada a Sistemas de Potência (LOASP) da Universidade Federal de Pernambuco onde essa Dissertação foi desenvolvida.

3.1 O Método Primal-Dual Simplex

O método primal-dual de PI para resolver o problema (2.1) opera sobre um problema modificado que emerge quando transformamos todas as restrições de desigualdade em restrições de igualdade, adicionando os vetores de variáveis de folga $s \geq 0$ e $z \geq 0$, como segue:

$$\begin{aligned} \text{Minimize} \quad & f(x) \\ \text{sujeito a} \quad & g(x) = 0 \\ & l + s - x = 0, \quad s \geq 0, \\ & x + z - u = 0, \quad z \geq 0. \end{aligned} \tag{3.1}$$

As condições de não-negatividade $s \geq 0$ e $z \geq 0$ em (3.1) são incorporadas em uma função de barreira logarítmica que é agregada à função objetivo, resultando no problema modificado:

$$\begin{aligned} \text{Minimize} \quad & f(x) - \mu_k \sum_{i=1}^p (\ln s_i + \ln z_i) \\ \text{sujeito a} \quad & g(x) = 0, \\ & l + s - x = 0, \quad s > 0, \\ & x + z - u = 0, \quad z > 0, \end{aligned} \tag{3.2}$$

em que $\mu_k > 0$ é o parâmetro de barreira, que é monotonicamente reduzido para zero quando as iterações avançam, ou seja, $\mu_0 > \mu_1 > \dots > \mu_k > \dots > \mu_\infty = 0$.

As condições de positividade estrita $s > 0$ e $z > 0$ devem ser impostas para que os ter-

mos logarítmicos sejam definidos. Entretanto, essas condições são tratadas implicitamente pelo controle do tamanho do passo, conforme é descrito abaixo.

As condições necessárias de otimalidade para o problema modificado (3.2), com o parâmetro de barreira μ_k fixo, podem ser derivadas a partir da função de Lagrange, $L(y; \mu_k)$, associada ao problema (3.2), definida como:

$$L(y; \mu_k) = f(x) - \mu_k \sum_{i=1}^p (\ln s_i + \ln z_i) + \lambda^T g(x) + \pi^T (l + s - x) + \nu^T (x + z - u) \quad (3.3)$$

em que $\lambda \in \mathbb{R}^m$, $\pi \in \mathbb{R}_+^p$ e $\nu \in \mathbb{R}_+^p$ são vetores de multiplicadores de Lagrange, conhecidos como variáveis duais, e $y = (s, z, \pi, \nu, \lambda, x)$ é o vetor com todas as variáveis.

Um mínimo local de (3.2) é caracterizado por um ponto estacionário da função de Lagrange, o qual deve satisfazer as condições necessárias de primeira-ordem de Karush-Kuhn-Tucker (KKT):

$$\nabla_s L = \pi - \mu_k S^{-1} e = 0, \quad (3.4a)$$

$$\nabla_z L = \nu - \mu_k Z^{-1} e = 0, \quad (3.4b)$$

$$\nabla_\pi L = l + s - x = 0, \quad (3.4c)$$

$$\nabla_\nu L = x + z - u = 0, \quad (3.4d)$$

$$\nabla_\lambda L = g(x) = 0, \quad (3.4e)$$

$$\nabla_x L = \nabla f(x) + \nabla g(x) \lambda - \pi + \nu = 0, \quad (3.4f)$$

em que $e = (1, 1, \dots, 1)^T$. O sistema de equações (3.4) é mais convenientemente expresso como:

$$S\pi - \mu_k e = 0, \quad (3.5a)$$

$$Z\nu - \mu_k e = 0, \quad (3.5b)$$

$$l + s - x = 0, \quad (3.5c)$$

$$x + z - u = 0, \quad (3.5d)$$

$$g(x) = 0, \quad (3.5e)$$

$$\nabla f(x) + \nabla g(x) \lambda - \pi + \nu = 0. \quad (3.5f)$$

Uma iteração do método primal-dual de PI para resolver (2.1) invariavelmente aplica apenas uma iteração do método de Newton para achar raízes de equações sobre (3.5), calcula um comprimento de passo na direção de Newton, atualiza as variáveis e reduz o parâmetro de barreira μ_k . O processo iterativo termina quando a inviabilidade primal, a inviabilidade dual e o resíduo de complementaridade estão abaixo de tolerâncias pré-determinadas. Os passos principais do algoritmo primal-dual de PI são como segue:

1. Faça $k = 0$, escolha $\mu_0 > 0$ e um ponto inicial y_0 que satisfaça as condições de estrita positividade $(s_0, z_0, \pi_0, \nu_0) > 0$.
2. Obtenha o sistema de Newton para (3.5) no ponto corrente,

$$\nabla_{yy}^2 L(y_k; \mu_k) \Delta y = -\nabla_y L(y_k; \mu_k),$$

e resolva para a direção de Newton Δy .

3. Calcule o comprimento de passo α_k na direção de Newton Δy e obtenha uma nova estimativa da solução como $y_{k+1} = y_k + \alpha_k \Delta y$.
4. Se y_{k+1} satisfaz o critério de convergência, então FIM. Caso contrário, faça $k \leftarrow k + 1$, reduza o parâmetro μ_k , e retorne para o passo 2.

Algoritmo 3.1: Método primal-dual de pontos-interiores para resolver (2.1).

3.1.1 Cálculo das Direções de Busca

Embora o sistema das condições de KKT (3.5) seja não-linear, sua solução é usualmente aproximada por uma única iteração do método de Newton para achar raízes de equações (a direção de Newton é apenas um meio para seguir o caminho de minimizadores $x(\mu_k)$ parametrizado por μ_k). Quando toma os termos de primeira-ordem na aproximação em série de Taylor do sistema de equações não-lineares (3.5) em torno do ponto y_k , obtem-se

o seguinte sistema linear indefinido:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta v \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S\pi - \mu_k e \\ Zv - \mu_k e \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla f(x) + \nabla g(x)\lambda - \pi + v \end{bmatrix} \quad (3.6)$$

em que $\Pi = \text{diag}(\pi_1, \dots, \pi_p)$, $Y = \text{diag}(v_1, \dots, v_p)$, e $\nabla_{xx}^2 L(y)$ é a Hessiana da função de Lagrange definida a seguir. No sistema acima omitiu-se o contador de iterações k para simplificar a apresentação.

A avaliação da Hessiana da função de Lagrange com relação às variáveis x , $\nabla_{xx}^2 L(y)$, envolve uma combinação da Hessiana da função objetivo $\nabla^2 f(x)$ com as Hessianas das funções de restrições $\nabla^2 g_i(x)$, $i = 1, 2, \dots, m$, da seguinte forma:

$$\nabla_{xx}^2 L(y) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 g_i(x). \quad (3.7)$$

É evidente que o cálculo da Hessiana $\nabla_{xx}^2 L(y)$ pode envolver um esforço computacional significativo se a sua implementação não for bem projetada e executada. O cálculo eficiente dos gradientes e da Hessiana é detalhado no Anexo A.

3.1.2 Atualização das Variáveis

Novas estimativas para as variáveis primais e duais são calculadas por:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k^P \Delta x, \\ s_{k+1} &= s_k + \alpha_k^P \Delta s, \\ z_{k+1} &= z_k + \alpha_k^P \Delta z, \\ \lambda_{k+1} &= \lambda_k + \alpha_k^D \Delta \lambda, \\ \pi_{k+1} &= \pi_k + \alpha_k^D \Delta \pi, \\ v_{k+1} &= v_k + \alpha_k^D \Delta v, \end{aligned} \quad (3.8)$$

em que $\alpha_k^P \in (0, 1]$ e $\alpha_k^D \in (0, 1]$ são os comprimentos de passos primal e dual, respectivamente.

Os comprimentos de passos máximos que podem ser tomados na direção de Newton são geralmente determinados pelos testes:

$$\alpha_k^P = \min \left\{ 1, \gamma \min_i \left\{ \frac{-s_i^k}{\Delta s_i} \left| \Delta s_i < 0, \frac{-z_i^k}{\Delta z_i} \left| \Delta z_i < 0 \right. \right\} \right\}, \quad (3.9a)$$

$$\alpha_k^D = \min \left\{ 1, \gamma \min_i \left\{ \frac{-\pi_i^k}{\Delta \pi_i} \left| \Delta \pi_i < 0, \frac{-v_i^k}{\Delta v_i} \left| \Delta v_i < 0 \right. \right\} \right\}. \quad (3.9b)$$

O escalar $\gamma \in (0, 1)$ é um fator de segurança para assegurar que o próximo ponto satisfará as condições de estrita positividade; um valor típico é $\gamma = 0.99995$.

O uso de comprimentos de passos distintos nos espaços primal e dual é uma vantagem do método primal-dual de pontos interiores, e tem provado na prática ser altamente eficaz, reduzindo o número de iterações para convergência entre 10% a 20% em um problema típico. Entretanto, em programação não-linear, a interdependência de variáveis primais e duais, presente na condição de viabilidade dual (3.5e), não permite rigorosamente o uso de comprimentos de passos distintos nos espaços primal e dual. Assim, um comprimento de passo comum para atualizar as variáveis primais e duais deve ser calculado por:

$$\alpha_k^P = \alpha_k^D \leftarrow \min \{ \alpha_k^P, \alpha_k^D \}. \quad (3.10)$$

Embora haja o acoplamento mencionado entre as variáveis primais e duais, na prática, o uso de um comprimento de passo comum ou o uso de comprimentos de passos distintos têm ambos desempenhado bem.

3.1.3 Redução do Parâmetro de Barreira

O esquema para reduzir μ_k que é descrito aqui é uma extensão de esquemas que são utilizados com sucesso em programação linear e programação quadrática [45]. Na k -ésima iteração, o resíduo das condições de complementaridade, chamado de resíduo de complementaridade, é obtido por:

$$\rho_k = s_k^T \pi_k + z_k^T v_k. \quad (3.11)$$

Se as iterações convergem para uma solução ótima, então $\rho_k \rightarrow 0$. O relacionamento entre ρ_k e μ_k , que está implícito em (3.5a) e (3.11), na forma:

$$\sum_{i=1}^p s_i \pi_i + \sum_{i=1}^p z_i v_i = 2p\mu_k = \rho \quad (3.12)$$

sugere que μ_k pode ser reduzido em função do decréscimo do resíduo de complementaridade, tal como:

$$\mu_{k+1} = \sigma \frac{\rho_k}{2p} \quad (3.13)$$

em que $\sigma = 0.2$ é o esperado, mas não necessariamente realizado, decréscimo no resíduo de complementaridade. O parâmetro $\sigma \in (0, 1)$ é chamado de parâmetro de centralização. Se $\sigma = 1$, então o sistema de KKT (3.5) define uma direção de centralização, um passo em direção a um ponto na trajetória de barreira. No outro extremo, $\sigma = 0$ fornece o passo de Newton puro, conhecido como a direção *affine-scaling*. Para balancear os objetivos de reduzir μ_k e melhorar a centralidade, σ pode ser escolhido no intervalo $(0, 1)$.

3.1.4 Testes de Convergência

As iterações do algoritmo PDPI são consideradas terminadas assim que:

$$\max \left\{ \max_i \{l_i - x_i^k\}, \max_i \{x_i^k - u_i\}, \|g(x_k)\|_\infty \right\} \leq \epsilon_1, \quad (3.14a)$$

$$\frac{\|\nabla f(x_k) + \nabla g(x_k)\lambda - \pi_k + v_k\|_\infty}{1 + \|x_k\|_2 + \|\lambda_k\|_2 + \|\pi_k\|_2 + \|v_k\|_2} \leq \epsilon_1, \quad (3.14b)$$

$$\frac{\rho_k}{1 + \|x_k\|_2} \leq \epsilon_2. \quad (3.14c)$$

Se os testes (3.14a), (3.14b) e (3.14c) são satisfeitos, então a viabilidade primal, a viabilidade dual e as condições de complementaridade são satisfeitas, significando que x_k é um ponto de KKT de precisão ϵ_1 . Tolerâncias de convergência típicas são $\epsilon_1 = 10^{-4}$ e $\epsilon_2 = 10^{-2}\epsilon_1$.

3.2 O Método Preditor-Corretor

Para obter o algoritmo preditor-corretor de Mehrotra, em vez de aplicar o método de Newton a (3.5) para calcular a correção Δy para y_k , deve-se substituir o novo ponto $y_{k+1} = y_k + \Delta y$ diretamente em (3.5), para obter a aproximação:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta v \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S\pi \\ Zv \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla_x L(y) \end{bmatrix} + \begin{bmatrix} \mu_k e \\ \mu_k e \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \Delta S \Delta \pi \\ \Delta Z \Delta v \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.15)$$

em que

$$\begin{aligned} \nabla_x L(y) &= \nabla f(x) + \nabla g(x)\lambda - \pi + v, \\ \Delta S &= \text{diag}(\Delta s_1, \dots, \Delta s_p), \\ \Delta Z &= \text{diag}(\Delta z_1, \dots, \Delta z_p). \end{aligned}$$

A principal diferença entre os sistemas de equações (3.15) e (3.6) é que o vetor do lado direito de (3.15) não pode ser determinado de antemão por causa dos Δ -termos não-lineares $\Delta S \Delta \pi$ e $\Delta Z \Delta v$ em termos das incógnitas. A direção Δy que é obtida de (3.15) consiste de três componentes:

$$\Delta y = \Delta y_{\text{af}} + \Delta y_{\text{ce}} + \Delta y_{\text{co}}, \quad (3.16)$$

em que cada componente é determinada por um dos três vetores no lado direito de (3.15). As três componentes de direções podem ser interpretadas como segue:

Δy_{af} : É a direção *affine-scaling*, a direção pura de Newton que é obtida quando se faz $\mu_k = 0$ em (3.6). É determinada pelo primeiro vetor no lado direito de (3.15), ou

seja, como a solução do sistema:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{af} \\ \Delta z_{af} \\ \Delta \pi_{af} \\ \Delta v_{af} \\ \Delta \lambda_{af} \\ \Delta x_{af} \end{bmatrix} = - \begin{bmatrix} S\pi \\ Zv \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla_x L(y) \end{bmatrix} \quad (3.17)$$

e é responsável pela otimização, ou seja, por reduzir as inviabilidades primal e dual e o resíduo de complementaridade.

Δy_{ce} : É a direção de *centralização*, cujo tamanho é governado pelo parâmetro μ_k que é escolhido adaptativamente. A direção Δy_{ce} é determinada pelo segundo vetor no lado direito de (3.15), ou seja, como a solução do sistema:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{ce} \\ \Delta z_{ce} \\ \Delta \pi_{ce} \\ \Delta v_{ce} \\ \Delta \lambda_{ce} \\ \Delta x_{ce} \end{bmatrix} = \begin{bmatrix} \mu_k e \\ \mu_k e \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.18)$$

e mantém o ponto corrente afastado da fronteira da região viável e idealmente próximo da trajetória de barreira para melhorar as chances de tomar um passo grande na próxima iteração.

Δy_{co} : É a direção de *correção* que tenta compensar algumas das não-linearidades na direção Δy_{af} . A direção Δy_{co} é determinada pelo terceiro vetor no lado direito

de (3.15), ou seja, como a solução do sistema:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{co} \\ \Delta z_{co} \\ \Delta \pi_{co} \\ \Delta v_{co} \\ \Delta \lambda_{co} \\ \Delta x_{co} \end{bmatrix} = - \begin{bmatrix} \Delta S \Delta \pi \\ \Delta Z \Delta v \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.19)$$

Claramente, as direções Δy_{af} e Δy_{ce} , quando combinadas, definem a direção de Newton que é calculada de (3.6). Entretanto, para lidar com as não-linearidades em (3.15), a direção Δy_{af} é calculada separadamente e antes da direção Δy_{ce} . Esse arranjo no cálculo de Δy nos provê a possibilidade de escolhermos μ_{k+1} adaptativamente, e com um meio de aproximarmos os termos de segunda-ordem $\Delta S \Delta \pi$ e $\Delta Z \Delta v$.

3.2.1 O Passo Preditor

Para determinar um passo que aproximadamente satisfaz (3.15), primeiro retira-se os μ_k -termos e os Δ -termos no lado direito de (3.15) e resolve-se para a direção *affine-scaling*:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{af} \\ \Delta z_{af} \\ \Delta \pi_{af} \\ \Delta v_{af} \\ \Delta \lambda_{af} \\ \Delta x_{af} \end{bmatrix} = - \begin{bmatrix} S \pi \\ Z v \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla f(x) + \nabla g(x) \lambda - \pi + v \end{bmatrix}. \quad (3.20)$$

A direção Δy_{af} é utilizada em duas formas distintas: (a) para aproximar os Δ -termos no lado direito de (3.15) e (b) para dinamicamente estimar o parâmetro de barreira μ_{k+1} .

Para estimar μ_{k+1} , primeiro considera-se a regra padrão (3.9) para determinar o passo

α_{af} que seria dado se a direção Δy_{af} fosse, de fato, utilizada:

$$\alpha_{af}^P = \min \left\{ 1, \gamma \times \min_i \left\{ \frac{-s_i^k}{\Delta s_i^{af}} \left| \Delta s_i^{af} < 0, \frac{-z_i^k}{\Delta z_i^{af}} \left| \Delta z_i^{af} < 0 \right. \right\} \right\}, \quad (3.21a)$$

$$\alpha_{af}^D = \min \left\{ 1, \gamma \times \min_i \left\{ \frac{-\pi_i^k}{\Delta \pi_i^{af}} \left| \Delta \pi_i^{af} < 0, \frac{-v_i^k}{\Delta v_i^{af}} \left| \Delta v_i^{af} < 0 \right. \right\} \right\}. \quad (3.21b)$$

Segundo, calcula-se uma estimativa do resíduo de complementaridade por:

$$\rho_{af} = (s_k + \alpha_{af}^P \Delta s_{af})^T (\pi_k + \alpha_{af}^D \Delta \pi_{af}) + (z_k + \alpha_{af}^P \Delta z_{af})^T (v_k + \alpha_{af}^D \Delta v_{af}). \quad (3.22)$$

Finalmente, obtem-se uma estimativa μ_{af} para μ_{k+1} de:

$$\mu_{af} = \min \left\{ \left(\frac{\rho_{af}}{\rho_k} \right)^2, 0.2 \right\} \frac{\rho_{af}}{p}. \quad (3.23)$$

Esse procedimento escolhe μ_{af} pequeno se a direção Δy_{af} produzir um decréscimo grande no resíduo de complementaridade, ou seja, se $\rho_{af} \ll \rho_k$, e escolhe μ_{af} grande caso contrário.

3.2.2 O Passo Corretor

Em vez de calcular a direção composta $\Delta y_{ce} + \Delta y_{co}$ para adicionar a Δy_{af} e então obter Δy , calcula-se a direção Δy de uma só vez de:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta v \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S \pi - \mu_{af} e + \Delta S_{af} \Delta \pi_{af} \\ Z v - \mu_{af} e + \Delta Z_{af} \Delta v_{af} \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla f(x) + \nabla g(x) \lambda - \pi + v \end{bmatrix}. \quad (3.24)$$

Uma vez que os passos preditor e corretor são baseados sobre a mesma fatorização de matriz (ver as matrizes de coeficientes em (3.17) e (3.24)), o esforço adicional no método preditor-corretor está na solução do sistema linear extra para calcular a direção Δy_{af} , e

no teste extra utilizado para calcular μ_{af} . Os benefícios que geralmente são obtidos desse esforço extra são reduções no contador de iterações e no tempo total de solução. Os passos principais do algoritmo preditor-corretor de pontos interiores são como segue:

1. Faça $k = 0$, escolha $\mu_0 > 0$ e um ponto inicial y_0 que satisfaça as condições de estrita positividade $(s_0, z_0, \pi_0, v_0) > 0$.
2. Forme a matriz $\nabla_{yy}^2 L(y_k; \mu_k)$ e obtenha a sua fatorização.
 - (a) Resolva o sistema (3.17) para a direção Δy_{af} , calcule α_{af} de (3.21), e obtenha μ_{af} de (3.22).
 - (b) Resolva o sistema (3.24) para a direção Δy .
3. Calcule o comprimento de passo α_k na direção Δy e obtenha um novo ponto como $y_{k+1} = y_k + \alpha_k \Delta y$.
4. Se y_{k+1} satisfaz o critério de convergência, então FIM. Caso contrário, faça $k \leftarrow k + 1$, reduza o parâmetro μ_k , e retorne para o Passo 2.

Algoritmo 3.2: Método preditor-corretor de pontos interiores para resolver (2.1).

3.3 O Método Preditor-Corretor Múltiplo

O método preditor-corretor múltiplo foi originalmente desenvolvido para PL por Carpenter et al. [29], e também calcula Δy_{af} e μ_{af} por meio das equações (3.20) e (3.23). No entanto, diferentemente do método preditor-corretor simples, o método preditor-corretor múltiplo executa múltiplos passos de correção. Definindo $\Delta y^0 = \Delta y_{af}$, o l -ésimo passo de correção é calculado por

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta v \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S \pi - \mu_{af} e + \Delta S^l \Delta \pi^l \\ Z v - \mu_{af} e + \Delta Z^l \Delta v^l \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla f(x) + \nabla g(x) \lambda - \pi + v \end{bmatrix}. \quad (3.25)$$

Após a última correção ser executada, faz-se $\Delta y = \Delta y^l$ e obtém-se um novo ponto calculando o comprimento de passo e atualizando as variáveis da mesma forma que nos métodos primal-dual simples e preditor-corretor de Mehrotra.

O número de correções l por iteração pode variar. Se $l = 1$ para todo k , então o método preditor-corretor múltiplo comporta-se exatamente como o método preditor-corretor de Mehrotra. Definir quantos passos de correção por iteração é vantajoso é uma questão difícil, a qual Carpenter et al. [29] abordaram considerando dois casos distintos porém complementares: caso viável e caso inviável.

No caso viável, a única motivação para executar correções a partir de um ponto viável é reduzir o resíduo de complementaridade. Neste caso, considera-se a execução da correção $(l + 1)$ apenas se $\rho^l < \rho^{l-1}$ e l é menor do que um número máximo pré-determinado de correções. Se $\rho^l \geq \rho^{l-1}$, então as correções são interrompidas e utiliza-se a direção de busca $\Delta y = \Delta y^{l-1}$.

No caso inviável, a determinação do l máximo deve integrar redução do resíduo de complementaridade com redução de inviabilidade. Neste caso, a correção $(l + 1)$ é tentada apenas se l é menor do que o número máximo permitido M e $G^l < G^{l-1}$, em que G é a norma do resíduo das equações KKT, ou seja, $G = \|\nabla_y L(y)\|$.

3.4 O Método de Múltiplas Correções de Centralidade

Gondzio observou [30], a partir de exaustivos experimentos computacionais, que o desempenho de um algoritmo primal-dual de pontos-interiores é significativamente degradado se há enormes discrepâncias entre os produtos de complementaridade, ou seja, se $s_i \pi_i \lll s_j \pi_j$ para alguns índices i e j . Os termos $s_i \pi_i$ e $z_i v_i$ que são muito pequenos ou muito grandes, quando comparados com a média $\mu_{ave} = \rho/2n$, são indesejáveis (com os primeiros geralmente sendo mais danosos). Se y_k é mal centralizado, ou seja, se $s_i \pi_i \lll s_j \pi_j$ para alguns índices i e j , então o lado direito de (3.6) é muito mal escalonado. A direção de Newton concentra-se na redução dos produtos grandes, mas devido aos produtos pequenos apenas passos curtos são permitidos, o que desacelera o processo de convergência.

3.4.1 As Correções de Centralidade

Para definir as correções de centralidade, assuma que o passo preditor Δy_{af} foi calculado, assim como o comprimento de passo α_{af} que seria dado se Δy_{af} fosse de fato utilizado. Assim, busca-se por um termo de correção Δy_{co} de forma tal que um passo $\tilde{\alpha} > \alpha_{af}$, dado por

$$\tilde{\alpha} = \min \{ \alpha_{af} + \delta_{\alpha}, 1 \}, \quad (3.26)$$

e uma direção de busca composta Δy , dada por

$$\Delta y = \Delta y_{af} + \Delta y_{co}, \quad (3.27)$$

podem ser aplicados sem que o ponto tentativa $y_k + \tilde{\alpha}\Delta y$ viole a condição de estrita positividade $(s, z, \pi, \nu) > 0$. Para tornar isto possível, alguns requisitos devem ser impostos sobre Δy_{co} . Claramente, sempre que $\alpha_{af} < 1$, o ponto tentativa

$$\tilde{y} = y_k + \tilde{\alpha}\Delta y_{af} \quad (3.28)$$

pode ter componentes que violam $(s, z, \pi, \nu) > 0$. Neste caso, o termo corretor Δy_{co} deve compensar aquelas componentes negativas e ainda conduzir o ponto tentativa \tilde{y} de volta para a vizinhança da *trajetória de barreira*.

Sejam $q = S\pi$ e $r = Z\nu$ vetores de produtos de complementaridade. Dados Δy_{af} e α_{af} do passo preditor, e um pequeno incremento de comprimento de passo $\delta_{\alpha} > 0$, primeiro obtenha o passo aumentado $\tilde{\alpha}$ por meio de (3.26) e calcule o ponto tentativa \tilde{y} por meio de (3.28). O ponto tentativa fornece

$$\tilde{q} = \tilde{S}\tilde{\pi}, \quad (3.29a)$$

$$\tilde{r} = \tilde{Z}\tilde{\nu}. \quad (3.29b)$$

A seguir, identifique os componentes de \tilde{q} e \tilde{r} que não pertencem ao intervalo $(\underline{\beta}\mu_{af}, \bar{\beta}\mu_{af})$, chamados de *violadores*, em que $\underline{\beta}$ e $\bar{\beta}$ são valores relativos fornecidos. O esforço nos passos de correção é concentrado apenas sobre a correção desses *violadores* para assim melhorar a centralidade de y_{k+1} . Para fazer isto, primeiro projeta-se \tilde{q} e \tilde{r} componente a

componente sobre o hipercubo $H = [\underline{\beta}\mu_{af}, \bar{\beta}\mu_{af}]^{2n}$, para definir um *alvo*, definido como

$$q_i^t = \begin{cases} \underline{\beta}\mu_{af}, & \text{se } \tilde{q}_i < \underline{\beta}\mu_{af}, \\ \bar{\beta}\mu_{af}, & \text{se } \tilde{q}_i > \bar{\beta}\mu_{af}, \\ \tilde{q}_i, & \text{caso contrário.} \end{cases} \quad (3.30a)$$

$$r_i^t = \begin{cases} \underline{\beta}\mu_{af}, & \text{se } \tilde{r}_i < \underline{\beta}\mu_{af}, \\ \bar{\beta}\mu_{af}, & \text{se } \tilde{r}_i > \bar{\beta}\mu_{af}, \\ \tilde{r}_i, & \text{caso contrário.} \end{cases} \quad (3.30b)$$

Então, uma correção de centralidade Δy_{co} é calculada por

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{co} \\ \Delta z_{co} \\ \Delta \pi_{co} \\ \Delta v_{co} \\ \Delta \lambda_{co} \\ \Delta x_{co} \end{bmatrix} = \begin{bmatrix} q_t - \tilde{q} \\ r_t - \tilde{r} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.31)$$

O lado direito da equação (3.31) tem elementos não nulos apenas em um subconjunto das posições de $q_t - \tilde{q}$ e $r_t - \tilde{r}$ que correspondem aos violadores, ou seja, as componentes de \tilde{q} e \tilde{r} que não pertencem a $(\underline{\beta}\mu_{af}, \bar{\beta}\mu_{af})$. Entretanto, esse vetor ainda pode ser mal escalonado se existirem componentes muito grandes em \tilde{q} e \tilde{r} . Assim, para prevenir efeitos indesejáveis deste escalonamento ruim, todos os componentes de \tilde{q} e \tilde{r} que são menores do que $-\bar{\beta}\mu_{af}$ são substituídos por este valor, o que corresponde a limitar o decaimento esperado dos componentes de \tilde{q} e \tilde{r} que são muito grandes.

O termo de correção de centralidade Δy_{co} que resolve (3.31) é então utilizado para corrigir a direção prevista:

$$\Delta y = \Delta y_{af} + \Delta y_{co}. \quad (3.32)$$

Em seguida, um passo α_k na direção Δy é determinado e uma nova estimativa da solução y_{k+1} é calculada, como no método primal-dual simples. O processo de correção (3.32) pode ser repetido um número desejado de vezes. Para aplicar uma nova correção, faz-se $\Delta y_{af} \leftarrow \Delta y$ e $\alpha_{af} \leftarrow \alpha_k$, ou seja, a direção Δy em (3.32) torna-se a direção prevista para o

novos corretor.

3.4.2 Quantas Correções Realizar?

Múltiplas correções de centralização são de interesse prático apenas se a redução no contador de iterações traduz-se em redução no tempo de processamento total. Seguindo propostas em [30], o processo de correção termina quando o tamanho do passo determinado para a direção Δy em (3.32) não aumenta suficientemente quando comparado ao tamanho de passo encontrado anteriormente para Δy_{af} , ou seja, se

$$\alpha_k < \alpha_{af} + \gamma_\alpha \delta_\alpha, \quad (3.33)$$

em que γ_α é um percentual de aumento de passo mínimo aceitável. Além da condição (3.33), é necessário limitar o número de correções de centralidade por iteração. O algoritmo de Gondzio calcula a razão entre o esforço da fatoração da matriz de coeficientes e o esforço da solução do sistema linear, $r_{f/s}$, e permite uma correção de centralidade se $r_{f/s} > 10$, duas correções se $r_{f/s} > 30$, e três correções se $r_{f/s} > 50$. Em programação não-linear, este problema é bem mais complexo. É sugerido em [46] que a razão $r_{f/s}$ seja obtida utilizando o tempo real de CPU para essas operações como verificado no passo preditor da primeira iteração do algoritmo. É sugerido também que δ_α seja dinamicamente escolhido. Seja K o número máximo definido de correções de centralidade por iteração. Como o passo inicial α_{af} é conhecido, e idealmente o passo deve ser próximo da unidade, então define-se $\delta_\alpha = (1 - \alpha_{af})/M$. Adicionalmente, considera-se que δ_α não deve ser menor do que 0,1 (pessimismo demais) e nem maior do que 0,2 (muito otimismo).

Capítulo 4

Estudo dos Requisitos para Utilização Prática de Programas de FPO

*Nossa maior fraqueza é a desistência. O
caminho certo para o sucesso é sempre tentar
apenas uma vez mais.*

Thomas A. Edison

HÁ VÁRIAS DIFICULDADES em aplicações práticas de programas de FPO, principalmente em tempo-real. A principal delas é que o FPO no mundo real é matematicamente e computacionalmente muito diferente da formulação clássica [12]. Por exemplo, os problemas de FPO, na prática, não são contínuos e suaves; muitos dos controles movem-se em

passos discretos (como os tapes dos transformadores, as susceptâncias em paralelo, etc.), e esses passos algumas vezes são muito grandes ou irregulares (vários tamanhos de passo para uma mesma variável de controle).

Infelizmente, não há uma técnica de solução que possa tratar esses tipos de problemas eficientemente com todo rigor na modelagem. A prática universal tem sido aproximar a formulação de tal forma que o problema possa ser resolvido pelas técnicas de otimização disponíveis. Numa aplicação em tempo-real, o modelo do FPO e a técnica de solução do modelo são fortemente influenciados pela urgência em implementar o controle calculado, de forma que rapidez de processamento e confiabilidade são requisitos indispensáveis.

Em muitas situações práticas é difícil, indesejável, ou impossível modificar um grande número de controles do sistema todos ao mesmo tempo. Portanto, na formulação de problemas de FPO, é importante levar em conta procedimentos normais de engenharia que norteiam a operação de um sistema real. Por exemplo, o procedimento para controle de tensão em tempo-real que é apresentado em [3] considera uma sequência prioritária, pré-estabelecida, para os objetivos do controle, dependendo do nível de segurança desejado para o sistema.

O objetivo de maior prioridade é remover as violações de limites de tensão. A segunda maior prioridade é minimizar aquelas violações que não podem ser eliminadas por completo. O terceiro objetivo é minimizar as perdas ativas na transmissão. Em tal aplicação, o estado do sistema e a estrutura da rede são derivados do estimador de estados. As variáveis de controle são selecionadas e seus movimentos são limitados, pelas seguintes razões:

- evitar grandes perturbações do sistema de potência do seu estado de equilíbrio corrente;
- prover “margem de manobra” para ajustes corretivos em situações de contingências;
- obter soluções rápidas, um requisito para aplicações em tempo-real.

É evidente que restringir o movimento das variáveis de controle pode levar a soluções sub-ótimas em cada execução. Entretanto, devido o programa ser executado periodicamente, como parte da sequência de monitoramentos em tempo-real, espera-se que com as sucessivas execuções do programa o estado do sistema mova em direção ao ótimo.

Ainda com relação ao procedimento para controle de tensão que é proposto em [3], se violações de tensões são detectadas na parte monitorada do sistema, então o cálculo de ajustes corretivos é iniciado para determinar um conjunto de ações de controle para aliviar as violações. Caso esse cálculo apresente-se inviável (não há um conjunto de controles que corrijam as violações), o processo de controle é automaticamente direcionado para calcular os ajustes de controles que minimizam as violações de restrições. Quando nenhuma violação de tensão é detectada, então os controles de potência reativa são despachados para minimizar as perdas de potência ativa na transmissão.

4.1 Otimização por Áreas

Em se tentando otimizar somente uma porção de um sistema interligado, há um problema fundamental de quanto se pode afetar a região vizinha. Uma solução aproximada para esse problema é fazer com que o sistema interno não altere os fluxos ou as magnitudes das tensões na região de fronteira com o sistema externo. Ou seja, o sistema externo não é modelado. Os fluxos das linhas de interconexão entre as áreas são tratadas como cargas fixas e as tensões das barras de fronteira são restritas a valores constantes.

Essa aproximação é extremamente conservadora, e a menos que o sistema interno seja grande e com alto grau de liberdade, ela pode sobrelimitar o processo de otimização. Uma aproximação razoável e menos restritiva é modelar as barras de fronteiras como barras de geração com a geração de potência ativa fixa. Faixas limites são impostas nas tensões e na geração potência reativa dessas fontes fictícias. A dificuldade dessa aproximação reside em se predefinir realisticamente os valores dessas faixas. Até então, uma aproximação mais comum e mais fácil tem sido fornecer detalhes da modelagem do fluxo de potência da rede externa a uma zona piloto localizada em torno da parte do sistema que se deseja otimizar.

Em alguns sistemas elétricos a definição de áreas de otimização já existe naturalmente. Por exemplo, o sistema de subtransmissão da CELPE (Companhia Energética de Pernambuco) encontra-se naturalmente dividido em 13 subsistemas não-interligados, denominados de regionais, tendo como barras de geração apenas os pontos de conexão com o sistema de transmissão da CHESF. Essa topologia do sistema de subtransmissão da CELPE permite naturalmente a otimização da operação por regionais, favorecendo assim a definição natural de áreas de otimização. Tal característica favorece o processo de otimização. Tanto

é possível que os 13 regionais sejam otimizados simultaneamente, quanto é possível a otimização de cada regional individualmente, diminuindo as dimensões dos problemas de otimização a serem resolvidos.

4.2 Restrições de Contingências

Restrições de contingências são fundamentais no controle econômico e seguro. Elas são intrínsecas aos níveis de segurança 1 e 2 [1]. Entretanto, poucos trabalhos desenvolvidos sobre FPO tem considerado efetivamente restrições de contingências. O número total de restrições de contingências que podem ser impostas sobre a solução do FPO é enorme. Dependendo do tamanho do sistema de potência, cada contingência pode envolver centenas ou milhares de restrições. Uma vez que uma lista típica de contingências é extensa, o número total de restrições pode ser da ordem de milhões. Felizmente, apenas uma pequena parte delas estará ativa na solução. Procedimentos para a consideração de restrições de contingências levando em consideração os níveis de segurança 1 e 2 são descritos em [1].

Para se ter uma idéia da complexidade que é a obtenção de soluções restritas por contingências, considere a forma geral do problema de FPO em (2.1), o qual é rescrito abaixo particionando-se o vetor de variáveis x nos vetores de variáveis de controle x_c , e de variáveis de estado x_e , como segue:

$$\text{Minimize } f(x_c, x_e) \quad (4.1a)$$

$$\text{sujeito a } g(x_c, x_e) = 0 \quad (4.1b)$$

$$l_c \leq x_c \leq u_c \quad (4.1c)$$

$$l_e \leq x_e \leq u_e \quad (4.1d)$$

Considerando a inclusão de duas contingências apenas, o problema (4.1) passa a ser escrito

como

$$\text{Minimize } f(x_c, x_e) \quad (4.2a)$$

$$\text{sujeito a } g(x_c, x_e) = 0 \quad (4.2b)$$

$$g'(x_c, x_e') = 0 \quad (4.2c)$$

$$g''(x_c, x_e'') = 0 \quad (4.2d)$$

$$l_c \leq x_c \leq u_c \quad (4.2e)$$

$$l_e \leq x_e \leq u_e \quad (4.2f)$$

$$l_e \leq x_e' \leq u_e \quad (4.2g)$$

$$l_e \leq x_e'' \leq u_e \quad (4.2h)$$

Uma vez que as contingências ocasionam alterações na configuração da rede elétrica, as equações que modelam a rede nas configurações pós-contingências, $g'(x_c, x_e') = 0$ e $g''(x_c, x_e'') = 0$, diferem das equações no caso base, $g(x_c, x_e) = 0$. O que há de comum entre estas restrições são as variáveis de controle x_c .

A solução restrita por contingências consiste então em determinar o conjunto de variáveis de controle que minimiza a função objetivo no caso base e satisfaz as restrições no caso base e nas configurações de rede pós-contingências.

O exemplo acima demonstra que o número de contingências consideradas na solução praticamente multiplica por igual número o número de restrições de igualdade do caso base e o número de restrições de limites simples sobre as variáveis de estado.

4.3 Objetivos Operacionais

Mesmo que o objetivo primário da otimização da operação de um sistema elétrico seja, por exemplo, minimizar perdas ativas na transmissão, no contexto da aplicação da ferramenta de FPO em tempo-real a necessidade de implementar várias outras funções objetivo surge naturalmente, conforme as discussões e análises que seguem.

Minimização de Perdas na Transmissão: Os controles para atingir este objetivo são aqueles sem custos diretos, ou seja, todos exceto gerações de potência ativa e intercâmbios.

Como as perdas ativas em geral estão associadas com controle de tensão e de potência reativa, este é dito um objetivo de ajuste fino da operação. Ele tende a reduzir a circulação de potência reativa, proporcionando um perfil de tensão mais plano.

As perdas podem ser minimizadas somente numa parte do sistema ou no sistema completo. No primeiro caso, a solução algumas vezes diminui as perdas internas ao custo de um aumento das perdas em áreas vizinhas. Um objetivo similar é minimizar perdas potência reativa série. Este objetivo tem vários efeitos benéficos [1], tais como: quase minimiza as perdas ativas, mantém o perfil de tensões plano, e quase maximiza as reservas de potência reativa dos geradores.

Minimização de Desvio do Ponto de Operação: Este objetivo é geralmente definido como a soma dos quadrados ponderados dos desvios das variáveis de controle dos valores alvos especificados, como exemplificado a seguir:

$$\text{Minimize} \quad \sum_{i \in \mathcal{G}} \omega_v (V_i - V_i^0)^2 + \sum_{(i,j) \in \mathcal{T}} \omega_t (t_{ij} - t_{ij}^0)^2 + \sum_{i \in \mathcal{C}} \omega_b (b_i^{sh} - b_i^{sh0})^2 \quad (4.3)$$

As restrições para a função objetivo (4.3) são as mesmas do problema de minimização de perdas ativas (2.38). O primeiro termo da função objetivo (4.3) minimiza os desvios quadrados nas tensões terminais dos geradores, o segundo termo minimiza os desvios quadrados nos tapes, e o terceiro termo minimiza os desvios quadrados nas susceptâncias em paralelo. Os coeficientes ω_v , ω_t e ω_b representam ponderações (pesos relativos) da importância de um controle em relação a outro. Quanto maior for um peso relativo menos desejável será um desvio naquela variável de controle.

A função objetivo de mínimo desvio do ponto de operação pode ser utilizada para planejamento corretivo. Ela também oferece uma forma prática de interfacear uma função de planejamento com uma outra hierarquicamente superior. Por exemplo, um cálculo no modo estudo pode definir um controle desejável de tensão e potência reativa, enquanto o cálculo em tempo-real pode manter a operação livre de violações de restrições operativas com um mínimo de desvio do valor ótimo planejado.

Conforme análise apresentada em [1], este objetivo deve ser utilizado com bastante cautela, pois é composto pela soma de quantidades não similares (por exemplo, desvios de tensões, de tapes e de gerações de potência ativa), ou seja, ele é composto ou multiobjetivo.

Não há um critério geral para escolha automática dos pesos relativos. A escolha dos pesos relativos parece depender dos tipos e dos locais das violações de limites, assim como do carregamento e da configuração do sistema.

Em função das características do sistema de subtransmissão CELPE e dos recursos de controle da operação disponíveis (essencialmente o controle de tapes e de compensação de reativo por bancos de capacitores em paralelo), a função mínimo desvio é composta basicamente dos desvios quadrados dos tapes e das susceptâncias em paralelo.

Minimização do Número de Controles Ajustados: Algumas vezes é impraticável ou indesejável reajustar um grande número de controles ao mesmo tempo. Este objetivo aplica-se quando os meios são limitados para a implementação de muitos controles simultaneamente, ou a operação frequente de equipamentos é não recomendada.

Duas questões importantes, também relacionadas aos objetivos operacionais, são tratadas a seguir: a supressão de ajustes ineficazes e o tratamento de inviabilidades na solução do problema de FPO.

4.4 Supressão de Ajustes Ineficazes

Para aplicações do FPO em tempo-real, uma restrição especial sobre qualquer objetivo da otimização é a supressão dos ajustes pouco efetivos. A supressão de determinados ajustes de controles decorre da necessidade de prevenir o movimento daqueles controles que, se ajustados, teriam pouco efeito sobre o objetivo ou sobre o atendimento de limites operacionais. Esta restrição, a qual é bastante difícil de se especificar analiticamente, está relacionada com, mas sem ser idêntica, o objetivo de minimização do número de controles.

Segundo [4], há pouca esperança de se desenvolver uma técnica rápida para solução do problema de minimização do número de controles, e o objetivo deve ser desenvolver um bom método de aproximação da solução ótima que seja rápido suficiente para uso prático. Nesse sentido é melhor obter soluções aproximadamente ótimas para problemas realistas do que soluções rigorosamente ótimas para problemas irrealis. A restrição do número de controles ajustados pode ser imposta heurísticamente de diversas formas. Uma delas consiste em primeiro resolver o problema de FPO completo, e então iteragir, sucessivamente

inibindo o ajuste de controles pouco eficazes.

Diversos controles cujas ações são limitadas em número são ajustáveis apenas em passos discretos, e métodos para lidar com variáveis discretas tendem a diminuir o número de movimentos de controle, mantendo os controles discretos de menor importância nos seus valores iniciais [4]. A dificuldade no tratamento desse requisito decorre do fato das ações de controle não poderem ser ordenadas de maneira simples, porque a importância de uma ação de controle não está necessariamente relacionada à sua magnitude. Decorre ainda de que cada controle participa na minimização da função objetivo e no atendimento das restrições, e assim não há uma maneira simples de separar esses dois efeitos.

E seguindo a lógica descrita acima, recentemente foi proposta em [8] uma metodologia para redução do número de ações de controle, com as seguintes características:

- Resolvem-se três problemas de FPO com estruturas idênticas, num processo de solução com três fases, ditas Fase I, Fase II e Fase III.
- A solução de FPO calculada na Fase I emprega todos os controles e é utilizada para definir o problema de FPO da Fase II.
- Os multiplicadores de Lagrange obtidos na solução do FPO da Fase II são usados para identificar as ações de controle mais efetivas. O número máximo de controles permitido deve ser especificado pelo usuário com base em práticas operacionais.
- A solução de FPO calculada na Fase III reajusta as magnitudes dos controles selecionados como efetivos para compensar a ação dos controles que deixam de ser modificados.
- Os problemas de FPO nas três fases tem a mesma estrutura e assim podem ser resolvidos pelo mesmo algoritmo de otimização.

Pequenos ajustes geralmente estão relacionados com as variáveis de controle que são contínuas, tais como tensões terminais de geradores. Como o sistema de subtransmissão da CELPE está dividido em 13 regionais, cada subsistema apresenta uma única barra de tensão controlada: a barra de conexão com o sistema de transmissão da CHESF. Portanto, o número de variáveis de controle contínuas sujeitas a pequenas variações que podem trazer apenas benefícios incrementais é muito pequeno.

4.5 Inviabilidades na Solução do FPO

Um aspecto muito importante de um programa de FPO é como ele se comporta quando o problema é matematicamente inviável, ou seja, quando os limites operacionais não podem ser respeitados. Ao invés de finalizar o processo como não resolvido, o FPO deveria fornecer uma solução que fosse a melhor possível, com ou sem instruções interativas. É desejável que o algoritmo de FPO seja capaz de identificar a inviabilidade do problema de forma decisiva e rápida. Infelizmente, a maioria dos métodos de FPO são deficientes neste propósito.

Uma vez tendo sido um problema de FPO considerado inviável, ele pode ser alterado e resolvido de duas formas alternativas [1]:

1. Com os controles e/ou restrições do FPO modificados. As modificações possíveis incluem:
 - adição de novos controles (liberando controles previamente fixados, conectando novos geradores, etc);
 - expansão dos valores dos limites operacionais (de valores de longo-prazo para médio-prazo, ou em último caso ignorar os limites);
 - modificação da topologia da rede;
 - redução ou corte de carga (normalmente essa deve ser a última ação a ser tomada).
2. Com a função objetivo modificada para prover uma solução na qual os limites que causam a inviabilidade da solução sejam minimamente violados.

O termo *solução de mínima violação* pode ser definido de diferentes maneiras. A mais comum é penalizar as violações na forma de mínimos quadrados. Isso significa aumentar a função objetivo, tal como o custo de geração ou perdas elétricas, com uma série de funções de penalidades de mínimos desvios.

Para ilustrar esta idéia, considere o FPO expresso na forma padrão (2.1), reescrita

abaixo por conveniência:

$$\begin{aligned} & \text{Minimize} && f(x) \\ & \text{sujeito a} && g(x) = 0 \\ & && l \leq x \leq u \end{aligned}$$

Sejam \mathcal{V}_m e \mathcal{V}_M os conjuntos dos índices das restrições de limites $l_i \leq x_i$ e $x_i \leq u_i$, respectivamente, que são violadas. Como uma variável x_i não pode violar seus dois limite simultaneamente, temos que $\mathcal{V}_m \cap \mathcal{V}_M = \emptyset$. O problema modificado para minimizar violações de limites pode ser da forma:

$$\begin{aligned} & \text{Minimize} && f(x) + c \sum_{i \in \mathcal{V}_m} w_i (x_i - l_i)^2 + c \sum_{i \in \mathcal{V}_M} w_i (x_i - u_i)^2 \\ & \text{sujeito a} && g(x) = 0, \\ & && l_i \leq x_i \leq u_i, \quad \text{para } i \notin \mathcal{V}_m \cup \mathcal{V}_M, \end{aligned} \tag{4.4}$$

em que c é um número positivo grande, que pondera a minimização das violações em relação ao objetivo original, e w_i é o peso da restrição denotando o seu grau de importância em relação as demais restrições violadas.

A escolha de c é arbitrária, e a solução exata pode ser obtida atribuindo-se um valor suficientemente grande para c de forma que a soma dos quadrados das violações seja o termo dominante da função objetivo modificada. No entanto, conforme discussões em [3], a busca de violações rigorosamente mínimas (atribuindo-se valores muito grandes para c) pode requerer modificações excessivas nos controles, muitas das quais proporcionando apenas reduções superficiais das violações.

As estratégias (1) e (2) não são mutuamente exclusivas. Portanto, se a estratégia (1) falhar na busca de uma solução viável, pode-se recorrer a estratégia (2), e vice-versa. Deve-se observar que expandir uniformemente os limites pode fornecer soluções muito diferentes daquelas projetadas para minimizar violações.

Em [47] é proposta uma metodologia para identificação e tratamento de inviabilidades no FPO via o método primal-dual de PI e suas variantes. A metodologia emprega uma comutação de algoritmos de solução quando ocorre dificuldade de convergência e inviabilidade é detectada. Na detecção de inviabilidade considera-se que um dos primeiros

sintomas de um problema inviável, no contexto da solução via o método primal-dual de PI, é a tendência a zero do comprimento de passo primal definido pela equação (3.9), o que resulta na estagnação do processo iterativo. Isso ocorre porque uma ou mais variáveis de folga tendem rapidamente para zero. Adicionalmente, a inviabilidade do problema está relacionada com o comportamento do multiplicador de Lagrange associado ao limite que está provocando a inviabilidade. Este multiplicador terá um valor não nulo e, relativamente aos demais, muito maior em valor absoluto, representando a “pressão” do processo de convergência para ir além, no sentido de ultrapassar o limite imposto pelo problema.

O procedimento em [47] inicia o monitoramento de inviabilidade quando o comprimento de passo α_k torna-se demasiadamente pequeno (≈ 0). Mais especificamente, se $\alpha_k \leq 0,0001$ durante 3 ou mais iterações consecutivas, então o problema de FPO possivelmente é inviável, e os conjuntos de índices dos limites inferiores e superiores possivelmente inviáveis são definidos como segue:

$$\mathcal{V}_L = \{i \mid s_i \leq \tau_1 \text{ e } \pi_i \geq \tau_2, i = 1, 2, \dots, n\}, \quad (4.5)$$

$$\mathcal{V}_U = \{i \mid z_i \leq \tau_1 \text{ e } v_i \geq \tau_2, i = 1, 2, \dots, n\}. \quad (4.6)$$

É desejável que todos os limites inviáveis estejam incluídos nos conjuntos \mathcal{V}_L e \mathcal{V}_U , e que nenhum limite viável seja incorretamente incluído num desses conjuntos, porém, não há garantia que essa situação ideal ocorra sempre na metodologia proposta em [47].

Como dito anteriormente, quando inviabilidade no FPO é detectada ela é geralmente tratada aumentando-se o número de controles, modificando a função objetivo, ou relaxando algumas restrições. Esta abordagem é natural dado que inexistência de um ponto viável é a origem do problema. Na metodologia em [47] o problem relaxado é na forma (4.4). No entanto, para a implementação computacional eficiente todas as restrições de limites são mantidas na formulação, preservando o tanto quanto possível a estrutura do sistema de Newton (3.6), sendo os limites inviáveis removidos implicitamente pela expansão dos limites (diminuindo o limite inferior ou aumentando o limite superior). Em resumo, resolve-se

o problema modificado:

$$\begin{aligned} \text{Minimize} \quad & f(x) + \sum_{i \in \mathcal{V}_L} w_i(x_i - l_i)^2 + \sum_{i \in \mathcal{V}_U} w_i(x_i - u_i)^2 \\ \text{sujeito a} \quad & g(x) = 0, \\ & \tilde{l} \leq x \leq \tilde{u}, \end{aligned} \tag{4.7}$$

em que $\tilde{l}_i = l_i - \delta$ se $i \in \mathcal{V}_L$ e $\tilde{u}_i = u_i + \delta$ se $i \in \mathcal{V}_U$; δ é o montante da relaxação nos limites violados.

Devido a formulação e a resolução de um sistema linear de grande porte em cada iteração ser a tarefa mais dispendiosa no algoritmo de PI, a metodologia mantém a estrutura desse sistema linear inalterada. A idéia de preservação da estrutura do problema de FPO é motivada por estratégias de partida quente em programação quadrática sequencial [48] e ela torna a comutação de algoritmos bastante simples.

4.6 Discretização de Variáveis

Algumas variáveis de controle do FPO são contínuas (tensões de geradores, potência ativa de geradores, etc) e outras são discretas (tapes de transformadores, susceptâncias de capacitores e reatores em paralelo, etc). Normalmente, dada a enorme complexidade que é a solução de um problema de otimização não-linear com variáveis discretas [49, 50], os métodos de otimização empregados na solução do FPO tratam todas as variáveis como contínuas. Uma vez encontrada uma solução ótima contínua, cada variável discreta que é tratada como contínua é aproximada para um valor discreto, geralmente o seu valor discreto mais próximo. Todavia, após tal procedimento, a solução encontrada dificilmente é uma solução inteira-mista rigorosamente ótima, de forma que novas soluções de FPO podem ser requeridas [5].

A modelagem exata das variáveis de controle discretas juntamente com as contínuas torna o problema de FPO num problema de programação não-linear inteira-mista [51], de forma que soluções rigorosamente ótimas, envolvendo controles discretos, requerem procedimentos de pesquisa combinatorial, o que torna a ferramenta de FPO inaceitavelmente lenta para aplicações em tempo-real [52].

Métodos clássicos de programação inteira, tais como *branch-and-bound* (B&B) e *cutting plane* são não polinomiais. Consequentemente, eles são lentos e pouco atrativos para aplicações em problemas de grande porte. Conforme descrição em [50], o método B&B é baseado na idéia de enumerar de forma “inteligente” todos os pontos viáveis de um problema de otimização combinatorial. Nesta enumeração o método constrói uma prova de que uma solução é ótima baseado em partições sucessivas do espaço de solução. O termo *branch* (ramo) em *branch-and-bound* refere-se a este processo de particionamento; o termo *bound* (limite) refere-se a limites inferiores que são usados para construir uma prova de otimalidade sem realizar a busca exaustiva.

Na literatura de sistemas de potência a referência [53] trata as variáveis discretas propondo um método de pesquisa aproximada, num problema de planejamento de potência reativa. Na referência [54] os dispositivos discretos capacitores e reatores em paralelo são tratados num problema de FPO usando os algoritmos de penalidade de Dribeck's. Na referência [52] é proposto um algoritmo de discretização baseado em penalidades. O algoritmo trata a discretização dos capacitores e reatores em paralelo durante o processo de FPO baseado no método de Newton sem uma pesquisa combinatorial. O artigo mostra que o algoritmo pode ser facilmente incorporado aos programas de FPO existentes.

Pesquisa recente em [55] procurou obter soluções discretas pela simples reformulação do problema de FPO com variáveis discretas ainda como um problema de otimização contínua, seja pela inclusão de penalidades na função objetivo ou pela incorporação de restrições adicionais no problema. Todavia, os resultados numéricos obtidos não foram satisfatórios, uma vez que foram semelhantes aos resultados obtidos pelo simples arredondamento da solução contínua para valores discretos próximos. Ficou caracterizado que a natureza discreta das variáveis deve ser necessariamente explorada pela técnica de solução em vez da simples reformulação do problema como um problema de otimização contínua, uma vez que tal reformulação requer a obtenção de soluções ótimas globais o que é igualmente de grande complexidade.

4.7 Interface Gráfica com o Usuário

Os principais programas para análise de sistemas de potência existentes no mercado, sejam programas de FPO ou não, fazem uso de sofisticadas interfaces gráficas com o usuário,

comumente chamadas de GUI's. O termo GUI significa *Graphical User Interface*, e refere-se a idéia universal de ícones, botões, menus *popup*, etc., que são visualmente apresentados ao usuário como a linha de frente de um software aplicativo. Os recursos de interface gráfica do programa **OOTrans** analisado são ilustrados no Apêndice B.

Uma GUI fornece uma plataforma amigável para o usuário interagir com a ferramenta computacional. Um aspecto muito importante de uma GUI é que ela pode prover uma maneira simples e segura para o usuário do aplicativo interagir com o programa sem o conhecimento de comandos de programação. O funcionamento geral das GUI's baseia-se na reação a ações ou eventos, ou seja, o estado de uma GUI é alterado apenas quando o usuário ou outros elementos de código desencadeiam uma determinada ação ou evento. Quando bem projetada e programada, uma GUI, além de constituir uma interface homem-máquina amigável, pode reduzir a praticamente zero a possibilidade de erros no uso do programa computacional, uma vez que a digitação e memorização de sequências e sintaxes de comandos são reduzidas ao máximo.

Em programas de FPO os parâmetros dos componentes da rede elétrica, dos modelos de otimização e dos algoritmos de solução, assim como as funções e ferramentas de análise podem ser convenientemente acessados através da GUI. Através de diagramas unifilares é possível montar e modificar a topologia da rede elétrica graficamente. Todas as informações relacionadas com um componente qualquer do sistema elétrico podem ser facilmente acessadas por meio de um duplo clique na representação do componente no diagrama unifilar.

Um programa de FPO deve permitir a entrada de dados da rede elétrica nos diversos formatos utilizados pelas empresas, como os formatos ANAREDE, PTI, CDF, etc. Deve também dispor de recursos para verificação de inconsistência nos dados, tais como: (i) verificação de topologia para identificação de barras isoladas, ilhamentos, e ramos paralelos com mesma identificação, (ii) verificação de dados de barras como barras controladas por mais de um dispositivo de controle e barras com geração negativa, (iii) verificação de ramos como ramos com impedâncias muito grandes ou muito pequenas, com reatância negativa, ou com fluxos excedendo a impedância de surto (SIL), etc.

Após um processo de solução qualquer (um fluxo de potência simples ou fluxo de potência ótimo), o usuário poderá solicitar diversos tipos de relatórios, além dos relatórios usuais de uma solução de fluxo de potência. Além de relatórios impressos, diversos tipos

de gráficos poderão ser gerados a partir de uma GUI.

Os programas de FPO devem dispor de recursos para visualização dos dados de entrada e dos resultados em diagramas unifilares. O usuário pode criar diagramas unifilares simplesmente clicando-e-arrastando componentes de uma *palette* de componentes para a área do diagrama. Como o diagrama está conectado a ferramenta de análise, os dados da rede podem ser modificados no diagrama unifilar.

A Figura 4.1 ilustra um dos recursos para exibição de resultados disponível no simulador PowerWorld [56] (ver também www.powerworld.com). Neste simulador os fluxos nos circuitos são exibidos no diagrama unifilar de forma animada (flechas movendo-se ao longo da linha, no sentido do fluxo), ilustrando tanto o sentido do fluxo quanto a magnitude relativa do fluxo. Disjuntores nos circuitos, nas cargas e nos geradores permitem que o usuário retire ou insira esses componentes com um simples clique do mouse. Ao simular a remoção de um componente no diagrama unifilar o estado do sistema é automaticamente recalculado e o impacto dessa mudança pode ser imediatamente visualizado.

Mapas de contorno surgem também como um outro recurso visual muito interessante introduzido pela ferramenta PowerWorld. Um mapa de contorno prover uma “visão de pássaro” de como tensões, cargas, perdas e sensibilidade de fluxos, fatores de distribuição de perdas, preço *spot*, e fluxos de ramos e interfaces variam ao longo do sistema. Os contornos de cores chamam a atenção para aquelas áreas do sistema que apresentam problemas. Esta funcionalidade apresenta-se como sendo de grande importância no monitoramento em tempo real de sistemas.

4.8 Análise de Sensibilidade

Quase sempre é desejável ter informações sobre a sensibilidade da solução em relação a vários aspectos do problema. E uma característica muito interessante do FPO resolvido por métodos de pontos-interiores são as informações de sensibilidade contidas nos multiplicadores de Lagrange. Como conhecido na literatura [44, 51], um multiplicador de Lagrange diz algo sobre a sensibilidade da função objetivo em relação a presença da restrição a ele associada. Em outras palavras, o multiplicador de Lagrange indica o quão forte a restrição influencia no valor ótimo da função objetivo. Quando uma determinada restrição é inativa,

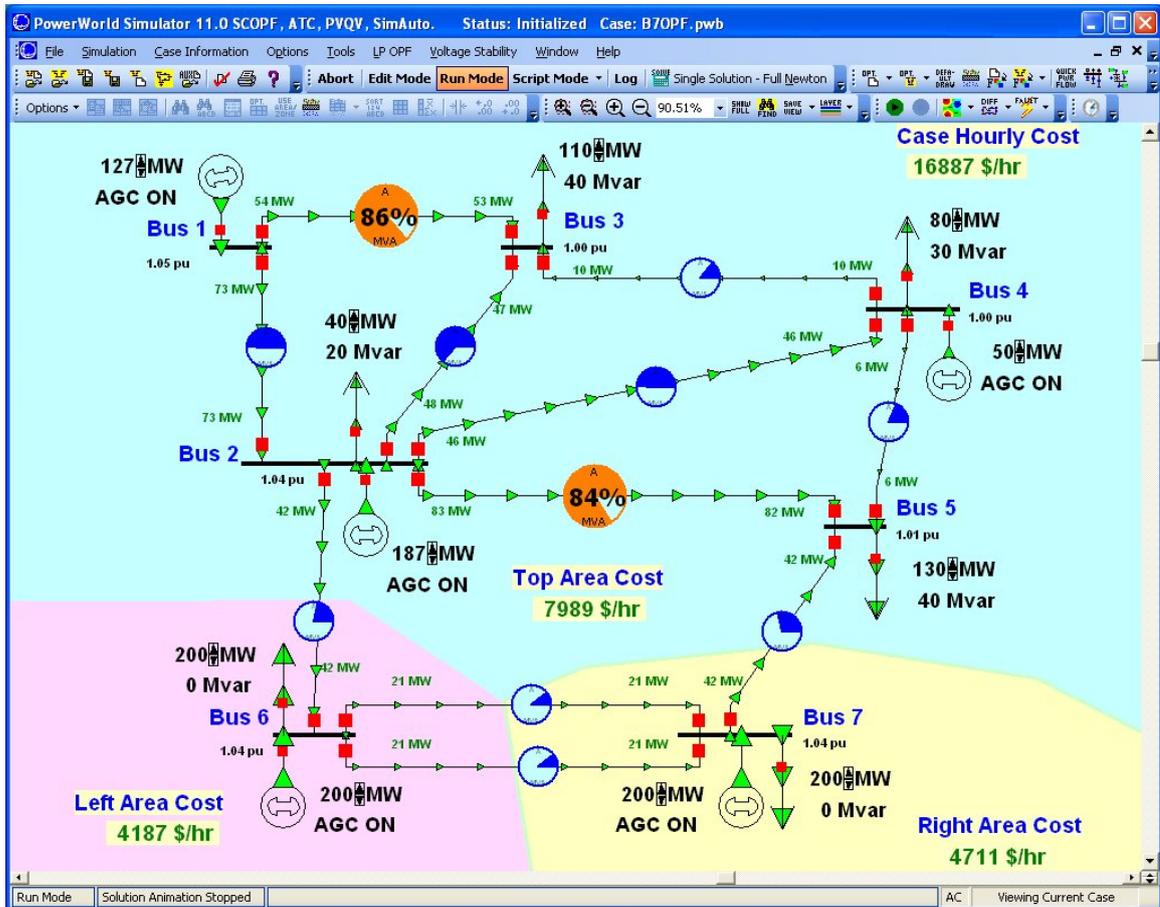


Figura 4.1: Diagrama unifilar com animação para exibição de resultados no programa PowerWorld.

a solução ótima x_* e o valor da função $f(x_*)$ são indiferentes à presença ou não dessa restrição, e assim o multiplicador de Lagrange correspondente deverá ser nulo.

É importante ressaltar que apesar do resultado obtido num processo de otimização ser ótimo, sua otimalidade está associada ao conjunto de restrições especificado. Dessa forma, informações sobre a sensibilidade das restrições são bastante úteis a medida que as restrições podem não ser definidas rigidamente no contexto de um problema. Por exemplo, num modelo as restrições muitas vezes representam propriedades desejáveis da solução em vez de essenciais. Assim, relaxando um pouco uma dada restrição poderá não ser crítico se a função objetivo irá melhorar significativamente. Alternativamente, no sentido oposto, pode ser benéfico impor requisitos mais restritivos quando o impacto no objetivo ótimo é insignificante. Portanto, uma análise pós-otimalidade de sensibilidade possibilita que um

melhor ponto de operação possa ser encontrado. Por exemplo, a análise de sensibilidade pode identificar os limites operativos que mais fortemente restringem o valor ótimo da função objetivo.

Pode-se concluir que em programas de otimização em geral são necessários meios para exibição de informações relativas ao processo de otimização, como valores de folgas das restrições de limites e dos multiplicadores de Lagrange associados. É desejável que um programa de FPO calcule e forneça diversas informações de sensibilidade como, por exemplo, sensibilidades de fluxos em linhas e interfaces (para identificação dos controles mais efetivos no alívio do carregamento de um dado circuito), sensibilidade de perdas (para análises econômicas e definição de penalidades no aumento de perdas do sistema), preços *spot*, etc.

Capítulo 5

Aplicação de Programa de FPO em Sistema Real de Subtransmissão

Nos campos da observação, o acaso favorece apenas as mentes preparadas.

Louis Pasteur

ESTE CAPÍTULO apresenta os resultados numéricos obtidos com um programa de fluxo de potência ótimo denominado **OOTrans** (Otimização da **O**peração da **T**ransmissão), quando aplicado na minimização de perdas elétricas no sistema de subtransmissão CELPE. O programa foi exaustivamente testado nos 13 regionais que compõem o sistema CELPE.

Os objetivos principais desses experimentos iniciais são: (i) verificar o desempenho computacional dos algoritmos de otimização implementados com relação a robustez e rapidez de convergência, (ii) comparar o desempenho dos algoritmos que competem entre si como o algoritmo mais eficiente para solução do FPO, e (iii) ajustar os parâmetros dos algoritmos de otimização implementados.

5.1 Características do Sistema de Subtransmissão

A rede elétrica de subtransmissão CELPE não é totalmente interligada, sendo composta por 13 regionais, cujos quantitativos de número total de barras $|\mathcal{N}|$, número de barras de geração $|\mathcal{G}|$, número de barras de carga com compensação de potência reativa fixa ou nula $|\mathcal{F}|$, número de barras de carga com compensação de potência reativa controlável $|\mathcal{C}|$, número de linhas (Linhas), número de transformadores com tape fixo (Fixo), e número de transformadores com tape variável (LTC), são apresentados na Tabela 5.1. As localizações dos 13 regionais no Estado de Pernambuco são ilustradas na Figura 5.1.

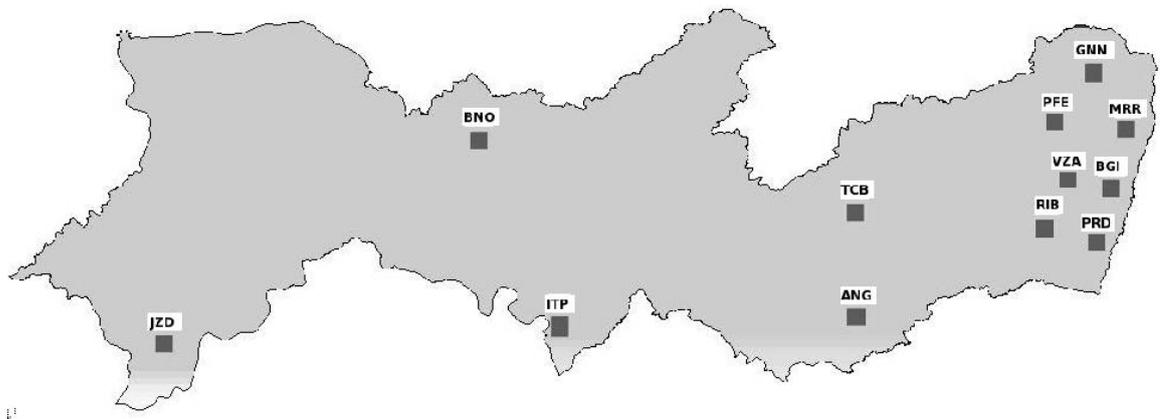


Figura 5.1: Localização dos regionais do sistema elétrico CELPE.

A Tabela 5.2 apresenta os quantitativos referentes as variáveis de controle discretas existentes nos regionais, a saber: as compensações de potência reativa controláveis em paralelo (bancos de capacitores chaveáveis), e os tapes dos transformadores com LTC. Com relação as compensações de potência reativa em paralelo, são mostrados o número total de barras por regional, e os números de barras com 1, 2, 3 e 4 módulos capacitivos. A tabela mostra ainda, com relação a compensação de potência reativa, por regional, o menor

Tabela 5.1: Número de barras, circuitos e transformadores do sistema CELPE.

Regional	Número de Barras				Linhas	Transformadores	
	$ N $	$ G $	$ F $	$ C $		Fixo	LTC
Angelim	47	1	33	13	32	14	12
Bongi	51	1	38	12	36	25	6
Bom Nome 69 kV	22	1	13	8	12	4	11
Bom Nome 138 kV	38	1	27	10	21	9	16
Goianinha	23	1	12	10	25	20	2
Itaparica	5	1	2	2	2	4	-
Juazeiro	29	1	21	7	19	12	7
Mirueira	42	1	19	22	32	17	6
Pau Ferro	54	1	43	10	40	27	7
Pirapama	94	1	80	13	65	37	8
Ribeirão	42	1	30	11	33	21	5
Tacaimbó	49	1	39	9	31	13	14
Várzea	18	1	13	4	13	9	2
Total	514	13	370	131	361	212	96

(Δ_{\min}) e o maior (Δ_{\max}) incremento de potência reativa no chaveamento dos módulos de compensação.

Os quantitativos de barras por número de módulos de compensação são informações importantes na análise do grau de dificuldade na obtenção de uma solução ótima discreta. As informações de incrementos mínimos e máximos são dados importantes na análise do impacto de estratégias de arredondamento de soluções contínuas para valores discretos.

As três últimas colunas da Tabela 5.2 apresentam os dados dos transformadores com LTC, a saber: o número total de transformadores com LTC, o números de tapes discretos n_p , e o valor aproximado dos incrementos de tapes (Δ_p). Alguns regionais apresentam transformadores com número de tapes e incrementos de tapes distintos, de forma que para simplificar a apresentação dos dados foram indicados apenas um valor de n_p e de Δ_p , aquele de maior frequência no regional.

5.2 Os Sistemas Testes

A Tabela 5.3 apresenta, para fins de comparação com as soluções ótimas obtidas, o montante de perdas em cada regional (em MW e como um percentual da carga total), na

Tabela 5.2: Quantidades e modulações das susceptâncias *shunts* e tapes de LTC's.

Regional	Barras com Compensação de Reativo							LTC's		
	Total	Por Modulação				Variações		Total	n_p	$\approx \Delta_p$
		1	2	3	4	Δ_{\min}	Δ_{\max}			
Angelim	13	8	5	-	-	1.02	4.06	12	32	.006
Bongi	12	-	8	4	-	3.05	6.09	6	16	.012
Bom Nome 69 kV	8	4	4	-	-	1.02	3.05	11	32	.006
Bom Nome 138 kV	10	3	6	1	-	0.51	2.03	16	16	.012
Goianinha	10	1	8	1	-	1.02	10.0	2	32	.008
Itaparica	2	-	-	2	-	1.02	2.03	-	-	-
Juazeiro	7	-	4	2	1	1.02	2.03	7	16	.012
Mirueira	12	1	5	5	1	2.03	4.06	6	16	.012
Pau Ferro	10	2	7	1	-	1.02	3.05	7	16	.012
Pirapama	13	3	6	4	-	1.02	4.06	8	32	.006
Ribeirão	11	2	7	1	1	1.02	2.03	5	32	.006
Tacaimbó	9	2	7	-	-	1.02	4.06	14	32	.006
Várzea	4	1	1	2	-	2.03	4.06	2	32	.006
Total	121	27	68	23	3	-	-	96	-	-

condição de carga máxima.

Com relação a complexidade dos problemas de FPO de cada regional, a Tabela 5.4 mostra o número de variáveis, n , o número de restrições de igualdades, m , e o número de restrições de limites simples sobre variáveis, p . A Tabela 5.4 mostra também que o número de soluções discretas possíveis é expressivamente elevado, mesmo otimizando os regionais individualmente. A título de exemplo, observa-se que o número de soluções discretas possíveis para o regional Angelim é de $2,95147905179352825856 \times 10^{20}$.

Faz-se necessário o desenvolvimento de heurísticas para a transformação de soluções contínuas em soluções discretas, uma vez que o tratamento de um número de soluções discretas dessa ordem de grandeza é impraticável numa aplicação em tempo-real. Esse elevado número de soluções discretas decorre principalmente do elevado número de tapes discretos associados com um único LTC (32 tapes). Por outro lado, os incrementos de tapes são valores pequenos, da ordem de 0.006, de maneira que a discretização das soluções contínuas obtidas para os valores discretos mais próximos não constituirá em degradação significativa da solução ótima.

Ao contrário dos tapes, os incrementos das compensações de potência reativa são de

Tabela 5.3: Montante de perdas por regional na solução não-otimizada.

Regional	Carga Máxima		Perdas Ativas	
	MW	Mvar	(MW)	(%)
Angelim	91,30	38,10	9,64	10,56
Bom Nome 138 kV	48,20	22,90	7,02	14,56
Bom Nome 69 kV	46,10	19,60	5,17	11,21
Bongi	322,10	153,70	3,53	1,10
Goianinha	114,50	49,50	4,55	3,97
Itaparica	4,50	2,10	0,07	1,56
Juazeiro	60,40	29,40	1,77	2,93
Mirueira	258,60	119,10	2,22	0,86
Pau Ferro	155,30	70,00	11,78	7,59
Pirapama	204,70	98,40	4,05	1,98
Ribeirão	80,60	37,40	2,54	3,15
Tacaimbó	106,40	52,50	4,24	3,99
Várzea	133,70	57,90	0,91	0,68
Totais	1.626,40	750,60	57,49	3,54

ordens significativas, variando de 1.02 a 10.06 Mvar. Assim, o esforço da otimização discreta deverá ser voltado para o tratamento das susceptâncias em paralelo, recorrendo ainda ao auxílio de heurísticas e do “bom senso de engenharia” para reduzir o espaço de busca e melhorar o desempenho computacional do algoritmo de otimização discreta.

5.3 Os Dados Necessários

O programa **OOTrans**, como qualquer outro programa de FPO, requer basicamente três conjuntos de dados: (i) os dados da rede elétrica (dados de barras e de linhas, como os que são requeridos por um programa de fluxo de potência tradicional), (ii) os dados para a otimização (como os limites físicos de equipamentos e os limites operacionais), e (iii) os parâmetros do algoritmo de otimização.

Os dados da rede elétrica são fornecidos como arquivos padrões de programas de fluxo de potência. O programa **OOTrans** trabalha com quatro formatos de arquivos diferentes, dentre eles: o formato ANAREDE (amplamente utilizado pela CELPE e pelas empresas de energia elétrica no Brasil) e o formato *IEEE Common Data Format*. Internamente ao

Tabela 5.4: Número de variáveis n , número de restrições de igualdades m , número de limites simples sobre as variáveis p , e número de soluções discretas possíveis, por regional.

Regional	n	m	p	Número de Soluções Discretas
Angelim	126	100	80	$(2^8 \times 3^5) \times (32^{12}) \approx 2,95 \times 10^{20}$
Bongi	128	109	78	$(3^8 \times 4^4) \times (16^{06}) =$
Bom Nome 69 kV	66	46	45	$(2^4 \times 3^4) \times (32^{11}) =$
Bom Nome 138 kV	107	80	70	$(2^3 \times 3^6 \times 4^1) \times (16^{16}) =$
Goianinha	64	51	46	$(2^2 \times 3^8 \times 4^1) \times (32^{02}) =$
Itaparica	13	10	9	$4^2 = 16$
Juazeiro	77	62	49	$(3^4 \times 4^2 \times 5^1) \times (16^{07}) =$
Mirueira	119	90	78	$(2^1 \times 3^5 \times 4^5 \times 5^1) \times (16^{06}) =$
Pau Ferro	134	116	81	$(2^2 \times 3^7 \times 4^1) \times (16^{07}) =$
Pirapama II	224	201	131	$(2^3 \times 3^6 \times 4^4) \times (32^{08}) =$
Ribeirão	108	91	67	$(2^2 \times 3^7 \times 4^1 \times 5^1) \times (32^{05}) =$
Tacaimbó	128	104	80	$(2^2 \times 3^7) \times 32^{14} =$
Várzea	46	39	29	$(2^1 \times 3^1 \times 4^2) \times (32^{02}) =$

programa **OOTrans**, o formato padrão de dados da rede é o formato *IEEE Common Data Format*. Quando o formato do arquivo de dados selecionado é o ANAREDE ou outro que não o IEEE, então uma função de conversão do arquivo para o formato interno padrão (*IEEE Common Data Format*) é automaticamente executada, de forma que o arquivo de dados ANAREDE passa a ser disponível também no formato *IEEE Common Data Format*.

Os dados adicionais para a otimização, como os limites físicos de equipamentos e os limites operacionais, são fornecidos num segundo arquivo de terminação .fpo, o qual tem uma formatação que foi desenvolvida especialmente para o programa **OOTrans**, uma vez que não há, na literatura especializada, uma padronização definida para esse conjunto de dados. Pode-se considerar essa formatação como sendo ainda provisória, e um arquivo exemplo é apresentado no final deste relatório.

O terceiro conjunto de dados inclui os parâmetros do algoritmo de otimização. Esses parâmetros não são fornecidos via arquivos, mas sim através da interface gráfica com o usuário do programa. Valores padrões são definidos para todos os parâmetros, os quais são automaticamente exibidos quando o programa é carregado na memória. O usuário, se assim desejar, pode alterar qualquer desses parâmetros, a qualquer instante. A Tabela 5.5 lista os parâmetros, uma breve descrição dos mesmos, e os seus valores padrões.

Além dos ajustes dos parâmetros relacionados, o usuário pode influenciar o processo

de solução escolhendo entre vários algoritmos de fluxo de potência e vários algoritmos de otimização por pontos interiores. Os algoritmos de fluxo de potência disponíveis são: (i) método de Newton completo, (ii) método desacoplado rápido versão BX, (iii) método desacoplado rápido versão XB, (iv) método de Gauss-Seidel, e (v) método DC. O algoritmo de solução de fluxo de potência padrão é o método desacoplado rápido versão BX.

Os algoritmos de otimização por métodos de pontos-interiores baseados na função de barreira logarítmica, que estão disponíveis no programa **OOTrans**, são:

MPD : método primal-dual simples,

MPC : método preditor-corretor,

PCM : método preditor-corretor múltiplo, e

MCC : método de múltiplas correções de centralidade.

Tabela 5.5: Valores padrões dos parâmetros dos algoritmos.

Parâmetro	Descrição	Valor
Fluxo de Potência		
k_{\max}	Número máximo de iterações de fluxo de potência	20
ϵ_P	Resíduo máximo no balanço de potência ativa	0.001
ϵ_Q	Resíduo máximo no balanço de potência reativa	0.001
ϵ_V	Tolerância na convergência dos módulos de tensão	0.0001
α	Comprimento de passo na atualização das variáveis	1
Otimização		
k_{\max}	Número máximo de iterações de pontos-interiores	30
μ_0	Parâmetro de barreira inicial	0.01
σ	Parâmetro de centralização na redução de μ_k	0.2
γ	Fator de redução do comprimento de passo	0.99995
M	Número máximo passos de correção (métodos MPC e MCC)	5
β_{\min}	Limite inferior do hipercubo de projeção	0.1
β_{\max}	Limite superior do hipercubo de projeção	10
γ_α	Aumento mínimo (%) aceitável no comprimento de passo	10
ϵ_1	Tolerância da inviabilidade primal	10^{-4}
ϵ_2	Tolerância da inviabilidade dual	10^{-4}
ϵ_3	Tolerância do resíduo de complementaridade	10^{-5}
ϵ_μ	Tolerância do parâmetro de barreira final	10^{-8}

Os algoritmos listados estão em ordem crescente de robustez numérica. O programa **OOTrans** é o único programa de FPO que implementa os métodos preditor-corretor múltiplo e de múltiplas correções de centralidade. Estes algoritmos de pontos-interiores foram originalmente desenvolvidos para a solução de problemas de *programação linear* [29, 30], e posteriormente estendidos para a solução de problemas não-lineares de fluxo de potência ótimo [31, 46].

5.4 Resumo de Análises de Fluxo de Potência

No que diz respeito a topologia da rede, o sistema de subtransmissão CELPE não é totalmente interligado, sendo organizado em 13 subsistemas chamados de regionais. A carga total, em condições de carga máxima, é de aproximadamente 1.626,4 MW e 750,6 Mvar, enquanto as perdas elétricas globais são de 57,5 MW e 349,9 Mvar. Portanto, as perdas elétricas ativas totalizam cerca de 3,53% da carga ativa total.

Os regionais Angelim, Bom Nome 138kV, Bom Nome 69kV e Pau Ferro apresentam-se como os mais críticos em termos de violações de tensões, e/ou carregamento de circuitos. Alguns regionais são bastantes sensíveis à configuração de banco de capacitores, como o Regional Bom Nome 69 kV e Bom Nome 138 kV. A Tabela 5.7 resume outras análises de fluxo de potência, as quais consideram também casos de contingências e a condição de carga mínima.

Tabela 5.7: Quadro síntese da análise de redes do sistema CELPE (extraído do Relatório III do Projeto de P&D).

Regional	Configuração	Tipo de Violação			
		Tensão nas Barras		Carregamento	
		C. Máx.	C. Mín.	C. Máx.	C. Mín.
Angelim	Normal	√		√	
	Perda da LT 02J4	√		√	
Bongi	Normal				
	Perda da LT 02J3			√	
	Perda da LT 02J4			√	
	Perda da LT 02L5				
	Perda da LT 02L6				
Bom Nome 69 KV	Normal	√			
Bom Nome 138KV	Normal	√			
Goianinha	Normal				
	Perda da LT 02L3				
	Perda da LT 02L4				
	Perda da LT 02M5	√		√	√
	Perda da LT 02M6	√		√	
	Perda da LT 02M9	√		√	
Itaparica	Normal				
Juazeiro	Normal				
Mirueira	Normal				
	Perda da LT 02J4				
	Perda da LT 02J5			√	
	Perda da LT 02J6			√	
	Perda da LT 02L5				
	Perda da LT 02V3				
	Perda da LT 02V4				
Pau Ferro	Normal	√		√	
	Perda da LT 02V7	√	√	√	√
	Perda da LT 02V8	√	√	√	√
Pirapama	Normal				
	Perda da LT 02J2	√			
	Perda da LT 02J5	√	√		
	Perda da LT 02J6	√	√		
	Perda da LT 02J8	√		√	
Ribeirão	Normal				
Tacaimbó	Normal				
Várzea	Normal				

5.5 Otimização dos Regionais

Esta seção apresenta os resultados numéricos obtidos na otimização dos 13 regionais do sistema CELPE. As simulações foram inicialmente realizadas utilizando os valores padrões dos parâmetros dos algoritmos, ou seja, com o usuário simplesmente informando os nomes dos arquivos de dados correspondentes aos regionais e clicando no botão de execução. Os algoritmos de otimização não foram ajustados para a forma mais adequada para cada regional. Procedendo desta maneira, pode-se comprovar a robustez dos algoritmos de otimização.

5.5.1 Solução com Parâmetros Padrões

A Tabela 5.8 resume o desempenho computacional dos 4 algoritmos de otimização, no que se refere ao número de iterações para convergência, bem como apresenta o montante de perdas elétricas na solução ótima encontrada para cada regional. Vale salientar que os montantes de perdas apresentados na Tabela 5.3, referentes a solução não-otimizada, já faz uso de praticamente todos os recursos de compensação de reativos existentes nos regionais. Ainda assim, as soluções otimizadas apresentaram reduções de perdas em todos os regionais, sendo as reduções mais significativas nos regionais Bom Nome 138 kV e Bom Nome 69 kV.

Uma vez que as soluções iniciais de fluxo de potência já fazem uso dos recursos de compensação de potência reativa, o maior benefício da solução otimizada revela-se na expressiva melhoria no perfil de tensão dos regionais, como pode ser deduzido comparando-se as soluções otimizadas com as soluções simples de fluxo de potência.

No que diz respeito ao desempenho computacional dos algoritmos de otimização, os quatro algoritmos de pontos-interiores (MPD, MPC, PCM e MCC) apresentaram desempenho bastante satisfatório, uma vez que todos eles obtiveram a convergência na otimização dos treze regionais. Observa-se da Tabela 5.8 que o número de iterações para convergência é bastante modesto, variando de 6 a 13 iterações.

Observa-se também da Tabela 5.8 que o número de iterações não cresce na mesma proporção de crescimento do tamanho dos sistemas. Por exemplo, o Regional Itaparica com apenas 5 barras requer 9 iterações com o algoritmo MPD, enquanto o Regional Pirapama

Tabela 5.8: Desempenho dos algoritmos de otimização utilizando os parâmetros padrões.

Regional	Número de Iterações				Perdas Ativas	
	MPD	MPC	PCM	MCC	(MW)	(%)
Angelim	12	8	7	7	9,33	10,22
Bom Nome 138 kV	12	8	8	8	5,89	12,22
Bom Nome 69 kV	11	8	7	6	4,35	9,44
Bongi	10	8	6	8	3,34	1,04
Goianinha	10	7	6	6	4,42	3,86
Itaparica	9	6	6	6	0,05	1,11
Juazeiro	9	6	6	6	1,67	2,76
Mirueira	11	9	6	6	2,07	0,80
Pau Ferro	13	9	7	8	11,30	7,28
Pirapama	10	7	7	7	4,00	1,95
Ribeirão	10	7	6	6	2,52	3,13
Tacaimbó	13	11	8	10	4,04	3,80
Várzea	9	6	6	6	0,92	0,69
Totais	139	100	86	90	53,90	3,31

com 94 barras requer 10 iterações do mesmo algoritmo, ou seja, o número de barras do Regional Pirapama é cerca de 20 vezes maior que o do Regional Itaparica enquanto o acréscimo no número de iterações é de apenas 1.

A última linha da Tabela 5.8 resume o desempenho global dos algoritmos, totalizando os dados na otimização dos treze regionais. Claramente, o melhor desempenho computacional, em termos de número de iterações, foi o do método preditor-corretor múltiplo (PCM), o qual demandou um total de 86 iterações para otimizar os treze regionais, uma média de apenas 6 a 7 iterações por regional, inobstante as dimensões desses regionais. Todavia, pode-se dizer que o desempenho do método de múltiplas correções de centralidade (MCC), com um total de 90 iterações, é comparável ao do método preditor-corretor múltiplo. Em seguida, vem o método preditor-corretor (MPC) com 100 iterações e, por fim, o método primal-dual simples com um total de 136 iterações.

5.5.2 Influência do Ponto Inicial

O programa **OOTrans** dispõe de várias regras (heurísticas) para a escolha de um ponto inicial para o algoritmo de otimização selecionado. Por exemplo, as variáveis primais

x podem ser inicializadas segundo os valores obtidos numa solução inicial do fluxo de potência tradicional (não-otimizado), ou utilizando o ponto médio entre os limites para as variáveis sujeitas a limites mínimos e máximos, ou ainda uma inicialização “flat” como ocorre nas soluções de fluxo de potência.

A Tabela 5.8 considera as variáveis primais x iniciadas por uma solução inicial de fluxo de potência. A Tabela 5.9 considera as variáveis primais x iniciadas pelo ponto médio para aquelas variáveis sujeitas a limites mínimos e máximos. Comparando-se as duas tabelas conclui-se que o desempenho é melhor quando o ponto inicial é escolhido segundo uma solução inicial de fluxo de potência. No entanto, essa não é uma regra geral. A influência da escolha do ponto inicial no processo de convergência precisa ser melhor estudada e testada.

Tabela 5.9: Desempenho dos algoritmos com as variáveis x iniciadas pela regra do ponto médio (opção *Ponto Médio* no Popup Menu).

Regional	Número de Iterações				Perdas Ativas	
	MPD	MPC	PCM	MCC	(MW)	(%)
Angelim	23	30	24	21	9,33	10,22
Bom Nome 138 kV	13	11	9	8	5,89	12,22
Bom Nome 69 kV	22	24	24	13	4,35	9,44
Bongi	12	7	6	6	3,34	1,04
Goianinha	11	10	7	7	4,42	3,86
Itaparica	9	6	6	6	0,05	1,11
Juazeiro	10	7	6	7	1,67	2,76
Mirueira	12	7	7	7	2,07	0,80
Pau Ferro	16	13	10	13	11,30	7,28
Pirapama	11	9	9	-	4,00	1,95
Ribeirão	11	8	7	6	2,52	3,13
Tacaimbó	17	11	11	8	4,04	3,80
Várzea	10	6	6	7	0,92	0,69
Totais	165	149	132	109	53,90	3,31

5.5.3 Influência da Regra do Comprimento de Passo

O programa **OOTrans** possibilita tanto o uso de comprimentos de passos iguais para atualização das variáveis primais e duais, quanto o uso de comprimentos de passos distintos: α_P para as variáveis primais x , s e z ; e α_D para as variáveis duais π , ν e λ .

Na programação linear, o uso de comprimentos de passos distintos nos espaços primal e dual geralmente apresenta melhor desempenho computacional (em termos de número de iterações para convergência) do que passos primal e dual iguais. Porém, essa vantagem não ocorre com frequência na programação não-linear, como é o caso do fluxo de potência ótimo.

O uso de comprimentos de passos iguais, embora possa eventualmente demandar um número ligeiramente maior de iterações, está a favor da robustez de convergência, uma vez que não despreza o forte acoplamento entre as variáveis primais e duais nas condições de otimalidade. As Tabelas 5.8 e 5.10 apresentam o desempenho dos algoritmos quando são utilizados comprimentos de passos iguais e distintos, respectivamente. Verifica-se que o uso de comprimentos de passos α_P e α_D distintos reduz o número de iterações para convergência em todos os algoritmos. No entanto, o algoritmo MCC não convergiu com passos distintos na otimização do Regional Tacaimbó.

Tabela 5.10: Desempenho dos algoritmos quando podem utilizar comprimentos de passos distintos: $\alpha_k^P \neq \alpha_k^D$.

Regional	Número de Iterações				Perdas Ativas	
	MPD	MPC	PCM	MCC	(MW)	(%)
Angelim	11	7	6	7	9,33	10,22
Bom Nome 138 kV	11	7	7	6	5,89	12,22
Bom Nome 69 kV	10	7	7	6	4,35	9,44
Bongi	10	7	6	7	3,34	1,04
Goianinha	9	6	6	6	4,42	3,86
Itaparica	8	6	6	6	0,05	1,11
Juazeiro	9	6	6	7	1,67	2,76
Mirueira	10	11	6	7	2,07	0,80
Pau Ferro	11	7	7	7	11,30	7,28
Pirapama	10	9	8	10	4,00	1,95
Ribeirão	11	6	6	6	2,52	3,13
Tacaimbó	12	10	7	-	4,04	3,80
Várzea	9	6	6	6	0,92	0,69
Totais	131	95	84	81	53,90	3,31

5.5.4 Influência do Parâmetro de Barreira Inicial

As Tabelas 5.8, 5.11 e 5.12 apresentam o desempenho computacional dos algoritmos para três escolhas do parâmetro de barreira inicial, a saber: $\mu_0 = 0.01$, $\mu_0 = 0.1$ e $\mu_0 = 1$. Os testes realizados sugerem escolher $\mu \in [10^{-2}, 1]$, com a melhor escolha sendo $\mu_0 = 0.01$. A faixa de convergência do parâmetro μ_0 é relativamente larga, desde que o parâmetro de centralização σ seja adequadamente escolhido (valor típico $\sigma = 0.2$). A maior limitação na escolha de μ_0 é que ele não seja escolhido demasiadamente pequeno.

Verifica-se que os quatro algoritmos de otimização convergiram para os três valores de parâmetros de barreira, a única exceção sendo o método preditor-corretor (MPC) na otimização do Regional Mirueira com $\mu_0 = 1$. Todavia, o melhor desempenho global, em termos de número de iterações, ocorre com o parâmetro de barreira inicial $\mu_0 = 0.01$. Vale salientar que se outros parâmetros forem ajustados a convergência desse algoritmo pode ser restaurada com esse mesmo valor de parâmetro de barreira.

Tabela 5.11: Desempenho dos algoritmos com $\mu_0 = 0.1$.

Regional	Número de Iterações				Perdas Ativas	
	MPD	MPC	PCM	MCC	(MW)	(%)
Angelim	14	8	8	8	9,33	10,22
Bom Nome 138 kV	14	9	8	9	5,89	12,22
Bom Nome 69 kV	13	8	8	8	4,35	9,44
Bongi	12	7	7	13	3,34	1,04
Goianinha	11	7	7	7	4,42	3,86
Itaparica	10	7	7	7	0,05	1,11
Juazeiro	11	7	7	7	1,67	2,76
Mirueira	13	8	7	8	2,07	0,80
Pau Ferro	15	9	9	10	11,30	7,28
Pirapama	12	7	7	7	4,00	1,95
Ribeirão	12	8	7	7	2,52	3,13
Tacaimbó	15	11	8	9	4,04	3,80
Várzea	10	7	7	7	0,92	0,69
Totais	162	103	97	107	53,90	3,31

Tabela 5.12: Desempenho dos algoritmos com $\mu_0 = 1$.

Regional	Número de Iterações				Perdas Ativas	
	MPD	MPC	PCM	MCC	(MW)	(%)
Angelim	15	9	9	9	9,33	10,22
Bom Nome 138 kV	15	11	9	9	5,89	12,22
Bom Nome 69 kV	15	8	8	8	4,35	9,44
Bongi	13	8	8	9	3,34	1,04
Goianinha	13	8	8	8	4,42	3,86
Itaparica	11	8	8	8	0,05	1,11
Juazeiro	12	8	8	8	1,67	2,76
Mirueira	14	-	8	8	2,07	0,80
Pau Ferro	16	10	9	19	11,30	7,28
Pirapama	13	11	8	8	4,00	1,95
Ribeirão	13	9	8	8	2,52	3,13
Tacaimbó	17	12	9	10	4,04	3,80
Várzea	12	8	8	8	0,92	0,69
Totais	179	110	108	120	53,90	3,31

5.6 Observações Finais

Os resultados numéricos que são apresentados neste capítulo demonstram a efetividade do programa computacional **OOTrans** desenvolvido e a ser continuamente melhorado. As Tabelas 5.8 a 5.12 resumem um conjunto de 260 simulações que foram realizadas envolvendo os treze regionais da CELPE e quatro algoritmos de otimização. Dentre as 260 simulações realizadas apenas 3 foram casos de não convergência. Todavia, a convergência desses casos pode ser facilmente restaurada através de ajustes simples de parâmetros do algoritmo.

Capítulo 6

Conclusões

*A mente que se abre para uma nova idéia jamais
voltará ao seu tamanho original.*

Albert Einstein

NESTA DISSERTAÇÃO foram abordadas algumas dificuldades na aplicação de programas de FPO em sistemas elétricos reais, sobretudo no que diz respeito a otimização da operação em tempo real. Essas dificuldades decorrem do fato de que o FPO no mundo real é matematicamente e computacionalmente muito diferente das formulações clássicas. Infelizmente, não há uma técnica de otimização que possa tratar esses tipos de problemas eficientemente com todo rigor na modelagem, e a prática universal tem sido aproximar a

formulação de tal forma que o problema possa ser resolvido pelas técnicas de otimização disponíveis.

Precedendo a discussão das dificuldades relacionadas à aplicação de programas de FPO em sistemas elétricos reais, foram apresentados nos Capítulos 2 e 3 algumas formulações clássicas do FPO e os algoritmos de pontos-interiores que ultimamente são empregados na solução. Esses algoritmos de pontos-interiores são os implementados no programa de FPO nessa pesquisa. A aplicação do FPO em tempo real exige que a solução seja obtida de forma rápida e confiável, sendo passível de implementação imediata por um sistema de controle automatizado ou pelo operador do sistema, e os algoritmos de pontos-interiores podem atender esse requisito de rapidez e robustez.

As análises apresentadas nos Capítulos 4 e 5 são as contribuições dessa pesquisa de mestrado. O Capítulo 4 faz uma análise não exaustiva de vários requisitos indispensáveis na aplicação de programas de FPO em sistemas reais. Dentre os requisitos analisados, é indiscutível que o mais crítico é o tratamento de variáveis discretas. Além do número de variáveis discretas num sistema elétrico ser muito elevado, esse problema combinatorial torna-se mais crítico ainda pelo grande número de valores discretos que é admitido por uma única variável, como é o caso de taps de transformadores que tipicamente admitem até 32 valores discretos, como compensação paralela de reativo, esses passos são de grande magnitude.

Para o tratamento de variáveis discretas, a prática usual é obter uma solução ótima contínua e então arredondar o valor ótimo contínuo para o valor discreto mais próximo. Este procedimento não apenas leva a solução sub-ótimas como também pode resultar em violações de limites. Pesquisa recente [55] procurou tratar o requisito cálculo de soluções discretas por meio de soluções aproximadas, de baixo custo computacional, com a pretensão apenas de obter soluções melhores do que o simples arredondamento para o valor discreto mais próximo. Todavia, os resultados ainda não foram plenamente satisfatórios.

Outro requisito igualmente, crítico, por tratar-se também de problema combinatorial, é o de minimização do número de controles ajustados. Nas soluções de FPO calculadas em geral todas as variáveis de controles sofrem ajustes em relação aos valores correntes na operação. Todavia, pode não ser praticável, ou até mesmo ser impossível, implementar todos os ajustes ótimos calculados. Além do mais, o benefício de alguns desses ajustes é apenas marginal, sendo desejável a supressão destes para simplificar a implementação dos

ajustes e preservar a vida útil dos equipamentos.

Recentemente o requisito supressão de ajustes ineficazes foi eficientemente tratado em [55]. Algumas características da metodologia proposta em [55] são:

- resolvem-se três problemas de FPO com estruturas idênticas, num processo de solução com três etapas, denominadas de Fase I, Fase II e Fase III;
- a solução de FPO calculada na Fase I emprega todos os controles e é utilizada para definir o problema de FPO da Fase II;
- os multiplicadores de Lagrange obtidos na solução do FPO da Fase II são usados para identificar as ações de controle mais efetivas. O número máximo de controles permitido deve ser especificado pelo usuário com base em práticas operacionais;
- a solução de FPO calculada na Fase III reajusta as magnitudes dos controles selecionados para compensar a ação dos controles que deixam de ser modificados;
- os problemas de FPO nas três fases tem a mesma estrutura e assim podem ser resolvidos pelo mesmo algoritmo de otimização.

A metodologia para supressão de ajustes ineficazes proposta xxx faz parte do desenvolvimento contínuo do programa OOTrans.

Um aspecto muito importante de um FPO em tempo real é como ele se comporta quando o problema é matematicamente inviável, ou seja, quando uma ou mais restrições do problema não podem ser respeitadas, Num caso de inviabilidade, ao invés de finalizar o processo como não resolvido, o programa de FPO deveria fornecer um ponto de operação no qual os limites inviáveis sejam minimamente violados, com ou sem instruções interativas. Ou seja, o algoritmo de FPO deve ser capaz de identificar a inviabilidade do problema de forma decisiva e rápida, bem como tratá-la da melhor maneira possível. No entanto, a maioria dos algoritmos de solução são deficientes neste propósito. A melhor maneira de abordar a detecção e o tratamento de inviabilidades é altamente dependente da forma que o FPO é formulado e da técnica de otimização utilizada na solução.

Como parte do desenvolvimento contínuo do programa OOTrans, em [55] é proposta uma abordagem para detecção e tratamento de inviabilidades no problema de FPO, particularmente na solução via método primal-dual de PI e suas variantes. Um algoritmo de

solução unificado capaz de tratar tanto casos viáveis quanto inviáveis é descrito. A metodologia emprega um chaveamento de algoritmos de solução quando uma dificuldade de convergência surge e a inviabilidade é detectada. Devido a formulação e a resolução de um sistema linear de grande porte em cada iteração ser a tarefa mais dispendiosa no algoritmo de PI, a metodologia proposta mantém a estrutura desse sistema linear inalterada. A idéia de preservação da estrutura do problema de FPO é motivada por estratégias de partida quente em programação quadrática sequencial [48], e ela torna a comutação dos algoritmos bastante simples.

Com relação a aplicação do programa OOTrans no sistema real de subtransmissão, verificou-se que os algoritmos de otimização (pontos-interiores) nele implementados são rápidos e confiáveis o suficiente para aplicações em tempo real. Não há deficiências críticas em relação ao algoritmo de otimização por ser um algoritmo para otimização contínua, é o tratamento eficiente dos controles discretos. O programa OOTrans é dotado de sofisticada interface gráfica com o usuário, a qual permite ao usuário ter total controle sobre o processo de otimização, desde a escolha do ponto inicial ao ajuste de qualquer parâmetro do algoritmo de otimização. A interface gráfica do OOTrans também permite uma fácil análise de todo o processo de solução ao dispor de recursos para a pesquisa de qualquer informação relativa a solução.

6.1 Sugestões de Trabalhos Futuros

A presente dissertação fez uma análise de vários requisitos para aplicação do FPO em situações reais. O tratamento eficiente desses requisitos ainda demanda por bastante pesquisa, como por exemplo:

- Desenvolvimento de metodologia para definição da sequência de implementação dos ajustes ótimos dos controles;
 - Desenvolvimento de metodologia para o tratamento eficiente das variáveis de controle discretas. Por tratamento eficiente entende-se um procedimento de solução sem o alto custo computacional (tempo de processamento) dos métodos tradicionais de otimização discreta, tal como o algoritmo Branch e Bound, porém sem degradar significativamente a qualidade da solução;
-

- Incorporação de restrições de estabilidade de tensão nas formulação do FPO, visando a obtenção de pontos de operação com margens mínimas de segurança de tensão;
 - Desenvolvimento de metodologia para pré-seleção das contingências mais críticas, visando a redução do problema de otimização na obtenção de soluções restritas por contingências;
 - Desenvolvimento de indicadores de inviabilidade mais precisos para a metodologia de tratamento de inviabilidades, evitando erros na identificação das restrições de limite inviáveis ou a classificação errônea de problemas viáveis como inviáveis.
-

Referências Bibliográficas

- [1] B. Stott, O. Alsac, and A. J. Monticelli, “Security analysis and optimization,” *Proceedings of the IEEE*, vol. 75, pp. 1623–1644, Dec. 1987.
- [2] J. Momoh, R. G. Koessler, M. S. Bond, and B. Stott, “Challenge to optimal power flow,” *IEEE Trans. on Power Systems*, vol. 12, pp. 444–455, Feb. 1997.
- [3] S.-K. Chang, G. E. Marks, and K. Kato, “Optimal real-time voltage control,” *IEEE Trans. on Power Systems*, vol. 5, pp. 750–758, Aug. 1990.
- [4] W. F. Tinney, J. M. Bright, K. D. Demaree, and B. A. Hughes, “Some deficiencies in optimal power flow,” *IEEE Trans. on Power Systems*, vol. 3, pp. 676–683, May 1988.
- [5] A. D. Papalexopoulos, C. F. Imparato, and F. F. Wu, “Large scale optimal power flow: Effects of initialization, decoupling and discretization,” *IEEE Trans. on Power Systems*, vol. 4, pp. 748–759, May 1989.
- [6] E. Vaahedi and H. M. Z. El-Din, “Considerations in applying optimal power flow to power systems operation,” *IEEE Trans. on Power Systems*, vol. 4, pp. 694–703, May 1989.

- [7] Q. Ding, N. Li, and X. Wang, "Implementation of interior point method based voltage/reative power optimization," in *Proc. of the IEEE PES General Meeting*, (Edmonton, Canada), June 2000.
- [8] G. L. Torres, C. A. Cañizares, and V. R. Simoni, "Supressão de ajustes ineficazes na solução de fluxo de potência ótimo," in *Aceito para o SBSE 2010: Simpósio Brasileiro de Sistemas Elétricos*, (Belém, PA, Brasil), pp. 1–7, May 2010.
- [9] P. Kundur, *Power System Stability and Control*. McGraw-Hill, Inc., 1994.
- [10] A. Monticelli, *Fluxo de Carga em Redes de Energia Elétrica*. Edgard Blücher Ltda, 1983.
- [11] M. Huneault and F. D. Galiana, "A survey of the optimal power flow literature," *IEEE Trans. on Power Systems*, vol. 6, pp. 762–770, May 1991.
- [12] H. W. Dommel and W. F. Tinney, "Optimal power flow solution," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-87, pp. 1866–1876, Oct. 1968.
- [13] H. Wei, H. Sasaki, and R. Yokoyama, "An application of interior point quadratic programming algorithm to power system optimization problems," *IEEE Trans. on Power Systems*, vol. 11, pp. 260–266, Feb. 1996.
- [14] X. Yan and V. H. Quintana, "An efficient predictor-corrector interior point algorithm for security-constrained economic dispatch," *IEEE Trans. on Power Systems*, vol. 12, pp. 803–810, May 1997.
- [15] K. C. Almeida and F. D. Galiana, "Critical cases in the optimal power flow," *IEEE Trans. on Power Systems*, vol. 11, pp. 1509–1518, Aug. 1996.
- [16] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by Newton approach," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-103, pp. 2864–2880, Oct. 1984.
- [17] A. J. Monticelli and W. E. Liu, "Adaptive movement penalty method for the Newton optimal power flow," *IEEE Trans. on Power Systems*, vol. 7, pp. 334–342, Feb. 1992.
-

- [18] K. A. Clements, P. W. Davis, and K. D. Frey, "Treatment of inequality constraints in power system state estimation," *IEEE Trans. on Power Systems*, vol. 10, pp. 567–573, May 1995.
- [19] Y. Wu, A. S. Debs, and R. E. Marsten, "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flow," *IEEE Trans. on Power Systems*, vol. 9, pp. 876–883, May 1994.
- [20] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Trans. on Power Systems*, vol. 9, pp. 136–146, Feb. 1994.
- [21] G. D. Irisarri, X. Wang, J. Tong, and S. Mokhtari, "Maximum loadability of power systems using interior point non-linear optimization method," *IEEE Trans. on Power Systems*, vol. 12, pp. 162–172, Feb. 1997.
- [22] G. L. Torres and V. H. Quintana, "An interior point method for nonlinear optimal power flow using voltage rectangular coordinates," *IEEE Trans. on Power Systems*, vol. 13, pp. 1211–1218, Nov. 1998.
- [23] S. Granville, J. C. O. Mello, and A. C. G. Melo, "Application of interior point methods to power flow unsolvability," *IEEE Trans. on Power Systems*, vol. 11, pp. 1096–1103, May 1996.
- [24] K. R. Frisch, "The logarithmic potential method of convex programming," Manuscript, University Institute of Economics, Oslo, Norway, 1955.
- [25] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, 1968.
- [26] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.
- [27] N. Megiddo, "Pathways to the optimal set in linear programming," Technical Report RJ 5295, IBM Almaden Research Center, San Jose, CA, 1986.
- [28] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM J. on Optimization*, vol. 2, pp. 575–601, 1992.
-

- [29] T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno, “Higher-order predictor-corrector interior point methods with applications to quadratic objectives,” *SIAM J. on Optimization*, vol. 3, no. 4, pp. 696–725, 1993.
- [30] J. Gondzio, “Multiple centrality corrections in a primal-dual method for linear programming,” *Computational Optimization and Applications*, vol. 6, pp. 137–156, 1996.
- [31] G. L. Torres, *Nonlinear Optimal Power Flow by Interior-Point and Non-Interior-Point Methods*. PhD thesis, University of Waterloo, Canada, 1998.
- [32] G. L. Torres and V. H. Quintana, “On a nonlinear multiple-centrality-corrections interior-point method for optimal power flow,” *IEEE Trans. on Power Systems*, vol. 16, pp. 222–228, May 2001.
- [33] G. L. Torres, V. H. Quintana, and M. A. de Carvalho, “Higher-order interior-point methods for computing minimum load shedding and maximum loadability of power systems,” in *Proc. of the VIII SEPOPE*, (Brasilia, Brazil), May 2002.
- [34] 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 Systems*, vol. 15, pp. 170–176, Feb. 2000.
- [35] K. A. Clements, P. W. Davis, and K. D. Frey, “An interior point algorithm for weighted least absolute value power system state estimation,” *1991 IEEE/PES Winter Meeting*. Paper no. 91-WM 235-2 PWRs.
- [36] L. Vargas, V. Quintana, and A. Vannelli, “A tutorial description of an interior-point method and its application to security constrained economic dispatch,” *IEEE Trans. on Power Systems*, vol. 11, no. 3, pp. 1315–1324, 1993.
- [37] L. V. Barboza and R. Salgado, “Restoring solutions for unsolvable cases via minimum load shedding for a specified direction,” in *Proc. of IEEE 22nd Power Industry Computation Conference*, (Sydney, Australia), p. CDROM, May 2001.
- [38] J. Medina, V. H. Quintana, A. Conejo, and F. P. Thoden, “A comparison of interior-point codes for medium-term hydro-thermal coordination,” *IEEE Trans. on Power Systems*, vol. 13, pp. 836–843, Aug. 1998.
-

- [39] X. Wang, G. C. Ejebe, J. Tong, and J. G. Waight, "Preventive/corrective control for voltage stability using direct interior point method," *IEEE Trans. on Power Systems*, vol. 13, pp. 878–883, Aug. 1998.
- [40] K. Ponnambalam, V. H. Quintana, and A. Vannelli, "A fast algorithm for power system optimization problems using an interior point method," *IEEE Trans. on Power Systems*, vol. 7, pp. 892–899, May 1992.
- [41] C. N. Lu and M. R. Unum, "Network constrained security control using an interior point algorithm," *IEEE Trans. on Power Systems*, vol. 8, pp. 1068–1076, Aug. 1993.
- [42] D. G. Luenberger, *Linear and Nonlinear Programming*. Addison-Wesley Inc., 1984.
- [43] R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons, 1987.
- [44] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.
- [45] S. J. Wright, *Primal-Dual Interior-Point Methods*. SIAM, 1997.
- [46] G. L. Torres and V. H. Quintana, "On a nonlinear multiple-centrality-corrections interior-point method for optimal power flow," in *Proc. of IEEE 22nd Power Industry Computation Conference*, (Sydney, Australia), p. CDROM, May 2001. Also in *IEEE Trans. on Power Systems*, vol. 16, no. 2, pp. 222–228, May. 2001.
- [47] V. R. Simoni and G. L. Torres, "Tratamento de inviabilidades no fluxo de potência ótimo por métodos de pontos interiores," in *Anais do XIII Congresso Brasileiro de Automática*, (Bonito, MS), Sept. 2010.
- [48] J. Gondzio and A. Grothey, "Reoptimization with the primal-dual interior point method," *SIAM J. on Optimization*, vol. 13, pp. 842–864, 2003.
- [49] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. Wiley Interscience, 1999.
- [50] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.
- [51] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Academic Press Inc., 1981.
-

-
- [52] W. Liu, A. Papalexopoulos, and W. Tinney, “Discrete shunt controls in a Newton optimal power flow,” *IEEE Trans. on Power Systems*, vol. 7, pp. 334–340, Feb. 1992.
- [53] K. Aoli, M. Fan, and A. Nishikori, “Optimal var planning by approximation method for recursive mixed-integer linear programming,” *IEEE Trans. on Power Systems*, pp. 1741–1747, Nov. 1988.
- [54] A. G. Bakirtzis and A. P. S. Meliopoulos, “Incorporation in switching operations in power system corrective control computations,” *IEEE Trans. on Power Apparatus and Systems*, pp. 669–676, Aug. 1987.
- [55] V. R. Simoni, “Tratamento de requisitos para uso do fluxo de potência ótimo em tempo real,” Master’s thesis, Departamento de Engenharia Elétrica, Universidade Federal de Pernambuco, Brasil, 2010.
- [56] R. Klump, D. Schooley, and T. Overbye, “An advanced visualization platform for real-time power system operations,” in *Proc. of the 14th Power Systems Computation Conference*, (Sevilla, Spain), June 2002.
-

Anexo **A**

Implementação dos Algoritmos de Pontos-Interiores

A.1 Inicialização das Variáveis

- Escolha x_0 como dado por uma solução de fluxo de potência tradicional, ou como um “flat start” utilizando o ponto médio entre os limites superiores e inferiores para as variáveis sujeitas a limites, ou seja,

$$x_i^0 = \frac{l_i + u_i}{2}$$

- Uma vez escolhido x_0 , escolha as variáveis de folga s_0 e z_0 como

$$s_0 = \min\{\max\{x_0 - l, \tau(u - l)\}, (1 - \tau)(u - l)\} \quad (\text{A.1})$$

$$z_0 = u - l - s_0 \quad (\text{A.2})$$

em que $\tau \in (0, 1)$ é o percentual do valor máximo admissível para a variável.

- Obtidos s_0 e z_0 , escolha as variáveis duais π_0 e ν_0 como

$$\pi_0 = \mu_0 S_0^{-1} e \quad (\text{A.3})$$

$$\nu_0 = \mu_0 Z_0^{-1} e \quad (\text{A.4})$$

- Escolha $\lambda_i^0 = -1$ se este multiplicador está associado com uma restrição de balanço de potência ativa, e escolha $\lambda_i^0 = 0$ caso contrário.

A.2 Montagem da Matriz $\nabla g(x)$

Montar o sistema de Newton pode ser uma tarefa computacionalmente honerosa se as estruturas especiais da matriz gradiente $\nabla g(x)$ e das $m + 1$ matrizes Hessianas ($\nabla^2 g_i(x)$, $i = 1, 2, \dots, m$, e $\nabla^2 f(x)$), não forem eficientemente exploradas. Para esse fim, em [31] é proposto um mapeamento dos multiplicadores de Lagrange de *restrições para barras*,

$$\lambda \mapsto (\lambda^p, \lambda^q, \lambda^r), \quad (\text{A.5})$$

em que λ^p e λ^q são vetores $|\mathcal{N}|$ -dimensionais com os multiplicadores de Lagrange que são associados as restrições de balanço de potência ativa e potência reativa, respectivamente, e λ^r é o vetor de multiplicadores associados com as demais restrições.

O mapeamento (A.5) tem um papel crucial na implementação eficiente dos algoritmos, pois reduz o número de operações para calcular $\nabla_x L(y_k; \mu_k)$ e $\nabla_{xx}^2 L(y_k)$, e possibilita estruturas de dados muito eficientes e uso reduzido de memória de computador. Para tirar proveito do mapeamento (A.5), o vetor gradiente

$$\nabla_x L(y) = \nabla f(x) + \nabla g(x)\lambda - \pi + \nu \quad (\text{A.6})$$

é reescrito na forma

$$\nabla_x L(y) = \nabla p(x)\lambda^p + \nabla q(x)\lambda^q + \nabla r(x)\lambda^r \quad (\text{A.7})$$

em que $\nabla p(x)$ e $\nabla q(x)$ são as matrizes gradientes das funções de restrições balanço de potência ativa e potência reativa, respectivamente, e $\nabla r(x)$ é a matriz gradiente das demais restrições (fluxos, limites simples, etc). Diferentemente de $\nabla g(x)$ e $\nabla h(x)$, as matrizes $\nabla p(x)$ e $\nabla q(x)$ tem o mesmo padrão de elementos não-nulos, e assim compartilham os mesmos ponteiros (estrutura de adjacência) no armazenamento compacto de dados. Se $f(x)$ é a geração de potência ativa da barra de folga, então $\nabla f(x)$ está em $\nabla p(x)$ com $\lambda_1^p = 1$. Caso contrário, $\nabla f(x)$ está em $\nabla r(x)$.

A.3 Montagem das Matrizes Hessianas

O cálculo de $\nabla_{xx}^2 L(y_k)$, e portanto de $\nabla_{yy}^2 L(y_k)$, é uma tarefa computacionalmente intensiva. Como cada barra i introduz uma restrição de potência ativa e uma restrição de potência reativa em $g(x)$, ela contribui com duas Hessianas de funções injeção de potência em $\nabla_{xx}^2 L(y)$: $\nabla^2 P_i$ (como $\nabla^2 g_i(x)$) e $\nabla^2 Q_i$ (como $\nabla^2 g_{|N|+i}(x)$). Na hipótese das restrições de balanço de potência serem as únicas restrições não-lineares em $g(x)$, a Hessiana $\nabla_{xx}^2 L(y)$ pode ser reescrita como

$$\nabla_{xx}^2 L(y) = \sum_{i=1}^N (\lambda_i^p \nabla^2 P_i(x) + \lambda_i^q \nabla^2 Q_i(x)). \quad (\text{A.8})$$

Devido as estruturas especiais de $\nabla^2 P_i$ e $\nabla^2 Q_i$, a Hessiana $\nabla_{xx}^2 L(y)$ pode ser calculada de maneira bastante eficiente através de referências diretas aos elementos H_{ij} , L_{ij} , M_{ij} e N_{ij} do Jacobiano das equações do fluxo de potência:

$$\begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = - \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (\text{A.9})$$

Tirando proveito das estruturas de elementos não-nulos de $\nabla^2 P_i$ e $\nabla^2 Q_i$, e utilizando o mapeamento (A.5), a Hessiana $\nabla_{xx}^2 L(y)$ pode ser eficientemente calculada sem calcular qualquer das Hessianas $\nabla^2 P_i(x)$ e $\nabla^2 Q_i(x)$. Fórmulas diretas para calcular $\nabla_{xx}^2 L(y)$ foram

desenvolvidas em [31]. A submatriz $\nabla_{vv}^2 L$ é calculada por:

$$\nabla_{V_j V_j}^2 L = 2(G_{jj}\lambda_j^p - B_{jj}\lambda_j^q) \quad (\text{A.10a})$$

$$\nabla_{V_i V_j}^2 L = \frac{N_{ij}\lambda_i^p + L_{ij}\lambda_i^q}{V_i} + \frac{N_{ji}\lambda_j^p + L_{ji}\lambda_j^q}{V_j} \quad (\text{A.10b})$$

A submatriz $\nabla_{\theta v}^2 L$ é calculada por:

$$\nabla_{\theta_j V_j}^2 L = \frac{H_{ii}\lambda_i^p + M_{ii}\lambda_i^q + \sum_l (H_{li}\lambda_l^p + M_{li}\lambda_l^q)}{V_i} \quad (\text{A.10c})$$

$$\nabla_{\theta_i V_j}^2 L = -\frac{H_{ij}\lambda_i^p + M_{ij}\lambda_i^q - H_{ji}\lambda_j^p - M_{ji}\lambda_j^q}{V_j} \quad (\text{A.10d})$$

A submatriz $\nabla_{\theta\theta}^2 L$ é calculada por:

$$\nabla_{\theta_j \theta_j}^2 L = H_{ii}\lambda_i^q - M_{ii}\lambda_i^p - \sum_l (H_{li}\lambda_l^q - M_{li}\lambda_l^p) \quad (\text{A.10e})$$

$$\nabla_{\theta_i \theta_j}^2 L = H_{ij}\lambda_i^q - M_{ij}\lambda_i^p + H_{ji}\lambda_j^q - M_{ji}\lambda_j^p \quad (\text{A.10f})$$

As fórmulas (A.10b), (A.10d) e (A.10f) são avaliadas apenas se a barra i interliga-se com a barra j (devido as estruturas de elementos não-nulos de $\nabla_{vv}^2 L$, $\nabla_{\theta v}^2 L$ e $\nabla_{\theta\theta}^2 L$ serem idênticas a da matriz admitância de barra), e $i > j$ (devido a simetria de $\nabla_{xx}^2 L$). Para cada tape t_{ij} calcula-se os elementos

$$\nabla_{t_{ij}\theta_i}^2 L = \lambda_i^p \frac{\partial^2 P_i}{\partial t_{ij} \partial \theta_i} + \lambda_j^p \frac{\partial^2 P_j}{\partial t_{ij} \partial \theta_i} + \lambda_i^q \frac{\partial^2 Q_i}{\partial t_{ij} \partial \theta_i} + \lambda_j^q \frac{\partial^2 Q_j}{\partial t_{ij} \partial \theta_i} \quad (\text{A.11a})$$

$$\nabla_{t_{ij}\theta_j}^2 L = \lambda_i^p \frac{\partial^2 P_i}{\partial t_{ij} \partial \theta_j} + \lambda_j^p \frac{\partial^2 P_j}{\partial t_{ij} \partial \theta_j} + \lambda_i^q \frac{\partial^2 Q_i}{\partial t_{ij} \partial \theta_j} + \lambda_j^q \frac{\partial^2 Q_j}{\partial t_{ij} \partial \theta_j} \quad (\text{A.11b})$$

$$\nabla_{t_{ij}V_i}^2 L = \lambda_i^p \frac{\partial^2 P_i}{\partial t_{ij} \partial V_i} + \lambda_j^p \frac{\partial^2 P_j}{\partial t_{ij} \partial V_i} + \lambda_i^q \frac{\partial^2 Q_i}{\partial t_{ij} \partial V_i} + \lambda_j^q \frac{\partial^2 Q_j}{\partial t_{ij} \partial V_i} \quad (\text{A.11c})$$

$$\nabla_{t_{ij}V_j}^2 L = \lambda_i^p \frac{\partial^2 P_i}{\partial t_{ij} \partial V_j} + \lambda_j^p \frac{\partial^2 P_j}{\partial t_{ij} \partial V_j} + \lambda_i^q \frac{\partial^2 Q_i}{\partial t_{ij} \partial V_j} + \lambda_j^q \frac{\partial^2 Q_j}{\partial t_{ij} \partial V_j} \quad (\text{A.11d})$$

$$\nabla_{t_{ij}t_{ij}}^2 L = \lambda_i^p \frac{\partial^2 P_i}{\partial t_{ij} \partial t_{ij}} + \lambda_j^p \frac{\partial^2 P_j}{\partial t_{ij} \partial t_{ij}} + \lambda_i^q \frac{\partial^2 Q_i}{\partial t_{ij} \partial t_{ij}} + \lambda_j^q \frac{\partial^2 Q_j}{\partial t_{ij} \partial t_{ij}} \quad (\text{A.11e})$$

Anexo B

Estrutura do Programa OOTrans

ESTE ANEXO apresenta a estrutura do programa **OOTrans** (Otimização da Operação da **T**ransmissão), fazendo uma breve descrição das suas principais rotinas computacionais. A versão atual do programa **OOTrans** implementa a minimização de perdas ativas na transmissão e a minimização de corte de carga, conforme as formulações de FPO descritas no Capítulo 2, e dispõe dos quatro algoritmos de pontos-interiores descritos no Capítulo 3 para solução dos problemas de FPO, a saber: (i) primal-dual simples, (ii) primal-dual preditor-corretor, (iii) múltiplo preditor-correcotr, e (iv) múltiplas correções de centralidade. Uma análise do programa **OOTrans** quando aplicado a sistemas reais é apresentada no Capítulo 5, em conjunto com os resultados numéricos.

B.1 Utilizando o Programa OOTrans

O programa **OOTrans** constitui-se de 8 (oito) janelas de trabalho com funcionalidades específicas, como detalhadas a seguir. A janela principal de abertura do **OOTrans** apresenta-se como ilustrada na Figura B.1. Cada uma das 8 janelas de trabalho é acionada clicando-se no botão correspondente na coluna a esquerda da janela principal. No intuito de ilustrar a facilidade de uso do **OOTrans**, este relatório faz uma apresentação detalhada da aplicação do programa na otimização de um regional CELPE (Regional Angelim).

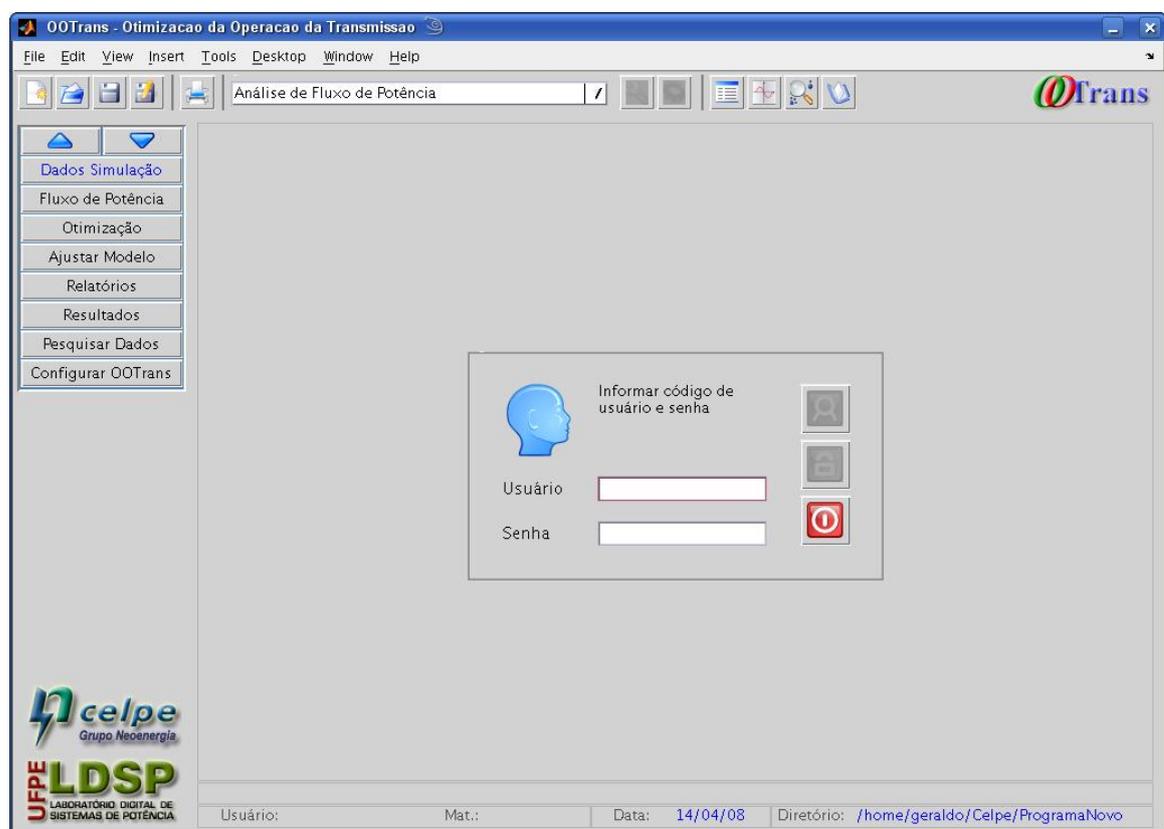


Figura B.1: Janela inicial do programa **OOTrans**.

Observa-se na janela principal a barra de ferramentas na parte superior da janela e os botões de acionamento das 8 janelas de trabalho na coluna a esquerda da janela. Todavia, sem a habilitação de um usuário previamente cadastrado pelo gerente (ou *superusuário*) do **OOTrans** nenhuma das janelas de trabalho poderá ser acionada.

B.1.1 Habilitação do Usuário

O primeiro passo, após carregar o programa, é a habilitação do usuário na janela de *login*, como ilustra a Figura B.2. Somente após a habilitação de um usuário será ativado o botão na barra de ferramentas que dá acesso as funcionalidades do **OOTrans**. Sem a devida habilitação de um usuário o programa não disponibilizará qualquer de suas funcionalidades, ou seja, as suas janelas de trabalho serão inacessíveis.



Figura B.2: Habilitação do usuário.

A barra de ferramentas que fica no topo da janela principal é como ilustra a Figura B.3. Clicando-se no botão



surge então a primeira janela de trabalho (**Dados Simulação**), como ilustra a Figura B.4, ou uma outra janela de trabalho que o usuário tenha clicado em um dos botões de acionamento na coluna da esquerda. Este botão tem dupla função: permitir o acesso as funcionalidade do programa quando um usuário tiver sido habilitado, ou restituir a janela de *login* da Figura B.1 quando o mesmo botão for novamente acionado.



Figura B.3: Barra de ferramentas após habilitação do usuário.

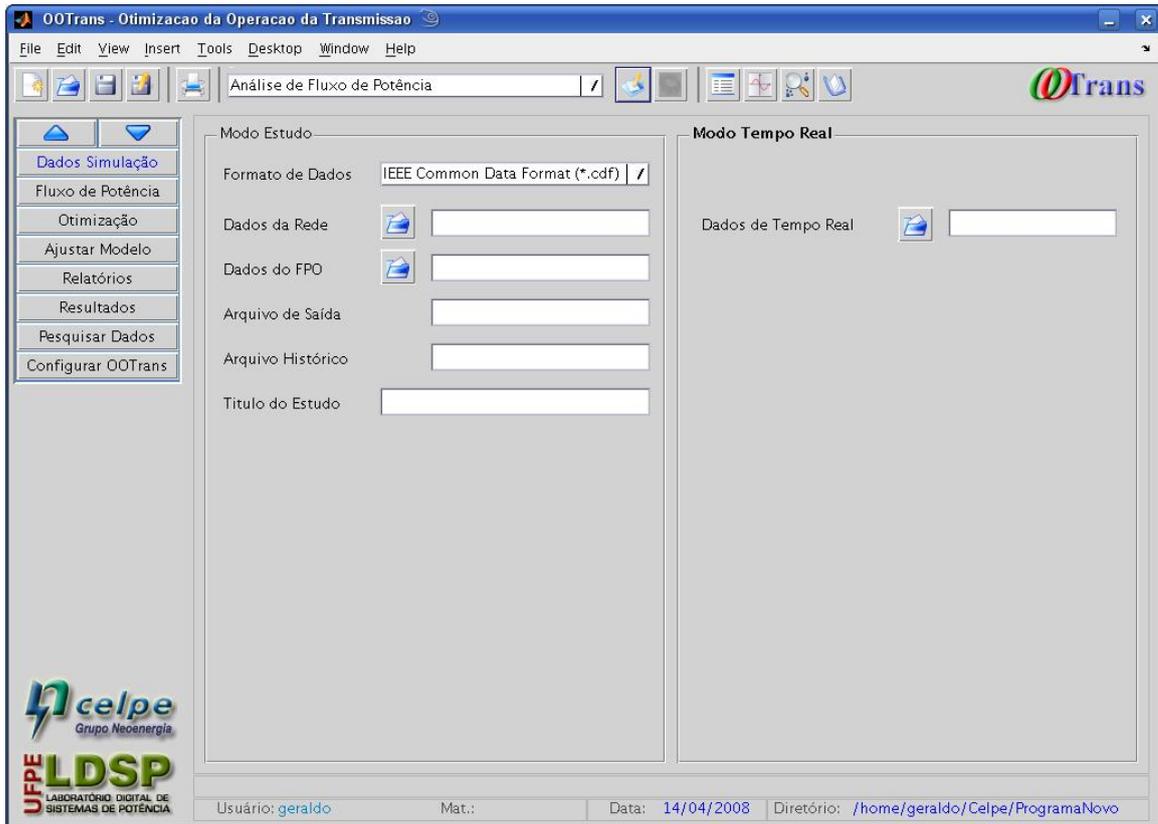


Figura B.4: Primeira pasta de configuração do programa OOTrans.

B.1.2 Informando os Arquivos de Dados

Retornando para a pasta **Dados Simulação** ilustrada na Figura B.4, para realizar uma simulação devemos primeiro informar o nome do arquivo que contém os dados da rede elétrica. Há três maneiras de fornecer o nome desse arquivo. Uma delas é clicar no botão **Abrir Arquivo** existente na barra de ferramentas (o segundo botão da esquerda para a direita):



Um clique no botão **Abrir Arquivo**,



aciona uma janela do sistema operacional que permite o usuário navegar na árvore de diretórios e selecionar o arquivo de dados da rede elétrica. A aparência dessa janela depende não somente do sistema operacional utilizado (Linux ou Windows) como também das configurações do *Desktop* desse sistema. No *desktop* em que este relatório está sendo gerado essa janela é como ilustra a Figura B.5.



Figura B.5: Janela de seleção de arquivos da rede.

A relação de arquivos que é listada na janela de navegação-e-seleção depende do formato de dados que foi selecionado no *Popup Menu* da janela de trabalho **Dados Simulação**, como ilustra a Figura B.6 a seleção do formato ANAREDE.

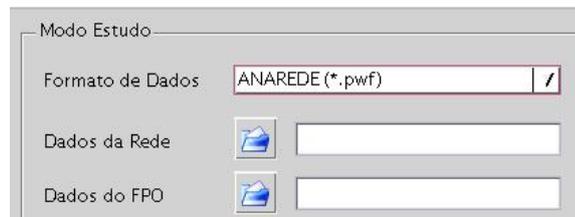


Figura B.6: Seleção do formato ANAREDE para os dados da rede.

No texto que segue, vamos considerar o formato de dados ANAREDE e otimizar o regional Angelim, cujo arquivo de dados da rede elétrica é *Angelim.pwf*. Após selecionar este arquivo na lista apresentada e clicar no botão **Open**, o painel **Modo Estudo** da janela de trabalho **Dados Simulação** toma a forma exibida na Figura B.7.

Observe que apesar do formato selecionado ter sido o ANAREDE, o nome do arquivo que é exibido na caixa de texto tem a terminação *.cdf*. Isto ocorre porque internamente o **OOTrans** trabalha apenas com o formato *IEEE Common Data Format*. Quando o formato

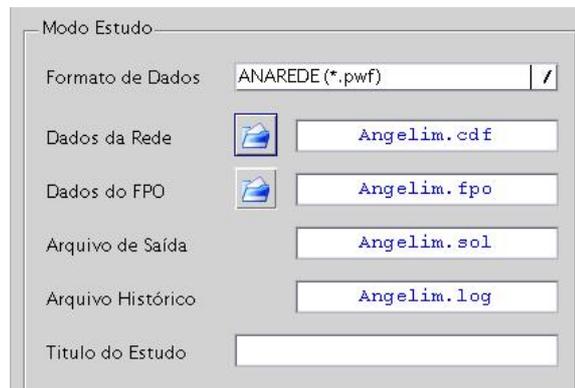


Figura B.7: Seleção do arquivo de dados `Angelim.pwf`.

selecionado é o ANAREDE ou outro qualquer, uma função de conversão desse formato para o formato interno padrão (*IEEE Common Data Format*) é automaticamente executada, de maneira que o arquivo de dados formato ANAREDE passa a ser disponível também no formato *IEEE Common Data Format*. Ou seja, o **OOTrans** dispõe de rotinas computacionais de conversão de tipos de arquivos de dados de rede elétrica usados por programas de fluxo de potência.

Observe que três novos nomes de arquivos são exibidos em caixas de textos no painel **Modo Estudo** da janela **Dados Simulação**, sob as denominações *Dados do FPO*, *Arquivo de Saída* e *Arquivo Histórico*. O arquivo denominado *Dados do FPO* (`Angelim.fpo`) é um arquivo de entrada, ou seja, um arquivo existente, contendo os dados da otimização, tais como limites de variáveis, etc. O programa automaticamente sugere o nome desse arquivo de dados, acrescentando a terminação `.fpo` ao nome principal do arquivo de dados da rede. Esse nome é apenas uma sugestão, e caso não seja esse o nome do arquivo com os dados de otimização, basta o usuário digitar o nome correto na caixa de texto ou clicar no botão de abrir arquivo ao lado da caixa de texto,



para acionar a janela de navegação e seleção de arquivos com terminação `.fpo`.

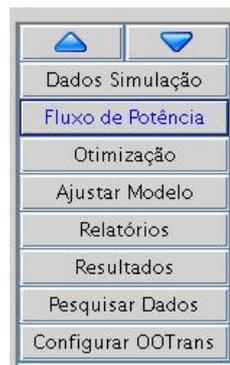
O arquivo denominado *Arquivo de Saída* (`Angelim.sol`) é um arquivo de saída onde serão escritos todos os relatórios relacionados a solução de fluxo de potência, conforme seleção na janela de trabalho **Relatórios**. Se um arquivo com este nome já existir, então

ele será sobrescrito com as novas informações. O nome desse arquivo pode ser alterado digitando-se um novo nome na caixa de texto correspondente.

O arquivo denominado *Arquivo Histórico* (*Angelim.log*) é um arquivo de saída onde são registradas todas as mensagens geradas pelas rotinas do programa, durante o processo iterativo. O nome desse arquivo pode ser alterado digitando-se um novo nome na caixa de texto correspondente. Neste exemplo, vamos aceitar todos os nomes de arquivos sugeridos e dar continuidade a simulação. No painel **Modo Estudo** há ainda uma caixa de texto onde o usuário pode fornecer um título para o estudo de caso sendo simulado. Se deixado em branco será utilizado o título fornecido no arquivo de dados da rede elétrica.

B.1.3 Configurando a Solução de Fluxo de Potência

Clicando-se no botão de acionamento da janela de trabalho **Fluxo de Potência**,



obtemos o acesso as funcionalidades dessa janela de trabalho, como ilustra a Figura B.8. Na janela **Fluxo de Potência** podemos selecionar o algoritmo de solução de fluxo de potência (o algoritmo padrão é o método desacoplado rápido, versão XB), e ajustar diversos parâmetros do algoritmo como o número máximo de iterações na solução de fluxo de potência, o resíduo máximo de balanço de potência ativa, o resíduo máximo de balanço de potência reativa, tolerância de convergência nos módulos de tensões, etc. Neste exemplo, vamos aceitar os valores padrões indicados na interface gráfica com o usuário.

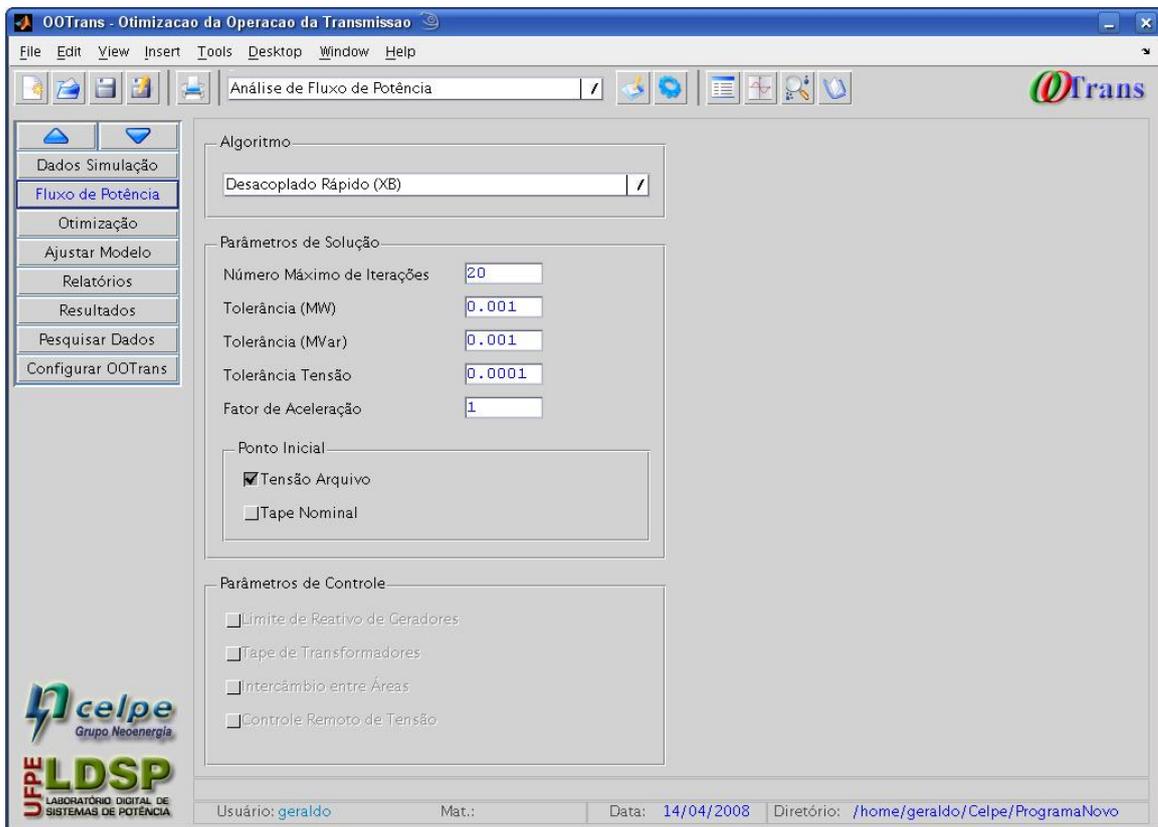


Figura B.8: Janela de trabalho **Fluxo de Potência**.

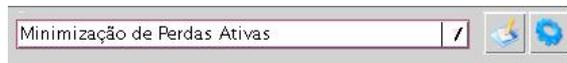
B.1.4 Configurando a Solução de Minimização de Perdas

Clicando-se no botão de acionamento da janela de trabalho **Otimização** devemos obter o acesso das funcionalidades relacionadas aos estudos de otimização como minimização de perdas elétricas e minimização de corte de carga. No entanto, de acordo as ações consideradas no presente exemplo até o momento, a imagem que obtemos é a de uma pasta vazia. Nada de anormal, pois até o momento o tipo de estudo indicado no *Popup Menu* da barra de ferramentas é ainda o de uma simples solução de fluxo de potência,



em vez de minimização de perdas elétricas ou minimização de corte de carga. A janela de trabalho **Otimização** somente exibirá as suas funcionalidades quando o problema selecionado for um problema de otimização.

Fazendo a seleção no *Popup Menu* da barra de ferramentas da opção *Minimizacao de Perdas Ativas*,



ativa automaticamente as funcionalidades da janela de trabalho **Otimização**, como mostra a Figura B.9.

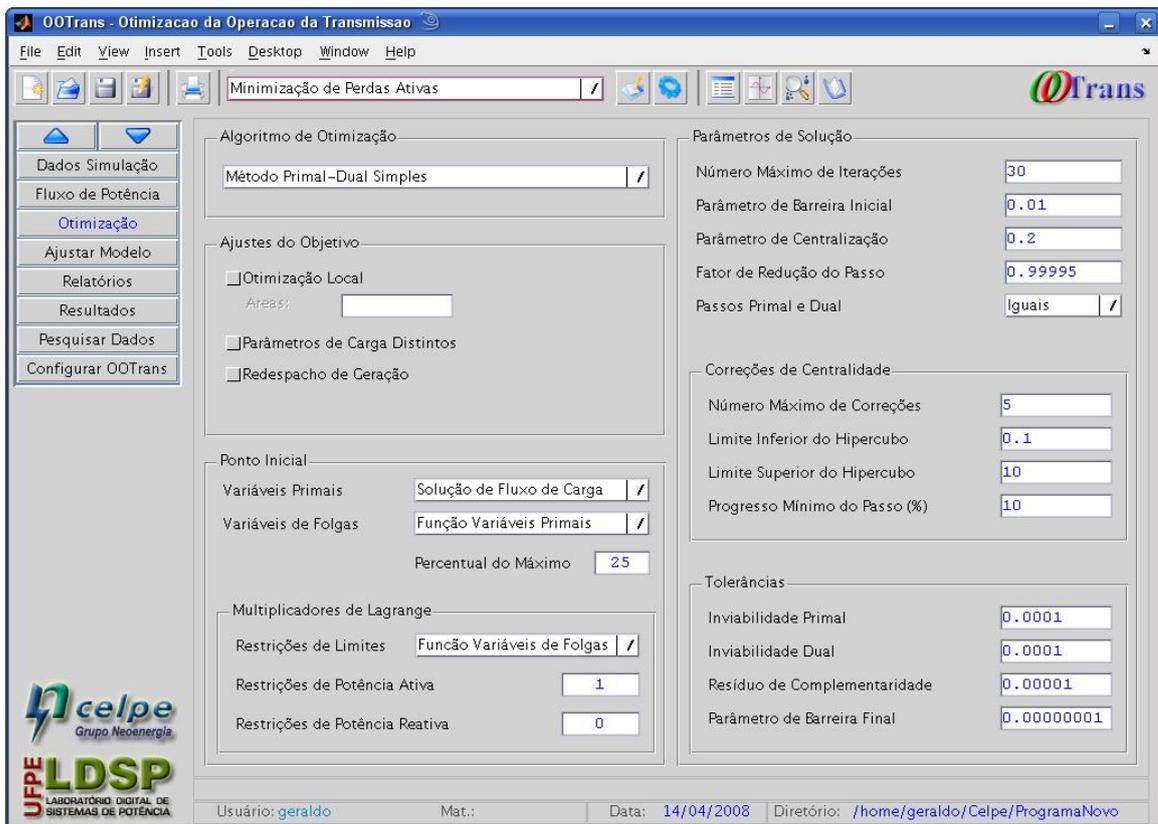


Figura B.9: Janela de trabalho **Otimização**.

Na janela de trabalho **Otimização** temos o total controle do processo de otimização, desde a seleção do algoritmo de pontos-interiores até o ajuste de praticamente todos os parâmetros do algoritmo selecionado. Os algoritmos disponíveis no **OOTrans** são:

1. método primal-dual simples,
2. método primal-dual preditor-corretor,

3. método primal-dual preditor-corretor múltiplo, e
4. método primal-dual com múltiplas correções de centralidade.

Os algoritmos listados estão em ordem crescente de robustez numérica, e o algoritmo padrão é o mais simples dos quatro: o método primal-dual simples. O usuário tem total controle sobre todos os parâmetros dos algoritmos de solução. Os valores de parâmetros padrões sugeridos na interface provavelmente terão sucesso na otimização de qualquer sistema, e raramente o usuário terá que ajustar algum parâmetro para obter sucesso na convergência.



No painel **Ponto Inicial** podemos atuar na escolha do ponto inicial do algoritmo de otimização. Observe, da Figura B.10, que o padrão é iniciar as variáveis x conforme uma solução inicial de fluxo de potência, as variáveis de folgas s e z em função da escolha das variáveis x , e as variáveis duais π e v em função da escolha das variáveis de folgas. No caso dos multiplicadores de Lagrange das restrições de igualdades, os mesmos podem ser iniciados com valores distintos por grupos de restrições: restrições de balanço de potência ativa ($\lambda = 1$) e restrições de balanço de potência reativa e fluxos ($\lambda = 0$).

No painel **Parâmetros de Solução** podemos ajustar praticamente todos os parâmetros do algoritmo de otimização selecionado. Devido a significativa robustez dos algoritmos de pontos-interiores implementados, os valores padrões de parâmetros apresentados na interface gráfica com o usuário provavelmente terão sucesso na otimização dos diversos sistemas sem a necessidade de ajustes diferenciados para cada sistema analisado. Podemos avançar para o passo seguinte, sem realizar qualquer modificação nos valores padrões que são apresentados na pasta **Otimização**.

B.1.5 Ajustando o Modelo de Otimização

Uma janela de trabalho com funcionalidades muito interessantes do **OOTrans** é a janela intitulada **Ajustar Modelo**, como ilustra a Figura B.11. Nesta janela o usuário poderá

Ponto Inicial

Variáveis Primais Solução de Fluxo de Carga /

Variáveis de Folgas Função Variáveis Primais /

Percentual do Máximo 25

Multiplicadores de Lagrange

Restrições de Limites Função Variáveis de Folgas /

Restrições de Potência Ativa 1

Restrições de Potência Reativa 0

Figura B.10: Painel para escolha do ponto inicial.

simular diversas modificações no problema de otimização sem ter que editar arquivos de dados.

No painel **Ajustes via Arquivo?** o usuário poderá informar o nome de um arquivo que contém as modificações desejadas, como inclusão e/ou exclusão de variáveis de controles, alterações de limites, inclusão e/ou exclusão de restrições de limites, etc.

Ajustes via Arquivo?

Sim

Não

No painel **Ajustes** o usuário poderá dar início a informação das modificações via a interface.

Ajustes

Grupo /

Sub-Grupo /

Neste painel escolhe-se inicialmente o grupo das modificações desejadas, ou seja, modificações de *controles* ou de *restrições*. Escolhido o grupo (controles ou restrições) o usuário

Parâmetros de Solução	
Número Máximo de Iterações	30
Parâmetro de Barreira Inicial	0.01
Parâmetro de Centralização	0.2
Fator de Redução do Passo	0.99995
Passos Primal e Dual	Iguais /

Correções de Centralidade	
Número Máximo de Correções	5
Limite Inferior do Hiper-cubo	0.1
Limite Superior do Hiper-cubo	10
Progresso Mínimo do Passo (%)	10

Tolerâncias	
Inviabilidade Primal	0.0001
Inviabilidade Dual	0.0001
Resíduo de Complementaridade	0.00001
Parâmetro de Barreira Final	0.00000001

escolhe a seguir o subgrupo, cujo lista de opções depende do problema de otimização selecionado. A Figura B.11 ilustra a seleção do grupo **Controles** e do subgrupo **Tape**. Uma vez selecionado o subgrupo, os controles correspondentes a esse subgrupo são listados no campo a esquerda. Ao selecionarmos qualquer item dessa lista os dados correspondentes são exibidos no painel superior direito denominado **Item Selecionado**. Nas caixas de edição **Mínimo** e **Máximo** podemos alterar os valores de limites indicados.

Entre os dois campos de listagens temos os botões que efetuam a transferência dos itens selecionados de uma listagem para outra, ou seja, que excluem ou incluem os controles selecionados. A Figura B.12 ilustra a exclusão da variável de controle **tape** do circuito 40. Para excluir controles (ou restrições) seleciona-se os itens para exclusão na coluna da esquerda e clica-se na seta para direita:

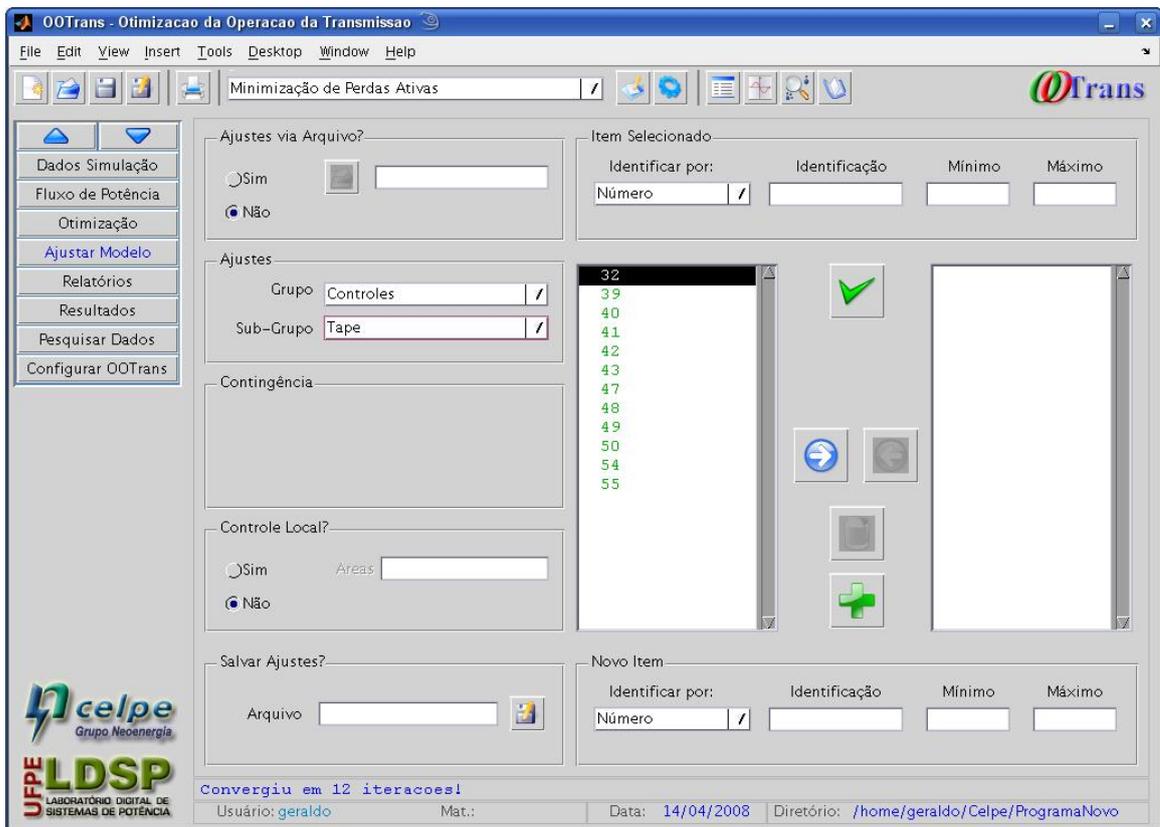


Figura B.11: Exemplo de ajuste nas variáveis de controle tape.



Para incluir controles (ou restrições) seleciona-se os ítems para inclusão na coluna da direita e clica-se na seta para esquerda



que estará ativa quando são selecionados itens na coluna direita. Observe das imagens apresentadas que vários outros recursos podem ser acionados no painel **Ajustes Modelo**, como o controle local, indicação de contingência, gerar arquivo de ajustes, etc.

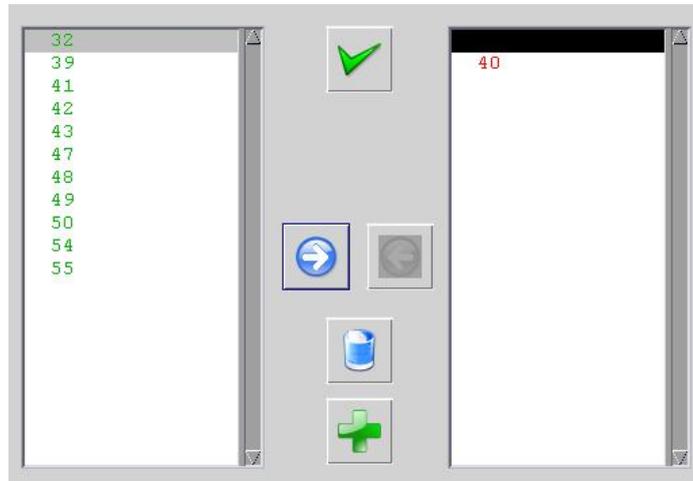
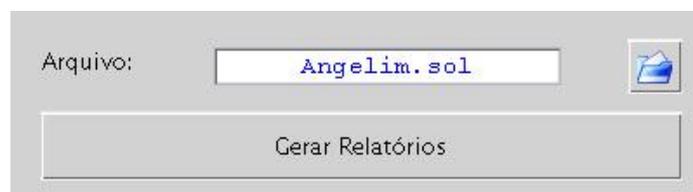


Figura B.12: Exemplo da exclusão da variável de controle tape do circuito 40.

B.1.6 Selecionando Relatórios para Impressão

Clicando-se no botão de acionamento da janela de trabalho **Relatórios** obtemos acesso a uma lista de *Checkbox's* onde podemos selecionar diversas modalidades de relatórios, desde relatórios usuais de solução de fluxo de potência, a relatórios diversos relacionados a solução do problema de otimização. A lista de relatórios selecionáveis é como ilustra a Figura B.13.

A seleção padrão é a de apenas 3 relatórios relacionados a solução de fluxo de potência, a saber: (1) sumário do sistema, (2) dados de barra na solução, e (3) fluxos nos circuitos. Os relatórios no painel **Relatórios de Fluxo de Potência** são gerados automaticamente apenas quando o estudo é de fluxo de potência. Se o estudo for de otimização então os relatórios são gerados apenas quando acionado o botão *Gerar Relatórios* na parte inferior do painel:



Como a funcionalidade do botão *Gerar Relatórios* um relatório qualquer pode ser solicitado tanto antes quanto após a execução da simulação.

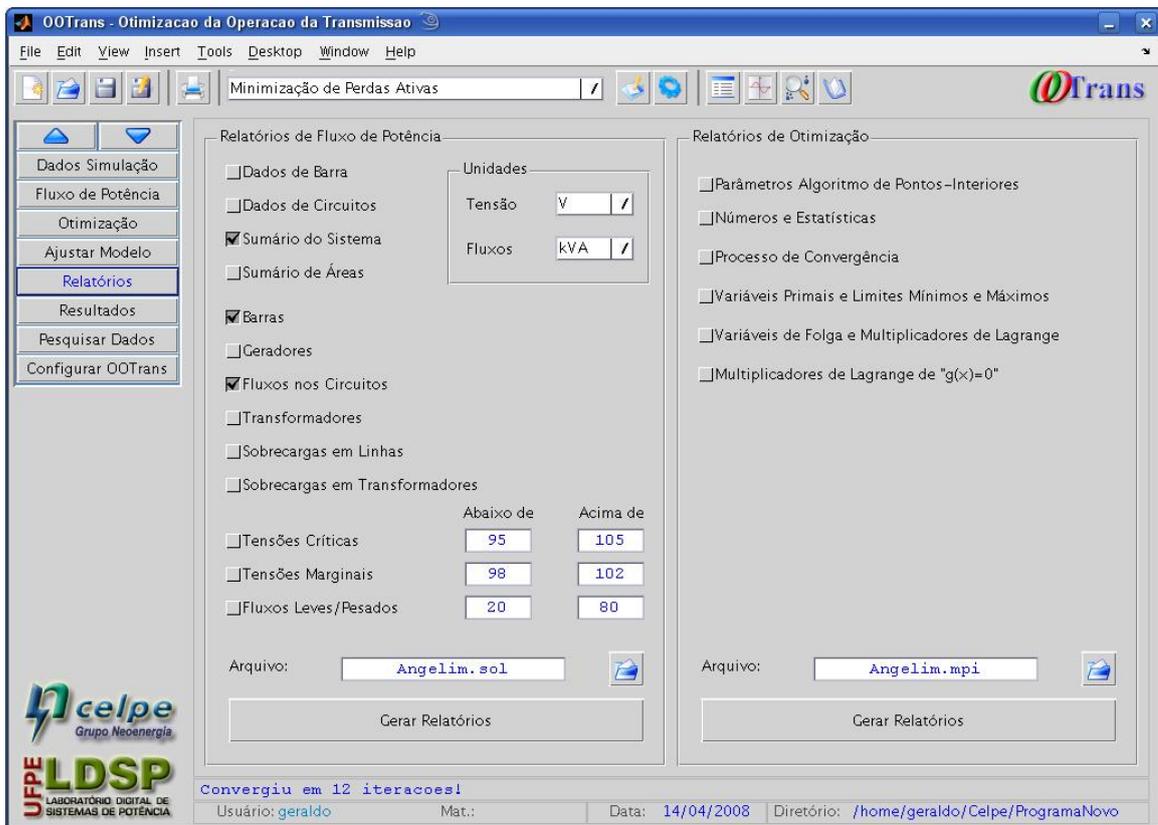


Figura B.13: Janela de trabalho para seleção de relatórios.

Ressalta-se que o painel **Relatórios de Otimização** é exibido apenas quando o estudo selecionado na barra de ferramentas é um problema de otimização, ou seja, minimização de perdas elétricas ou minimização de corte de carga. As informações contidas nestes relatórios são de grande utilidade na análise pós-otimalidade, como a identificação de restrições críticas e quais os “gargalos” que impedem uma maior otimização da operação do sistema.

B.1.7 Executando a Otimização

Observe que, até então, ao aceitarmos os valores padrões informados na interface gráfica com o usuário, a única ação efetiva que realizamos na interface gráfica do programa foi a informação do arquivo de dados da rede elétrica, e a seleção no *Popup Menu* da barra de ferramentas do problema de estudo como sendo *Minimização de Perdas Ativas*.

Desde o instante em que os arquivos de dados da rede elétrica foram definidos que o botão de execução do algoritmo de solução,

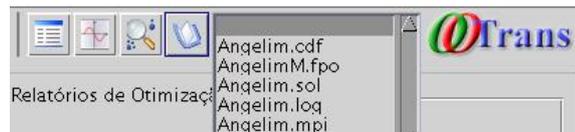


encontra-se habilitado na interface gráfica do programa. Clicando-se neste botão o processo de solução numérica é desencadeado, sendo exibidas, a cada iteração, na parte inferior da janela principal, as informações básicas do processo iterativo de solução, como o número da iteração corrente, os comprimentos de passos primal e dual, a inviabilidade primal, a inviabilidade dual, o resíduo de complementaridade, o parâmetro de barreira e o valor corrente da função objetivo, como ilustra a imagem abaixo.

2	0.5840	0.5840	6.686e-04	5.558e-02	4.559e-02	5.586e-04	1.008e+02
Usuário: geraldo	Mat.:	Data: 14/04/2008	Diretório: /home/geraldo/Celpe/ProgramaNovo				

B.1.8 Visualizando Resultados

Os resultados podem ser visualizados e analisados de diversas maneiras. Uma delas é a inspeção dos arquivos gerados, através da chamada automática de um editor de texto do sistema operacional. Isto pode ser feito simplesmente clicando no último botão da barra de ferramentas:



Clicando-se no botão



surge naturalmente a lista de todos os arquivos envolvidos na simulação, desde os arquivos de dados aos arquivos que foram gerados durante a simulação. Clicando-se no nome de um arquivo da lista o mesmo será automaticamente aberto por um editor de texto. Um clique no primeiro item da lista apenas carrega o editor de texto sem abrir qualquer arquivo.

Outros recursos para visualização de resultados são disponíveis nas janelas de trabalho **Resultados** e **Pesquisar Dados**. As janelas de trabalho **Resultados** e **Pesquisar Dados** podem ser rapidamente acionadas a partir da barra de ferramentas, clicando-se nos botões *resultados*,



e *pesquisar*,



B.1.9 Analisando a Solução

Outro recurso bastante prático e interessante do programa **OOTrans** é a facilidade de pesquisa de dados disponível na janela de trabalho **Pesquisar Dados**. A Figura B.14 ilustra os recursos disponíveis na janela de trabalho **Pesquisar Dados**. Primeiro, no painel **Objeto da Pesquisa** o usuário seleciona se deseja analisar em detalhes uma barra ou um circuito do sistema. A seleção padrão é a análise de barras. Uma vez feita essa seleção, identifica-se o objeto específico a ser analisado, ou seja, deve ser fornecido o número (externo) da barra ou o número do circuito. Feita a identificação, automaticamente são exibidos todos os dados, tanto da rede elétrica quanto do processo de otimização, relacionados com aquele objeto (barra ou circuito).

A Figura B.14 ilustra a pesquisa sobre a barra 1 do regional Angelim. No painel **Dados da Otimização** são exibidos todos os dados referentes ao processo de otimização, como valores das variáveis de folgas das restrições de limites e os valores dos multiplicadores de Lagrange associados aos respectivos limites. Esses dados são extremamente importantes para a análise pós-otimalidade onde são identificados os *gargalos* operativos do sistema, ou seja, quais restrições foram mais impeditivas para melhoria adicional do ponto de operação.

São exibidos também todos os dados relativos ao ponto de operação e os limites operativos considerados na solução do problema de otimização. No canto direito inferior é exibido um extrato do relatório de fluxo de potência envolvendo aquela barra, ou seja, os fluxos de potência em todos os circuitos que tem a barra pesquisada como um de seus terminais.

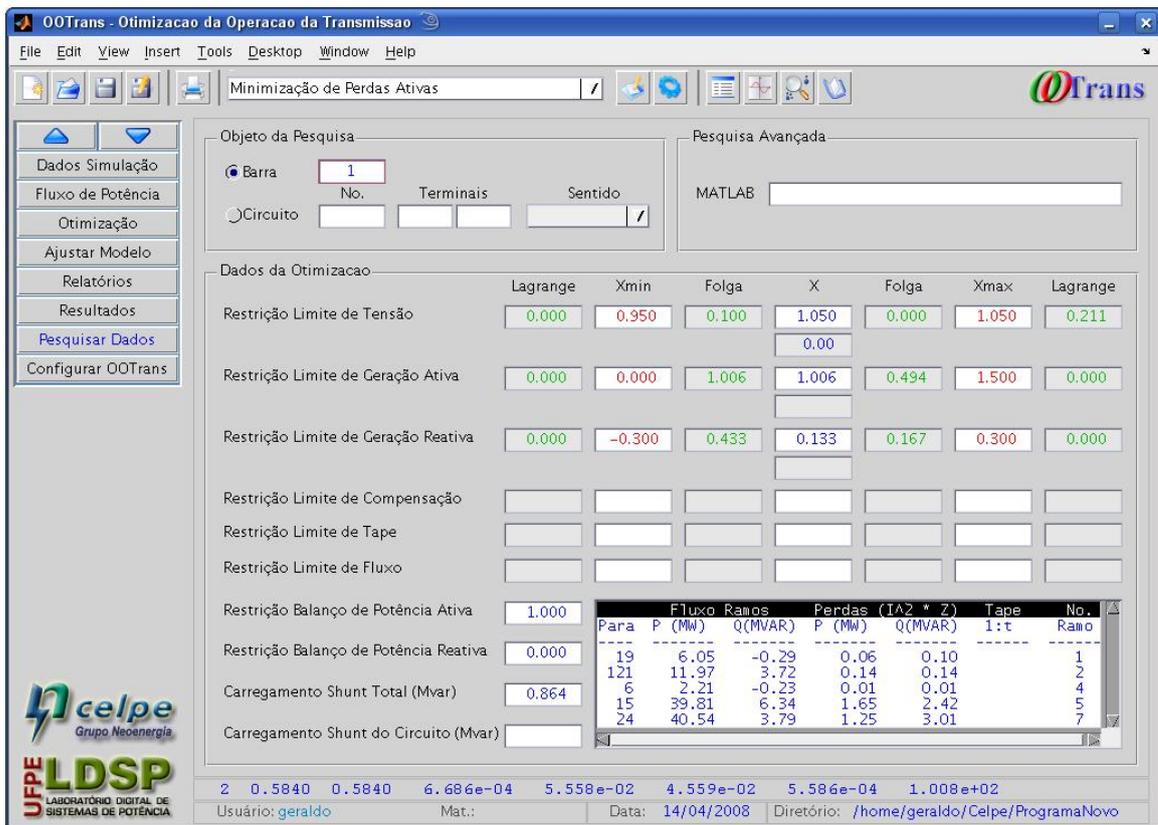


Figura B.14: Exemplo de pesquisa da barra 1 do regional Angelim.

Um recurso adicional existente no **OOTrans** que inexistente nos programas de otimização existentes no mercado é o da **Pesquisa Avançada**. Este painel disponibiliza para o usuário o *prompt* do MATLAB para a pesquisa de dados do processo de solução, como busca de valores máximos e mínimos, em que barras ou circuitos esses valores extremos ocorreram, exibição de dados diversos, etc.

B.2 As Rotinas Computacionais

O programa **OOTrans** é constituído por 40 rotinas computacionais, sendo 38 *function's* do MATLAB, também chamadas de arquivos-M, e 2 programas executáveis chamados de dentro do MATLAB mas que são desenvolvidos na linguagem Fortran. Os arquivos-M, em ordem alfabética, são:

acesso.m	gradiente.m	mpicorrcent.m
barreira.m	hessiana.m	mpimultcorr.m
config.m	jacobfluxos.m	mpinewtcomp.m
convergencia.m	jacobrestricoes.m	mpipredcorr.m
fluxodc.m	jacobsbarra.m	novoponto.m
fluxos.m	leituracdf.m	ootrans.m
fpcompleto.m	leiturafpo.m	passoalfa.m
fpdesacoplado.m	leitura.m	ponto inicial.m
fpgaussseidel.m	leiturapwf.m	principal.m
fpnewton.m	logindlg.m	restricoes.m
fprelatorios.m	montabbarra.m	retornofpo.m
fpresiduo.m	montaybarra.m	vetorkkt.m

ootrans

Este arquivo pode ser classificado como o principal do programa, uma vez que é ele quem carrega o **OOTrans** para execução. O arquivo `ootrans.m`, juntamente com o arquivo de mesmo nome e terminação `.fig`, `ootrans.fig`, gera a interface gráfica com o usuário do programa. O arquivo `ootrans.m` contém também a programação dos *Callback's* de todos os objetos da interface gráfica, ou seja, `ootrans.m` contém uma coleção de funções MATLAB que determinam as ações que são executadas quando cada objeto da interface é ativado.

O arquivo `ootrans.fig` é um arquivo binário que pode ser editado apenas com a ferramenta GUIDE (Graphical User Interface Development Environment) do MATLAB.

principal

Após a função `ootrans`, que é responsável pela interface gráfica e pela funcionalidade dos objetos, a função `principal` é a seguinte em nível hierárquico. É a função `principal` quem gerencia todo o processo de solução, extraindo da interface as informações necessárias para o correto fluxo do processamento, como os nomes dos arquivos de dados e de saídas, parâmetros dos algoritmos, seleção de algoritmos de fluxo de potência, seleção de algoritmos de otimização, tolerâncias de convergência, etc, e coordenando a chamada

de todas as rotinas para a solução do problema como especificado pelo usuário na interface.

As demais rotinas serão apresentadas na ordem em que elas são chamadas pela função principal.

leituracdf

Esta é a primeira função chamada pela função principal. Ela lê os dados de barras e de linhas do sistema elétrico no formato “IEEE Common Data Format” (*.cdf). A sintaxe de chamada da função é como segue:

```
[Barra,Linha,Lista,Indice,Geral] = leituracdf(Geral,handles)
```

A função `leituracdf` retorna na lista de argumentos de saída a estrutura `Barra` com os dados de barras do sistema, a estrutura `Linha` com os dados de linhas e transformadores, a estrutura `Lista` com vetores de listas de índices, e a estrutura `Índice` com os índices que são utilizados com maior frequência. A estrutura `Barra` apresenta os seguintes campos:

<i>Barra.campo</i>	Descrição
<code>numExt</code>	Número externo (batismo) da barra
<code>numSeq</code>	Número interno (sequencial) da barra
<code>nome</code>	Nome da barra
<code>area</code>	Área a que pertence a barra
<code>zona</code>	Zona a que pertence a barra
<code>tipo</code>	Tipo da barra
<code>tensao</code>	Tensão da barra (p.u.)
<code>angulo</code>	Ângulo de fase da tensão da barra (rad)
<code>cargaP</code>	Potência ativa da carga (p.u.)
<code>cargaQ</code>	Potência reativa da carga (p.u.)
<code>gerP</code>	Potência ativa gerada (p.u.)
<code>gerQ</code>	Potência reativa gerada (p.u.)
<code>baseKV</code>	Tensão base da barra (kV)
<code>tensaoRem</code>	Tensão da barra controlada (p.u.)
<code>maxQger</code>	Limite máximo da potência reativa gerada (p.u.)

minQger	Limite mínimo da potência reativa gerada (p.u.)
Gshunt	Condutância shunt da barra (p.u.)
Bshunt	Susceptância shunt da barra (p.u.)
Yshunt	Admitância shunt da barra (p.u.)
minVmod	Limite mínimo da tensão da barra (p.u.)
maxVmod	Limite máximo da tensão da barra (p.u.)
statusG	Status da geração (0-off, 1-on)
statusC	Status da carga (0-off, 1-on)
minPger	Limite mínimo da potência ativa (p.u.)
maxPger	Limite máximo da potência ativa (p.u.)
minBsht	Limite mínimo da susceptância shunt (p.u.)
maxBsht	Limite máximo da susceptância shunt (p.u.)
minSigm	Limite mínimo do parâmetro de carga
maxSigm	Limite máximo do parâmetro de carga
custoA	Coefficiente termo constante da curva quadrática
custoB	Coefficiente termo linear da curva quadrática
custoC	Coefficiente termo quadrático da curva quadrática
zipAp	Coefficiente potência ativa constante modelo ZIP da carga
zipBp	Coefficiente potência ativa constante modelo ZIP da carga
zipCp	Coefficiente potência ativa constante modelo ZIP da carga
zipAq	Coefficiente potência ativa constante modelo ZIP da carga
zipBq	Coefficiente potência ativa constante modelo ZIP da carga
zipCq	Coefficiente potência ativa constante modelo ZIP da carga
sigma	Coefficiente potência ativa constante modelo ZIP da carga

A estrutura Linha apresenta os seguintes campos:

Linha.campo	Descrição
barraDe	Número da barra terminal “de”
barraPa	Número da barra terminal “para”
area	Área a que pertence o circuito
circuito	Número do circuito

tipo	Tipo de circuito
Rserie	Resistência série do circuito (p.u.)
Xserie	Reatância série do circuito (p.u.)
Yshunt	Susceptância do carregamento shunt da linha
maxFlux	Fluxo máximo do circuito
tape	Tape do transformador
angTape	Ângulo de defasagem do transformador defasador
minTape	Limite mínimo do tape
maxTape	Limite máximo do tape
Yserie	Admitância série do circuito (p.u.)
fluxoPde	Fluxo ativo saindo da barra “de”
fluxoQde	Fluxo reativo saindo da barra “de”
fluxoSde	Fluxo aparente saindo da barra “de”
fluxoPpara	Fluxo ativo saindo da barra “para”
fluxoQpara	Fluxo reativo saindo da barra “para”
fluxoSpara	Fluxo aparente saindo da barra “para”
minDefa	Limite mínimo da defasagem do transformador
maxDefa	Limite máximo da defasagem do transformador

As demais estruturas (Lista e Indice), por não estarem completas após a execução da função `leituracdf`, são descritas na apresentação de outras funções.

montaybarra

A função `montaybarra` monta a matriz admitância de barra e as matrizes admitâncias “ramo-barra de” e “ramo-barra para”. A sintaxe de chamada da função é da forma:

```
[Ybarra, Yde, Ypara, Barra, Linha] = ...
                                montaybarra(Barra, Linha, Lista, Indice)
```

A função retorna nos argumentos de saída a matriz admitância de barra, `Ybarra`, e duas matrizes de admitâncias ramo-barra, `Yde` e `Ypara`, as quais, quando multiplicadas pelos

vetores das tensões complexas das barras “de” e “para” dos circuitos, respectivamente, fornecem os vetores das correntes injetadas em cada linha das barras “de” e “para”.

fpnewton

O programa **OOTrans** implementa os principais algoritmos para cálculo de fluxo de potência: o método de Newton completo, os métodos desacoplado rápido (versões BX e XB), o método de Gauss-Seidel, e o método DC. A função `fpnewton` calcula a solução do fluxo de potência pelo método de Newton completo. A sintaxe de chamada da função é:

```
[Barra,iIteracao,convergiu] = ...  
    fpnewton(Barra,Linha,Ybarra,Lista,Indice,handles)
```

A função `fpnewton` retorna, na estrutura `Barra`, as tensões de barras calculadas, o número de iterações, e um indicador de caso convergido ou não.

A função `fpnewton` faz a chamada, em cada iteração de fluxo de potência, da função `jacobsbarra`, a qual calcula o Jacobiano (matriz de derivadas primeiras) das funções injeções líquidas de potências nas barras.

fpdesacoplado

A função `fpdesacoplado` calcula a solução do fluxo de potência pelo método desacoplado rápido, tanto na versão BX quanto na versão XB. A sintaxe de chamada da função é:

```
[Barra,iIteracao,convergiu] = ...  
    fpdesacoplado(Barra,Linha,Ybarra,Lista,Indice,Geral,handles)
```

A função `fpdesacoplado` retorna, na estrutura `Barra`, as tensões de barras calculadas, o número de iterações, e um indicador de caso convergido ou não. Ela faz a chamada, em cada iteração de fluxo de potência, das funções `fpresiduo` e `montaybarra`. A finalidade da função `fpresiduo` é calcular os resíduos de potência nas barras normalizados pelos módulos das tensões de barras, conforme o equacionamento do fluxo de potência desacoplado.

A função `fpdesacoplado` identifica qual a versão do algoritmo desacoplado utilizar, se BX ou XB, na interface gráfica (pasta **Fluxo de Potência**) do programa **OOTrans**.

fpgaussseidel

A função `fpgaussseidel` calcula a solução de fluxo de potência pelo método de Gauss-Seidel. A sintaxe de chamada da função é:

```
[Barra, iIteracao, convergiu] = ...  
                                fpgaussseidel(Barra, Ybarra, Lista, handles)
```

O uso desse algoritmo é desaconselhado, uma vez a convergência é bastante lenta. A função `fpgaussseidel` retorna, na estrutura `Barra`, as tensões de barras calculadas, o número de iterações, e um indicador de caso convergido ou não.

fluxodc

A função `fluxodc` calcula a solução de fluxo de potência pelo método DC. A sintaxe de chamada da função é:

```
[Barra, Linha] = fluxodc(Barra, Linha, Lista, Indice)
```

Essa função calcula os ângulos de fase das tensões complexas de barras, exceto da barra de referência. Ela faz a chamada da função `montabbarra` para montar as matrizes de susceptâncias `Bbarra` e `Bde`, e o vetor das injeções de potência ativa pelos transformadores defasadores, para a solução do fluxo de potência DC.

fpcompleto

A função `fpcompleto` atualiza as estruturas de dados `Barra` e `Linha` com a solução do fluxo de potência. Mais especificamente, ela utiliza as tensões complexas das barras que foram calculadas pelo algoritmo de fluxo de potência para calcular a potência reativa das barras PV, a potência ativa da barra $V\theta$, e os fluxos de potência ativa e reativa nos circuitos. A sintaxe de chamada da função `fpcompleto` é da forma:

```
[Barra,Linha] = fpcompleto(Barra,Linha,Ybarra,Yde,Ypara,Lista)
```

fprelatorios

A função `fprelatorios` é uma função que trata especificamente da geração de relatórios de dados e da solução de fluxo de potência. Os relatórios a serem gerados são selecionados na interface gráfica do programa **OOTrans**, na pasta **Relatórios**. Por ser uma função de saída de dados, a sintaxe de chamada apresenta apenas argumentos de entrada, como segue:

```
fprelatorios(Barra,Linha,Lista,Indice,Geral,handles)
```

leiturfpo

A função `leiturfpo` faz a leitura dos dados da otimização da rede elétrica, os quais estão organizados num arquivo específico de terminação `.fpo`, segundo uma formatação própria do programa **OOTrans**, pois não há na literatura uma padronização definida para a formatação desses dados. Por dados da otimização entende-se os limites mínimos e máximos das variáveis, tais como os módulos das tensões, tapes, compensações shunts, fluxos nos ramos, etc, bem como a indicação das variáveis de controle como tapes, compensações shunts, etc. Os parâmetros dos algoritmos de otimização são especificados na interface gráfica do programa, com valores padrões definidos na inicialização do programa.

A sintaxe de chamada da função `leiturfpo` é a seguinte:

```
[Barra,Linha,Lista,Indice] = ...  
leiturfpo(Barra,Linha,Lista,Indice,Geral,handles)
```

Ela retorna os dados da otimização nas estruturas `Barra` e `Linha`, define algumas listas de índices na estrutura `Lista` e algumas quantidades na estrutura `Indice`. As estruturas de dados `Lista` e `Indice` são descritas abaixo.

ponto inicial

A finalidade da função `ponto inicial` é a inicialização eficiente dos algoritmos de otimização, ou seja, o cálculo de um ponto inicial para o processo iterativo, conforme opções

do usuário feitas na interface gráfica (na pasta **Otimização**). A sintaxe de chamada da função `leiturafpo` é a seguinte:

```
[Barra,Linha,Ybarra,Ytape,Vbarra,Fpo,Lista,Indice] = ...
pontoinicial(Barra,Linha,Ybarra,Yde,Ypara,Lista,Indice,Geral,handles)
```

Dependendo da regra indicada na interface gráfica para inicialização das variáveis primais x , s e z , a função `pontoinicial` pode fazer chamadas das funções `restricoes` e `gradiente`. É nesta função que são definidos os vetores de limites mínimos e máximos das variáveis x , l e u , bem como são criadas diversas listas de índices e de números, nas estruturas `Lista` e `Indice`. Os campos da estrutura `Lista` são:

<code>Lista.campo</code>	Descrição
<code>barraPQ</code>	Lista de barras tipo PQ
<code>barraPV</code>	Lista de barras tipo PV
<code>barraVT</code>	Lista de barras tipo $V\theta$
<code>barPQPV</code>	Lista de barras tipos PQ e PV
<code>tape</code>	Lista de transformadores
<code>varQ</code>	Lista de variáveis potência reativa gerada Q_G
<code>varB</code>	Lista de variáveis susceptância shunt b_{sh}
<code>trafos</code>	Lista de variáveis t_{ij} e/ou ϕ_{ij}
<code>varT</code>	Lista de variáveis <code>tape</code> t_{ij}
<code>varD</code>	Lista de variáveis ângulo de defasagem ϕ_{ij}
<code>varF</code>	Lista de variáveis fluxo F_{ij}
<code>varP</code>	Lista de variáveis potência ativa gerada P_G
<code>varS</code>	Lista de variáveis corte de carga σ
<code>varA</code>	Lista de variáveis de ângulo de fase θ
<code>varV</code>	Lista de variáveis módulo de tensão V
<code>fx2tm</code>	uso interno
<code>fx2ta</code>	uso interno
<code>rm2tm</code>	uso interno
<code>rm2ta</code>	uso interno
<code>yVmod</code>	uso interno

Os campos da estrutura Índice são:

Lista.campo	Descrição
nBarras	Número de barras do sistema
nSwing	Número de barras de balanço
nRamos	Número de circuitos
nTapes	Número de transformadores com LTC
nTrafo	Número de transformadores com LTC e defasadores
nDefa	Número de defasadores
nFluxo	Número de variáveis fluxo
eMultiG	1=res despacho de geração
eMultiC	1=múltiplos fatores de corte de carga
nVang	Número de variáveis θ
nVmod	Número de variáveis V
nPger	Número de variáveis P_G
nQger	Número de variáveis Q_G
nBsht	Número de variáveis b_{sh}
nSigma	Número de variáveis corte de carga σ
ixVmod	uso interno
ixTape	uso interno
ixDefa	uso interno
ixPger	uso interno
ixQger	uso interno
ixBsht	uso interno
ixFluxo	uso interno
ixSigma	uso interno
nX	Número de variáveis de decisão x
nXlim	Número de variáveis sujeitas a limites simples
igQ	uso interno
igF	uso interno
nG	Número de restrições de igualdades
iyVang	uso interno

iyVmod	uso interno
iyTape	uso interno
iyDefa	uso interno
iyPger	uso interno
iyQger	uso interno
iyBsht	uso interno
iyFluxo	uso interno
iySigma	uso interno
nY	Número total de variáveis (dimensão do sistema de Newton)
nIteracoes	Número máximo de iterações

Os campos da estrutura **Fpo** são:

Fpo.campo	Descrição
mi	Parâmetro de barreira μ
reduzmi	Parâmetro de centralização σ
alfa0	Fator de redução do comprimento de passo α_0
regraPasso	uso interno
nCorrecoes	Número máximo de passos de correção
minBeta	Limite inferior do hiper-cubo β_{\min}
maxBeta	Limite superior do hiper-cubo β_{\max}
tolPrimal	Tolerância de convergência da inviabilidade primal
tolDual	Tolerância de convergência da inviabilidade dual
tolComp	Tolerância de convergência do resíduo de complementaridade
tolMi	Tolerância do parâmetro de barreira
Xmin	Limite mínimo das variáveis
Xmax	Limite máximo das variáveis
funObj	Valor da função objetivo
restrG	Vetor de restrições $g(x_k)$
restrX	Valor das variáveis sujeitas a limites x_k
folgaXmin	Variáveis de folgas do limite inferior $x - s = \underline{x}$
folgaXmax	Variáveis de folgas do limite superior: $x + z = \bar{x}$

gradF	Gradiente da função objetivo $\nabla f(x_k)$
mlagrXmin	Multiplicador de Lagrange da restrição de limite inferior π
mlagrXmax	Multiplicador de Lagrange da restrição de limite superior ν
mlagrG	Multiplicador de Lagrange das restrições de igualdades λ
invPrimal	Inviabilidade primal
invDual	Inviabilidade dual
alfaP	Comprimento de passo primal α_p
alfaD	Comprimento de passo dual α_d
alfaPeq	Número máximo de iterações com passos muito pequenos
registro	uso interno
jacobG	Jacobiano das restrições de igualdades $\nabla g(x_k)^T$
gapXmin	Resíduo de complementaridade das restrições de limites mínimos
gapXmax	Resíduo de complementaridade das restrições de limites máximos
eqKKT	Vetor independente do sistema de Newton
convergiu	uso interno
deltaY	Vetor direção de busca Δy
iPrimal	uso interno
iDual	uso interno

jacobsbarra

A função `jacobsbarra` é de grande importância tanto no algoritmo de fluxo de potência quanto nos algoritmos de otimização. Ela calcula as derivadas parciais das injeções de potência complexas nas barras com relação aos módulos e ângulos das tensões. Opcionalmente, quando chamada com quatro argumentos de saída, retorna as derivadas parciais das injeções de potência com relação aos tapes e ângulos dos transformadores. A sintaxe de chamada da função é como segue:

```
[dSdV, dSdA, dSdTm, dSdTa] = ...
    jacobsbarra(Linha, Ybarra, Vbarra, Lista, Indice)
```

A matriz complexa dSdV contém as derivadas parciais

$$\frac{\partial \vec{S}_i}{\partial V_j} = \frac{\partial P_i}{\partial V_j} + j \frac{\partial Q_i}{\partial V_j},$$

a matriz complexa dSdA contém as derivadas parciais

$$\frac{\partial \vec{S}_i}{\partial \theta_j} = \frac{\partial P_i}{\partial \theta_j} + j \frac{\partial Q_i}{\partial \theta_j},$$

a matriz complexa dSdTm contém as derivadas parciais

$$\frac{\partial \vec{S}_i}{\partial t_{ij}} = \frac{\partial P_i}{\partial t_{ij}} + j \frac{\partial Q_i}{\partial t_{ij}},$$

e a matriz complexa dSdTα contém as derivadas parciais

$$\frac{\partial \vec{S}_i}{\partial \phi_{ij}} = \frac{\partial P_i}{\partial \phi_{ij}} + j \frac{\partial Q_i}{\partial \phi_{ij}}.$$

jacobrestricos

A função `jacobrestricos` calcula o Jacobiano $\nabla g(x_k)^T$ do vetor de funções de restrições de igualdades:

$$g(x) = \begin{pmatrix} P_i(V, \theta, t) + P_{D_i}(V_i) - P_{G_i} & i \in \mathcal{N} \\ Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} & i \in \mathcal{G} \\ Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} & i \in \mathcal{F} \\ Q_i(V, \theta, t) + Q_{D_i}(V_i) - Q_{G_i} - b_i^{sh} V_i^2 & i \in \mathcal{C} \\ F_{ij}(V, \theta, t) - F_{ij} & \{(i, j)\} \subseteq \mathcal{B} \end{pmatrix} \quad (\text{B.1})$$

Nos algoritmos de otimização, o Jacobiano $\nabla g(x_k)^T$ e o seu transposto aparecem na composição do sistema de Newton:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta v \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S\pi - \mu_k e \\ Zv - \mu_k e \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla f(x) + \nabla g(x)\lambda - \pi + v \end{bmatrix} \quad (\text{B.2})$$

A sintaxe de chamada da função `jacobrestricoes` é da forma:

```
Fpo = jacobrestricoes(Barra,Linha,Yde,Vbarra,dSdA,dSdV,dSdTm, ...
                    dSdTd,Fpo,Lista,Indice,Geral)
```

A função retorna a matriz $\nabla g(x_k)^T$ no campo `Fpo.jacobG`. Observe, da lista de argumentos de entrada, que a função `jacobrestricoes` faz uso das matrizes complexas que são calculadas na função `jacobsbarra`. A função `jacobrestricoes` faz ainda uma chamada da função `jacobfluxos` para calcular as derivadas parciais das restrições de fluxos nos ramos.

barreira

A função `barreira` calcula os resíduos de complementaridade das restrições de limites sobre as variáveis e efetua a redução do parâmetro de barreira, segundo as expressões matemáticas:

$$\mu_{k+1} = \sigma \frac{s_k^T \pi_k + z_k^T u_k}{2p} \quad (\text{B.3})$$

A sintaxe de chamada da função `barreira` é da forma:

```
Fpo = barreira(Fpo,Indice)
```

Os resíduos de complementaridade e o parâmetro de barreira são retornados em campos da estrutura `Fpo`, a saber: `Fpo.gapXmin`, `Fpo.gapXmax` e `Fpo.mi`.

vetorkkt

A função `vetorkkt` monta o vetor independente do sistema de Newton, ou seja, o vetor gradiente da função de Lagrange:

$$\nabla_y L(y_k, \mu_k) = \begin{bmatrix} S\pi - \mu_k e \\ Zv - \mu_k e \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla f(x) + \nabla g(x)\lambda - \pi + v \end{bmatrix} \quad (\text{B.4})$$

O vetor gradiente $\nabla_y L(y_k, \mu_k)$ calculado retorna na estrutura `Fpo`, no campo `Fpo.eqKKT`. A sintaxe de chamada da função `vetorkkt` é da forma:

`Fpo = vetorkkt(Fpo, Indice, Geral)`

convergencia

Uma vez calculado o vetor $\nabla_y L(y_k, \mu_k)$ pela função `vetorkkt`, pode-se facilmente fazer o teste de convergência:

$$\max \left\{ \max_i \{ \underline{x}_i - x_i^k \}, \max_i \{ x_i^k - \bar{x}_i \}, \|g(x_k)\|_\infty \right\} \leq \epsilon_1, \quad (\text{B.5a})$$

$$\frac{\|\nabla f(x_k) + \nabla g(x_k)\lambda - \pi_k + v_k\|_\infty}{1 + \|x_k\|_2 + \|\lambda_k\|_2 + \|\pi_k\|_2 + \|v_k\|_2} \leq \epsilon_1, \quad (\text{B.5b})$$

$$\frac{\rho_k}{1 + \|x_k\|_2} \leq \epsilon_2. \quad (\text{B.5c})$$

Este teste é feito pela função `convergencia`, cuja sintaxe de chamada é como segue:

`[Fpo, Indice] = convergencia(Barra, Fpo, Indice, iIteracao, handles)`

O estado corrente do processo de convergência é exibido na parte inferior da interface do programa.

hessiana

Esta é a mais complexa das funções do programa **OOTrans**. A função **hessiana** monta a matriz Hessiana que compõe o sistema linear de Newton, ou seja, monta a matriz de coeficientes

$$\nabla_{yy}^2 L(y) = \begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \quad (\text{B.6})$$

A complexidade dessa tarefa vem do fato da matriz $\nabla_{xx}^2 L(y)$ resultar do somatório de um elevado número de matrizes Hessianas, segundo a fórmula:

$$\nabla_{xx}^2 L(y) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 g_i(x). \quad (\text{B.7})$$

A montagem dessa matriz e a solução do sistema de Newton juntos são responsáveis por cerca de 90% do tempo total de solução do problema de FPO. Portanto, a implementação eficiente dessa tarefa exige grande habilidade de programação. A sintaxe de chamada da função **hessiana** é como segue:

```
L2Y = hessiana(Barra, Linha, Ybarra, Vbarra, dSdA, dSdV, Fpo, Lista, ...
              Indice, Geral)
```

A matriz de coeficientes calculada retorna na variável de saída L2Y.

passoalfa

A função `passoalfa` calcula o comprimento de passo ao longo da direção Δy utilizando o teste de razões mínimas abaixo:

$$\alpha_k^P = \min \left\{ 1, \gamma \min_i \left\{ \frac{-s_i^k}{\Delta s_i} \middle| \Delta s_i < 0, \frac{-z_i^k}{\Delta z_i} \middle| \Delta z_i < 0 \right\} \right\}, \quad (\text{B.8a})$$

$$\alpha_k^D = \min \left\{ 1, \gamma \min_i \left\{ \frac{-\pi_i^k}{\Delta \pi_i} \middle| \Delta \pi_i < 0, \frac{-v_i^k}{\Delta v_i} \middle| \Delta v_i < 0 \right\} \right\}. \quad (\text{B.8b})$$

A sintaxe de chamada da função `passoalfa` é da forma:

`Fpo = passoalfa(Fpo,Indice)`

Os comprimentos de passos retornam como os campos `Fpo.alfaP` e `Fpo.alfaD`. A função retorna ainda os índices das variáveis primais e duais que limitaram o comprimento do passo: `Fpo.iPrimal` e `Fpo.iDual`.

mpipredcorr

A função `mpipredcorr` executa as tarefas adicionais do algoritmo preditor-corretor de pontos-interiores. Ou seja, dada a direção de busca afim, Δy_{af} , ela calcula a estimativa do comprimento de passo se a direção afim fosse utilizada, conforme a regra

$$\alpha_{af}^P = \min \left\{ 1, \gamma \times \min_i \left\{ \frac{-s_i^k}{\Delta s_i^{af}} \middle| \Delta s_i^{af} < 0, \frac{-z_i^k}{\Delta z_i^{af}} \middle| \Delta z_i^{af} < 0 \right\} \right\}, \quad (\text{B.9a})$$

$$\alpha_{af}^D = \min \left\{ 1, \gamma \times \min_i \left\{ \frac{-\pi_i^k}{\Delta \pi_i^{af}} \middle| \Delta \pi_i^{af} < 0, \frac{-v_i^k}{\Delta v_i^{af}} \middle| \Delta v_i^{af} < 0 \right\} \right\}, \quad (\text{B.9b})$$

a seguir estima o parâmetro de barreira,

$$\rho_{af} = (s_k + \alpha_{af}^P \Delta s_{af})^T (\pi_k + \alpha_{af}^D \Delta \pi_{af}) + (z_k + \alpha_{af}^P \Delta z_{af})^T (v_k + \alpha_{af}^D \Delta v_{af}). \quad (\text{B.10})$$

$$\mu_{af} = \min \left\{ \left(\frac{\rho_{af}}{\rho_k} \right)^2, 0.2 \right\} \frac{\rho_{af}}{p}. \quad (\text{B.11})$$

e por fim, resolve o sistema linear para a direção de busca completa

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & Y & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I & I & \nabla g(x) & \nabla_{xx}^2 L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta v \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S\pi - \mu_{af}e + \Delta S_{af}\Delta\pi_{af} \\ Zv - \mu_{af}e + \Delta Z_{af}\Delta v_{af} \\ l + s - x \\ x + z - u \\ g(x) \\ \nabla f(x) + \nabla g(x)\lambda - \pi + v \end{bmatrix}. \quad (\text{B.12})$$

A sintaxe de chamada da função `mpipredcorr` é como segue:

`Fpo = mpipredcorr(Fpo,L2Y,Indice)`

mpimultcorr

A função `mpimultcorr` implementa os passos adicionais do método de pontos-interiores preditor-corretor com múltiplos passos de correção. A sintaxe de chamada da função é da forma:

`Fpo = mpimultcorr(Fpo,L2Y,Indice,handles)`

mpicorrcent

A função `mpicorrcent` implementa os passos adicionais do método de pontos-interiores com múltiplas correções de centralidade. A sintaxe de chamada da função é da forma:

`Fpo = mpicorrcent(Fpo,L2Y,Indice,handles)`

A função `mpicorrcent` é a única a fazer uso dos parâmetros de otimização no painel **Múltiplas Correções** ilustrado na figura abaixo.

Parâmetros de Solução	
Número Máximo de Iterações	30
Parâmetro de Barreira Inicial	0.01
Parâmetro de Centralização	0.2
Fator de Redução do Passo	0.99995
Passos Primal e Dual	Iguais /
Correções de Centralidade	
Número Máximo de Correções	5
Limite Inferior do Hiper cubo	0.1
Limite Superior do Hiper cubo	10
Progresso Mínimo do Passo (%)	10
Tolerâncias	
Inviabilidade Primal	0.0001
Inviabilidade Dual	0.0001
Resíduo de Complementaridade	0.00001
Parâmetro de Barreira Final	0.00000001

novoponto

A função novoponto faz a atualização das variáveis a cada iteração do algoritmo de pontos-interiores, conforme as expressões:

$$\begin{aligned}
 x_{k+1} &= x_k + \alpha_k^P \Delta x, \\
 s_{k+1} &= s_k + \alpha_k^P \Delta s, \\
 z_{k+1} &= z_k + \alpha_k^P \Delta z, \\
 \lambda_{k+1} &= \lambda_k + \alpha_k^D \Delta \lambda, \\
 \pi_{k+1} &= \pi_k + \alpha_k^D \Delta \pi, \\
 v_{k+1} &= v_k + \alpha_k^D \Delta v,
 \end{aligned}
 \tag{B.13}$$

Dentre as variáveis x , um procedimento especial é adotado para a atualização das variáveis tapes t_{ij} e ângulos de defasagens ϕ_{ij} . Este procedimento faz-se necessário porque no cálculo das grandezas de barras, como potências injetadas nas barras e suas derivadas, as variáveis t_{ij} e ϕ_{ij} aparecem implicitamente em elementos das matrizes admitância de barra e admitância ramo-barra. Portanto, a atualização dos tapes e das defasagens implicam também na atualização das referidas matrizes admitâncias. A sintaxe de chamada da função

novoponto é como segue:

```
[Barra,Linha,Ybarra,Yde,Ypara,Ytape,Fpo] = ...
    novoponto(Barra,Linha,Ybarra,Yde,Ypara,Ytape,Fpo,Lista,Indice)
```

restricoes

A função `restricoes` efetua o cálculo dos vetores de restrições no ponto corrente, ou seja, os vetores $g(x_k)$ e x_l^k . A função objetivo $f(x_k)$ também é calculada nessa função. A sintaxe de chamada da função é como segue:

```
[Fpo,Linha,Vbarra] = restricoes(Barra,Linha,Ybarra,Yde,Ypara, ...
                                Fpo,Lista,Indice,Geral,handles)
```

A função `restricoes` faz uso da função `fluxos` para calcular os fluxos nos circuitos, uma vez que alguns desses fluxos compõem o conjunto de restrições.

gradiente

A função `gradiente` calcula o vetor gradiente da função objetivo, $\nabla f(x_k)$. A sintaxe de chamada da função é da forma:

```
Fpo = gradiente(Barra,Fpo,Lista,Indice,Geral,handles)
```

O vetor gradiente retorna na estrutura `Fpo`, no campo `Fpo.gradF`.

retornofpo

A função `retornofpo` conclui a solução do FPO atualizando dados para a solução de fluxo de potência final. Mais especificamente, ela atualiza as cargas ativas e reativas das barras, no caso dos problemas de mínimo corte de carga e máximo carregamento, e atualiza a matriz admitância de barra de acordo com a solução ótima obtida. A sintaxe de chamada da função é como segue:

```
[Barra,Ybarra] = finalfpo(Barra,Ybarra,Lista,Indice,Geral)
```
