



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
Programa de Pós-Graduação em Engenharia Elétrica

**ELEMENTOS DE PROGRAMAÇÃO MATEMÁTICA:
APLICAÇÕES AO PROBLEMA DO FLUXO DE POTÊNCIA
ÓTIMO LINEARIZADO**

Emerson Eustáquio Costa

Belo Horizonte, novembro de 2006



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
Programa de Pós-Graduação em Engenharia Elétrica

Emerson Eustáquio Costa

**ELEMENTOS DE PROGRAMAÇÃO MATEMÁTICA:
APLICAÇÕES AO PROBLEMA DO FLUXO DE POTÊNCIA
ÓTIMO LINEARIZADO**

Dissertação de mestrado, apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Pontifícia Universidade Católica de Minas Gerais, como parte dos requisitos necessários para a obtenção do Grau de Mestre em Ciências em Engenharia Elétrica.

Orientador: Prof^o Dr. Luiz Danilo Barbosa Terra

Pontifícia Universidade Católica de Minas Gerais
Belo Horizonte
2006.

FICHA CATALOGRÁFICA
Elaborada pela Biblioteca da
Pontifícia Universidade Católica de Minas Gerais

C837e Costa, Emerson Eustáquio.
Elementos de programação matemática: aplicações ao problema do fluxo de potência ótimo linearizado / Emerson Eustáquio Costa. Belo Horizonte, 2006. 137f.

Orientador: Luiz Danilo Barbosa Terra
Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Engenharia Elétrica.
Bibliografia.

1. Programação (Matemática). 2. Programação linear. 3. Redes elétricas. I. Terra, Luiz Danilo Barbosa. II. Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 681.3.055

A Deus, luz eterna em meus caminhos.
A meus queridos pais que me apoiaram e
me conduziram por caminhos certos.

AGRADECIMENTOS

Primeiramente, agradeço a Deus pela minha existência.

A meus pais, pelo apoio, incentivo e confiança em minha vitória.

Ao meu Orientador, Prof. Dr. Luiz Danilo Barbosa Terra pelo apoio, paciência e direcionamento.

Aos meus Professores da PUC Minas – PPGEE pelos conhecimentos plantados, que me possibilitaram e permitiram a colheita.

Aos amigos que me ajudaram nesta caminhada.

À Universidade FUMEC pelo incentivo.

À minha querida esposa Grazielle Costa, pela paciência e dedicação.

"Qualidade nunca é um acidente;
ela é sempre o resultado de um
esforço inteligente."

John Rusk

RESUMO

Este trabalho buscou investigar aplicações de áreas da programação matemática, mais precisamente, programação linear aplicada ao problema do fluxo de potência ótimo linearizado, ou fluxo de potência ótimo de corrente contínua. A investigação envolveu o estudo de programação linear, o exame do tema complexidade de algoritmos e a análise do problema do fluxo de potência ótimo em suas diversas formulações. Aplicações de programação linear em Engenharia Elétrica, em particular, ao problema do fluxo de potência ótimo de corrente contínua, foram examinadas. Alguns problemas e exemplos em rede de energia foram formulados e apresentados.

Palavras-chave: Programação Matemática; Programação Linear; Fluxo de Potência em redes de Energia Elétrica; Fluxo de Potência Ótimo DC; Congestionamento.

ABSTRACT

This work aimed at investigating applications of mathematical programming, more specifically, linear programming to the linear optimal power flow problem or optimal direct current power flow problem. The investigation encompassed the study of linear programming, the examination of the theme algorithm complexity and the analysis of the problem of the optimal direct current power flow in its several formulations. Applications of linear programming in Electrical Engineering, in particular, to the optimal direct current power flow problem were appraised. The mathematical formulation of the problems was carried out and numerical examples using electrical energy networks presented.

Keywords: Mathematical Programming; Linear Programming; Power Flow in Electrical Energy Network; Optimal DC Power Flow; Congestion.

LISTA DE FIGURAS

Figura 1	Representação Gráfica de Conjunto Convexo e Não-Convexo	27
Figura 2	Representação do Ponto Interior	32
Figura 3	Método Simplex X Método de Pontos Interiores	34
Figura 4	Modelo RAM.....	47
Figura 5	Notação O – Complexidade Assintótica	48
Figura 6	Modelo Físico do Fluxo de Potência Ótimo	68
Figura 7	Sistema de Três Barras	86
Figura 8	Representação da Rede Elétrica do Exemplo 4.1	88
Figura 9	Sistema de Cinco Barras e Sete Linhas	100
Figura 10	Sistema de Seis Barras e Sete Linhas.....	116
Figura 11	Custo Unitário de Geração	117
Figura 12	Potência de Barra.....	118
Figura 13	Fluxos nas Linhas	119
Figura 14	Potência de Barra.....	121
Figura 15	Fluxos nas Linhas	121
Figura 16	Potência de Barra.....	123
Figura 17	Fluxos nas Linhas	123
Figura 18	Potência de Barra	125
Figura 19	Fluxos nas Linhas	126
Figura 20	Sistema IEEE14bus	127
Figura 21	Fluxos nas Linhas	130
Figura 22	Potência de Barra.....	130

LISTA DE TABELAS, QUADROS E GRÁFICOS

Tabela 1	Tamanho do Problema X Tempo de Execução	44
Gráfico 1	Funções Características de complexidade Temporal.....	51
Tabela 2	Número de Iterações para o Exemplo de Klee e Minty	60
Gráfico 2	Curvas de Custo dos Geradores Desdobrados	90
Tabela 3	Síntese dos Casos a serem Investigados	98
Tabela 4	Dados da Rede	99
Tabela 5	Dados da Barra	100
Quadro 1	Relatório de Análise de Sensibilidade Fornecido pelo LINDO	109
Quadro 2	Relatório do LIPSOL	111
Quadro 3	Relatório Fornecido pelo MatLab	113
Tabela 6	Número de Iterações dos Algoritmos	115
Tabela 7	Dados das Linhas para o Sistema de 6 barras e 7 Linhas	116
Tabela 8	Dados de Barra para o Sistema de 6 Barras e 7 Linhas	117
Tabela 9	Limites nas Variáveis de Controle.....	124
Tabela 10	Dados de Rede para o Sistema IEEE14bus.....	127
Tabela 11	Dados de Barra para o Sistema IEEE14bus	128

SIGLAS E ACRÔNIMOS

PPL	Problema de Programação Linear
PL	Programação Linear
FPO	Fluxo de Potência Ótimo
DE	Despacho Econômico
CC	Corrente Contínua
DC	Direct Current
MPI	Métodos de Pontos Interiores
SEP	Sistema Elétrico de Potência
KKT	Karush Kuhn Tucker
PD	Primal – Dual
RAM	Random Access Machine
GR	Gradiente
VM	Variáveis Métricas
pu	por unidade
<i>min</i>	Minimizar
s.a.	sujeito a
IEEE	Institute of Electrical and Electronics Engineers
LIPSOL	Linear Interior Points Solver
LINPROG	Linear Programming
n	Número de variáveis do sistema
L	Número de <i>bits</i> de entrada para o algoritmo

SUMÁRIO

Capítulo 1	Introdução	14
1.1	Considerações Iniciais.....	14
1.2	Relevância	15
1.3	Objetivo.....	16
1.4	Metodologia	16
1.5	Organização do texto.....	17
Capítulo 2	Programação Linear	18
2.1	Considerações Iniciais	18
2.2	Programação Linear	20
2.2.1	Características da Programação Linear.....	22
2.3	Modelagem.....	23
2.3.1	Modelagem em Programação Linear	23
2.3.1.1	Formulação do problema	24
2.3.1.2	Construção do modelo matemático do sistema	25
2.3.1.3	Cálculo da solução através do modelo	26
2.3.1.4	Teste do modelo e da solução	26
2.3.1.5	Estabelecimento de controles da solução	26
2.3.1.5	Implementação e acompanhamento	27
2.4	Teoremas da Programação Linear	27
2.4.1	Teorema 1	27
2.4.2	Teorema 2	28
2.4.3	Teorema 3	28
2.4.4	Teorema 4	28
2.5	O Método Simplex	28
2.6	Dualidade	29
2.6.1	Modelagem matemática	30
2.7	Análise de sensibilidade	30
2.8	O Método dos Pontos Interiores (MPI)	31
2.9	O Método de Newton na construção de Métodos de Pontos Interiores ..	34
2.9.1	O Método de Newton para uma variável	35
2.9.2	O Método de Newton para várias variáveis	36
2.10	Métodos de Pontos Interiores Primal – Dual	39
Capítulo 3	Complexidade de Algoritmos	42
3.1	Considerações Iniciais	42
3.2	Correção dos Algoritmos	42

3.3	Eficiência dos Algoritmos	43
3.4	Análise de Algoritmos	44
3.4.1	Notação O (grande Oh).....	47
3.4.2	Complexidades mais comuns	49
3.5	Técnicas de Análise de Algoritmos	52
3.6	Técnicas de Projetos de Algoritmos	53
3.6.1	Dividir para Conquistar (Divide to Conquer)	53
3.6.2	Programação Dinâmica	54
3.6.3	Algoritmos Gulosos	54
3.7	Análise de Complexidade em Programação Linear	56
3.7.1	A Complexidade do Método Simplex	56
3.7.2	A Complexidade dos Métodos dos Pontos Interiores	57
3.7.3	A Complexidade dos Métodos dos Elipsóides	58
3.7.4	Comparativo entre os Algoritmos	59
Capítulo 4	Fluxo de Potência Ótimo	61
4.1	Considerações Iniciais	61
4.2	Introdução ao Fluxo de Potência	62
4.3	O Problema do Fluxo de Potência Ótimo (FPO)	66
4.4	Formulação do Problema do Fluxo de Potência Ótimo	69
4.4.1	Restrições de Igualdade	70
4.4.1.1	Equações de Balanço de Potência Ativa	70
4.4.1.2	Equações de Balanço de Potência Reativa	71
4.4.1.3	Intercâmbio Líquido entre Áreas	72
4.4.2	Restrições de Desigualdades	73
4.4.2.1	Restrições Físicas	73
4.4.2.2	Restrições Operacionais	73
4.4.2.3	Restrições de Segurança	73
4.4.2.4	Restrições Simples ou de Canalizações	74
4.4.2.5	Restrições Funcionais	74
4.4.3	Função Objetivo	75
4.4.3.1	Mínimo Custo de Geração Ativa	75
4.4.3.2	Mínima Injeção de Potência Reativa	75
4.4.3.3	Mínima Injeção de Potência Ativa	76
4.4.3.4	Mínima Perda Ativa	76
4.4.3.5	Mínimo Corte de Carga	76
4.4.3.6	Mínimo Desvio de Potência Ativa Gerada	77
4.4.3.7	Mínimo Desvio de Ângulo de Defasamento	77
4.4.3.8	Mínimo Desvio de Tensão	78
4.4.3.9	Mínimo Desvio de Tap	78

	4.4.3.10	Mínimo Desvio de Intercâmbio	78
	4.4.3.11	Mínimo Desvio do Ponto de Operação	79
	4.4.4	Função Lagrangeana	79
	4.4.5	Função Penalidades	81
	4.4.6	As Condições de Otimalidade	81
4.5		Modelagem do Fluxo de Potência Linearizado	82
	4.5.1	Função Matricial	83
	4.5.2	Representação das Perdas no Modelo DC	85
	4.5.3	Matriz de Conexão de um Sistema Elétrico	85
Capítulo 5		Programação Matemática Aplicada ao Problema do FPO DC	97
5.1		Considerações Iniciais	97
5.2		Despacho Econômico com Restrições de Desigualdades – Casos IA, IB e IC	98
	5.2.1	Caso IA – Método Simplex	99
	5.2.2	Caso IB – Método de Pontos Interiores	111
	5.2.3	Caso IC – MatLab: Uso da <i>Toolbox Optimization</i> – LINPROG ...	112
	5.2.4	Análise da Complexidade dos Algoritmos – Casos IA, IB e IC....	114
5.3		Despacho Econômico – Caso II	115
5.4		Gerenciamento de Congestionamento – Caso III	119
5.5		Mínima Injeção de Potência Ativa – Caso IV	122
5.6		Gerenciamento de Congestionamento – Caso V	124
5.7		Gerenciamento de Congestionamento – Caso VI	126
5.8		Sumário	131
Capítulo 6		Conclusão	132
		Referências Bibliográficas	134
		Apêndice	137
		Matriz de Conexão para o sistema IEEE14bus	137

Capítulo 1 Introdução

1.1 Considerações Iniciais

O setor elétrico brasileiro e mundial vem passando por diversas transformações (ARAÚJO, 2005). A mudança do modelo de monopólio para o modelo competitivo impõe novas filosofias de operação e planejamento dos sistemas elétricos, envolvendo a geração, a transmissão e a distribuição. Além disto, em grande parte do sistema, o rápido aumento da demanda de energia tem obrigado os sistemas a operarem nos limites de suas capacidades e, por outro lado, a tentativa de expansão enfrenta problemas de características ambientais, sociais e crises financeiras que reduzem os investimentos no setor.

Como alternativa à expansão pode-se atuar, por exemplo, na operação dos sistemas, redespachando geradores e/ou atuando na regulação de equipamentos, tendo como objetivos a diminuição das perdas, a minimização do custo de geração, o aumento da capacidade de transmissão do sistema, ou seja, a otimização de um ou mais índices de seu desempenho.

A principal ferramenta computacional utilizada para determinar o ponto de operação ótimo dos sistemas elétricos é denominada fluxo de potência ótimo (FPO). O fluxo de potência ótimo pode ser usado em conjunto com o estado estimado para periodicamente ajustar o controle ótimo de saída na ordem de manter a tensão factível e as fontes de potências reativas (TERRA, 1989 p.189).

Em 1984, Kamarkar, citado por (WRIGHT, 2004), publicou um artigo no qual, o método de otimização apresentado raramente visita pontos extremos antes que seja encontrado o ponto ótimo, ou seja, o algoritmo acha soluções viáveis no interior

do polígono, evitando desta forma a complexidade combinatória derivada dos vértices da solução. Devido ao procedimento de solução proposto por Kamarkar, o método é chamado de “método de pontos interiores” (MPI), tem características esparsas e vem sendo amplamente utilizado na literatura.

O método de pontos interiores pertence a uma classe de algoritmos de otimização originalmente designados para problemas de programação linear. Entretanto, devido ao seu alto grau de desempenho, tal método foi estendido para problemas de programação quadrática, convexa e problemas gerais de otimização diferenciáveis.

Na aplicação do método de pontos interiores em problemas de fluxo de potência ótimo, em geral, são adotadas duas estratégias distintas. A primeira aplica o método a um problema de programação linear obtido pela linearização das equações de balanço de potência ativa e reativa do algoritmo de fluxo de potência. A segunda consiste em aplicar o método de pontos interiores diretamente ao problema de programação não-linear original do fluxo de potência ótimo.

1.2 Relevância

O problema do fluxo de potência ótimo é um problema de otimização complexo e de grande porte, que faz uso da programação matemática para calcular um conjunto de variáveis de estado e controle da rede, a partir dos dados de carga e dos parâmetros do sistema, enquanto minimiza ou maximiza uma função escalar que representa um índice de desempenho elevado (BAPTISTA, 2004),.

Segundo Lima (2004), o método de pontos interiores tem-se destacado como uma área da programação matemática aplicada na solução de problemas complexos e de grande porte.

Este trabalho busca investigar a aplicação da programação linear na resolução de problema de fluxo de potencia ótimo, modelo DC.

1.3 Objetivo

O objetivo principal é formular o problema do fluxo de potência ótimo em sua forma linearizada e aplicar os elementos da programação matemática na solução do problema.

Objetivos secundários envolvem: estudar a programação linear; examinar a complexidade de algoritmos; analisar as formulações do problema do fluxo de potencia ótimo DC.

1.4 Metodologia

A investigação envolveu a representação da rede elétrica em regime permanente equilibrado. A rede foi representada por fase. Os componentes passivos foram caracterizados pela reatância série, desprezando-se, portanto, as perdas. Os componentes ativos, cargas e geração, foram representados por injeções de potência de barra. Os modelos de simulação foram desenvolvidos e implementados no *software* MatLab©

A revisão bibliográfica abordou os temas programação linear, complexidade de algoritmos, fluxo de potência, fluxo de potência ótimo e aplicações de programação linear.

1.5 Organização do Texto

Este texto está dividido em 6 capítulos, incluído este capítulo de introdução, capítulo 1. Os outros capítulos são resumidamente descritos a seguir.

No capítulo 2 é apresentada uma revisão bibliográfica sobre programação linear.

No capítulo 3 é examinada a complexidade computacional dos algoritmos de otimização para solução de problemas de programação linear (PPL).

No capítulo 4 é estudado o problema do fluxo de potência ótimo, com a formulação e modelagem do problema, análise das variáveis e restrições e a função objetivo. Um exemplo demonstrativo do problema é formulado e resolvido, com a apresentação dos resultados obtidos.

No capítulo 5 é apresentada a programação matemática aplicada ao problema de fluxo de potencia ótimo DC

No capítulo 6 são feitas algumas conclusões e sugeridas direções para trabalhos futuros.

No apêndice é dada a matriz de conexão para o sistema de 14 barras e 20 linhas.

Capítulo 2 Programação Linear

2.1 Considerações Iniciais

Segundo Goldberg & Pacca (2000), durante a Segunda Guerra Mundial, um grupo de cientistas foi convocado na Inglaterra para estudar problemas de estratégia e de tática associados com a defesa do país. O objetivo era decidir sobre a utilização mais eficaz de recursos militares limitados. A convocação deste grupo marcou a primeira atividade formal de pesquisa operacional.

Os resultados positivos conseguidos pela equipe de pesquisa operacional inglesa motivaram os Estados Unidos a iniciarem atividades semelhantes. Apesar de ser creditada à Inglaterra a origem da Pesquisa Operacional, sua propagação deve-se principalmente à equipe de cientistas liderada por George B. Dantzig, dos Estados Unidos, convocada durante a Segunda Guerra Mundial. Ao resultado deste esforço de pesquisa, concluído em 1947, deu-se o nome de *Método Simplex*.

Com o fim da guerra, a utilização de técnicas de pesquisa operacional atraiu o interesse de diversas outras áreas. A natureza dos problemas encontrados era bastante abrangente e complexa, exigindo portanto uma abordagem que permitisse reconhecer os múltiplos aspectos envolvidos.

Uma característica importante da pesquisa operacional e que facilitou o processo de análise e de decisão foi a utilização de modelos. Eles permitem a experimentação da solução proposta. Isto significa que uma decisão pode ser mais bem avaliada e testada antes de ser efetivamente implementada. A economia obtida e a experiência adquirida pela experimentação justificam a sua utilização.

Com o aumento da velocidade de processamento e quantidade de memória dos computadores atuais, houve um grande progresso na pesquisa operacional. Este progresso foi devido, também, à larga utilização de microcomputadores, que se tornaram unidades isoladas dentro de empresas. Isso fez com que os modelos desenvolvidos pelos profissionais de Pesquisa Operacional fossem mais rápidos e versáteis, além de serem, também, interativos, possibilitando a participação do usuário ao longo do processo de cálculo.

No início da década de 50, começaram a surgir várias áreas, chamadas hoje coletivamente de *programação matemática*. As subáreas de programação matemática cresceram rapidamente com a programação linear desempenhando um papel fundamental nesse desenvolvimento. Entre essas subáreas estão a programação não-linear, que começou por volta de 1951, com a famosa condição de Karush-Kuhn-Tucker, aplicações comerciais, fluxos em redes, programação linear, programação inteira, programação dinâmica, programação estocástica e programação não-linear.

A *programação linear* é utilizada para analisar modelos onde as restrições e a função objetivo são lineares; a *programação inteira* se aplica a modelos que possuem variáveis inteiras (ou discretas); a *programação dinâmica* é utilizada em modelos onde o problema completo pode ser decomposto em subproblemas menores; a *programação estocástica* é aplicada a uma classe especial de modelos onde os parâmetros são descritos por funções de probabilidade; finalmente, a *programação não-linear* é utilizada em modelos contendo funções não-lineares.

Uma característica presente em quase todas as técnicas de programação matemática é que a solução ótima do problema não pode ser obtida em um único passo, devendo ser obtida iterativamente. É escolhida uma solução inicial (que

geralmente não é a solução ótima). Um algoritmo é especificado para determinar, a partir desta, uma nova solução, que geralmente é superior à anterior. Este passo é repetido até que a solução ótima seja alcançada (supondo que ela exista).

2.2 Programação Linear

O problema geral de programação linear, segundo Bazzara (1977), é utilizado para otimizar (maximizar ou minimizar) uma função linear de variáveis, chamada de "função objetivo", sujeita a uma série de equações ou inequações lineares, chamadas restrições. A formulação do problema a ser resolvido por programação linear segue alguns passos básicos descritos a seguir.

1. Deve ser definido o objetivo básico do problema, ou seja, a otimização a ser alcançada. Por exemplo, maximização de lucros, ou de desempenhos, ou de bem-estar social; minimização de custos, de perdas, de tempo. Tal objetivo será representado por uma função objetivo, a ser maximizada ou minimizada;
2. Para que esta função objetivo seja matematicamente especificada, devem ser definidas as variáveis de decisão envolvidas. Por exemplo, número de máquinas, a área a ser explorada, as classes de investimento à disposição, etc. Normalmente, assume-se que todas estas variáveis possam assumir somente valores positivos;
3. Estas variáveis normalmente estão sujeitas a uma série de restrições, normalmente representadas por inequações. Por exemplo, quantidade de equipamento disponível, tamanho da área a ser explorada, capacidade de um reservatório, exigências nutricionais para determinada dieta, etc.

Todas essas expressões, entretanto, devem estar de acordo com a hipótese principal da programação linear, ou seja, todas as relações entre as variáveis devem ser lineares. Isto implica proporcionalidade das quantidades envolvidas. Esta característica de linearidade pode ser interessante no tocante à simplificação da estrutura matemática envolvida, mas prejudicial na representação de fenômenos não lineares (por exemplo, funções de custo tipicamente quadráticas).

A forma canônica de um problema de programação linear é apresentada a seguir:

$$\begin{array}{l} \text{Max.} \quad c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \\ \text{sujeito a} \quad \left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ x_1, x_2, \dots, x_n \geq 0 \end{array} \right. \end{array}$$

onde $c_1x_1 + c_2x_2 + \dots + c_nx_n$ representa a função objetivo linear a ser maximizada e pode ser denotada ou representada por z . Os coeficientes c_1, c_2, \dots, c_n representam os custos (valores conhecidos) e x_1, x_2, \dots, x_n representam as variáveis de decisão; seus valores deverão ser obtidos pela solução, caso exista, do problema. As desigualdades $\sum_{j=1}^n a_{ij}x_j \leq b_i$ representam o conjunto de restrições lineares com $i \in \{1, 2, \dots, m\}$ e $j \in \{1, 2, \dots, n\}$. O conjunto de todos os coeficientes a_{ij} compõe a matriz dos coeficientes tecnológicos.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

E $x_1, x_2, \dots, x_n \geq 0$ assegura a não negatividade das variáveis de decisão.

2.2.1 Características da Programação Linear

Para representar um problema de otimização como um programa linear, diversas características necessitam ser previamente discutidas e analisadas junto à formulação do problema de programação linear. Entre estas características, Silva (1998) destaca:

1. **Proporcionalidade:** O valor da função objetivo é diretamente proporcional ao nível de atividades de cada variável de decisão;
2. **Aditividade:** Considera as atividades (variáveis de decisão) do modelo como entidades totalmente independentes, não permitindo que haja interdependência entre as mesmas, isto é, não permitindo a existência de termos cruzados, tanto na função objetivo, quanto nas restrições;
3. **Divisibilidade:** Assume que todas as unidades de atividades possam ser divididas em qualquer nível fracional, isto é, qualquer variável de decisão pode assumir qualquer valor fracionário;
4. **Certeza:** Assume que todos os parâmetros do modelo são constantes conhecidas. Em problemas reais, a certeza quase nunca é satisfeita, provocando a necessidade de análise de sensibilidade dos resultados.

É importante reconhecer que se um problema de programação linear está sendo usado para modelar uma situação dada, então as hipóteses acima são supostas satisfeitas pelo menos num certo domínio das variáveis.

2.3 Modelagem

Um modelo é uma representação de um sistema real, que pode já existir ou ser um projeto aguardando execução. No primeiro caso, o modelo pretende reproduzir o funcionamento do sistema, de modo a aumentar sua produtividade. No segundo caso, o modelo é utilizado para definir a estrutura ideal do sistema. A confiabilidade da solução, obtida através do modelo, depende da validação do modelo na representação do sistema real. A validação do modelo é a confirmação de que ele realmente representa o sistema real. A diferença entre a solução real e a solução proposta pelo modelo depende diretamente da precisão do modelo em descrever o comportamento original do sistema. Um problema simples pode ser representado por modelos também simples e de fácil solução. Já problemas mais complexos requerem modelos mais elaborados, cuja solução pode vir a ser bastante complicada.

2.3.1 Modelagem em Programação Linear

A análise e modelagem de problemas de pesquisa operacional, segundo Silva (1998), envolvem em geral um problema de programação linear que é visto em várias fases:

- Formulação do problema;
- Construção do modelo matemático do sistema;
- Cálculo da solução através do modelo;
- Teste do modelo e da solução;
- Estabelecimento de controles da solução;

- Implantação e acompanhamento.

2.3.1.1 Formulação do Problema

Nesta fase, o administrador do sistema e o responsável pelo estudo em pesquisa operacional deverão discutir para colocar o problema de maneira clara e coerente, definindo os objetivos a alcançar e quais os possíveis caminhos alternativos para que isso ocorra.

Além disso, serão levantadas as limitações técnicas do sistema e as relações desse sistema com outros da empresa, ou do ambiente externo, com a finalidade de criticar a validade de possíveis soluções em face destes obstáculos.

Deverá ainda ser acordada uma medida de eficiência para o sistema, que permita ao administrador ordenar as soluções encontradas, concluindo o processo decisório.

2.3.1.2 Construção do Modelo Matemático do Sistema

Os modelos que interessam em pesquisa operacional são os modelos matemáticos, isto é, modelos formados por um conjunto de equações e inequações.

Uma das equações do conjunto serve para medir a eficiência do sistema para cada solução proposta. É a função objetivo ou função de eficiência. As outras equações geralmente descrevem as limitações ou restrições técnicas do sistema. As variáveis que compõem as equações são de dois tipos:

— **Variáveis controladas ou de decisão** — são variáveis cujo valor está sob controle do administrador. Decidir, neste caso, é atribuir um valor particular a cada

uma dessas variáveis. Numa programação de produção, por exemplo, a variável de decisão é a quantidade a ser produzido num período, o que compete ao administrador controlar;

— **Variáveis não controladas** — são as variáveis cujos valores são arbitrados por sistemas fora do controle do administrador. Custos de produção, demanda de produtos, preço de mercado são variáveis não controladas.

Um bom modelo é aquele que tem desempenho suficientemente próximo do desempenho da realidade e é de fácil experimentação. Essa proximidade desejada é variável, dependendo do objetivo proposto. Um bom modelo para um objetivo pode ser péssimo para outro. A fidelidade de um modelo é aumentada à medida que ele incorpora características da realidade, com a adição de novas variáveis. Isso aumenta sua complexidade, dificultando a experimentação, o que leva a considerar o fator custo-benefício, quando se pensa em melhorar o desempenho de um modelo.

2.3.1.3 Cálculo da solução através do modelo

O objetivo desta fase é encontrar uma solução para o modelo proposto. Ao contrário das outras fases, que não possuem regras fixas, a solução do modelo é baseada geralmente em técnicas matemáticas existentes.

No caso de um modelo matemático, a solução é obtida pelo algoritmo mais adequado, em termos de rapidez de processamento e precisão da resposta. Isto exige um conhecimento profundo das principais técnicas existentes. A solução obtida, neste caso, é dita "ótima".

2.3.1.4 Teste do modelo e da solução

Esse teste é realizado com dados empíricos do sistema. Se houver dados históricos, eles serão aplicados no modelo, gerando um desempenho que pode ser comparado ao desempenho observado no sistema. Se o desvio verificado não for aceitável, a reformulação ou mesmo o abandono do modelo será inevitável. Caso não haja dados históricos, os dados empíricos serão anotados, com o sistema funcionando sem interferência, até que o teste possa ser realizado.

2.3.1.5 Estabelecimento de controles da solução

A construção e experimentação com o modelo identificam parâmetros fundamentais para a solução do problema. Qualquer mudança nesses parâmetros deverá ser controlada para garantir a validade da solução adotada. Caso algum desses parâmetros sofra desvio além do permitido, o cálculo de nova solução ou mesmo a reformulação do modelo poderá ser necessária.

2.3.1.6 Implementação e acompanhamento

Nesta fase, a solução será apresentada ao administrador, evitando-se o uso da linguagem técnica do modelo. O uso da linguagem do sistema em estudo facilita a compreensão e gera boa vontade para a implantação que está sendo sugerida. Essa implantação deve ser acompanhada para se observar o comportamento do sistema com a solução adotada. Algum ajuste pode ser requerido.

2.4 Teoremas da Programação Linear

Seja um conjunto convexo, isto é, um conjunto onde todos os pontos de um segmento de reta que une dois pontos deste conjunto pertencem ao mesmo conjunto. Na Figura 1 pode ser visto uma representação gráfica de conjunto convexo e não-convexo.



Figura 1 – Representação gráfica de conjunto convexo e não-convexo.

Goldbarg & Pacca (2000) elaboraram com maiores detalhes e demonstraram os teoremas da programação linear.

2.4.1 Teorema I

O conjunto de todas as soluções viáveis de um modelo de programação linear é um conjunto convexo.

2.4.2 Teorema II

Toda solução compatível básica do sistema de equações lineares de um modelo de programação linear é um ponto extremo do conjunto de soluções viáveis,

isto é, do conjunto convexo de soluções. Solução compatível básica (ainda que não ótima) é toda solução do sistema em que todas as restrições são satisfeitas.

2.4.3 Teorema III

Se uma função objetivo possui um único ponto ótimo finito, então ele é um ponto extremo do conjunto convexo de soluções viáveis.

2.4.4 Teorema IV

Se a função objetivo assume o valor ótimo, em mais de um ponto do conjunto de soluções viáveis (soluções múltiplas), então ela assume este valor para pelo menos dois pontos extremos do conjunto convexo e para qualquer combinação convexa desses pontos extremos, isto é, para todos os pontos do segmento de reta que unem estes dois pontos extremos, ou seja, a aresta do polígono que contém estes extremos.

2.5 O Método Simplex

Segundo Goldberg & Pacca (2000), um procedimento é uma seqüência finita de instruções e, algoritmo é um procedimento que termina em um número finito de operações.

O método simplex, através de seu algoritmo iterativo, faz a busca pela solução, caso exista, pelos vértices da região viável até encontrar uma solução que não possua soluções vizinhas melhores que ela. Esta é a solução ótima. A solução

ótima pode não existir em dois casos: quando não há nenhuma solução viável para o problema, devido a restrições incompatíveis; ou quando não há máximo (ou mínimo), isto é, uma ou mais variáveis podem tender a infinito e as restrições continuarem sendo satisfeitas, o que fornece um valor sem limites para a função objetivo.

O algoritmo simplex destaca-se como uma das grandes contribuições à programação matemática do século XX. É um algoritmo geral extremamente eficiente, como afirma Goldberg & Pacca (2000), para a solução de sistemas lineares e adaptável ao cálculo computacional. Sua compreensão funcional embasará vários outros métodos. Contrariando esta afirmação, Latoree citado em Araújo (2005) afirma que apesar do método simplex ser muito eficiente na prática, ele apresenta complexidade exponencial, ou seja, o número de iterações cresce exponencialmente com o número de variáveis do problema. No capítulo 3, será abordada a complexidade dos algoritmos.

2.6 Dualidade

Dualidade é um conceito amplo que engloba a possibilidade do tratamento de duas naturezas distintas de uma mesma entidade. Em programação matemática modelos primal-dual são modelos que preservam as seguintes condições:

- Possuem funções objetivo simétricas, ou seja, se o primal for maximização o dual será de minimização e vice-versa;
- Possuem simetria na descrição das restrições, ou seja, se na forma canônica primal possui restrições \leq então o dual possuirá restrições \geq ;

- Os termos independentes no primal surgem como os coeficientes da função objetivo no dual e vice-versa;
- O número de restrições do primal é igual ao número de variáveis do dual e vice-versa;
- A matriz de restrição do primal é a transposta da matriz de restrição do dual e vice-versa.

2.6.1 Modelagem Matemática

Seja um problema de programação linear, na forma padrão, dado por:

$$\begin{aligned}
 & \text{Min. } c^t x \\
 & \text{s.a.: } Ax = b \\
 & x \geq 0
 \end{aligned} \tag{2.1}$$

Esta formulação é denominada problema primal e está associada a um problema dual:

$$\begin{aligned}
 & \text{Max. } b^t y \\
 & \text{s.a.: } A^t y + z = c \\
 & y \text{ livre}; z \geq 0
 \end{aligned} \tag{2.2}$$

2.7 Análise de Sensibilidade

A análise de sensibilidade é basicamente uma técnica utilizada para avaliar os impactos que o programa sofre quando existem modificações nas condições de modelagem. Segundo Goldbarg & Pacca (2000), a análise de sensibilidade é,

basicamente, o estudo de um modelo de programação matemática submetido a mudanças em suas condições iniciais. As mudanças poderão abranger:

- Mudança no vetor de custos;
- Mudança no vetor de termos independentes;
- Mudança na matriz de restrições;
- Acréscimo de restrições;
- Acréscimo de novas variáveis.

2.8 O método dos Pontos Interiores (MPI)

Os métodos dos pontos interiores tiveram seu reconhecimento em 1984, quando Karmarkar propôs um algoritmo polinomial que requer $(n^{3,5}L)$ operações aritméticas e (nL) iterações no pior caso, assegurando que o processo iterativo seja da ordem de 50 vezes mais rápido que o método simplex (WRIGHT, 2004). Inicialmente, foi muito criticado pela comunidade científica sobre o desempenho do método, mas resultados apresentados por (ADLER, 1989) deram um impulso ao desenvolvimento desta classe de métodos. A revolução dos métodos dos pontos interiores, assim como muitas outras revoluções, inclui idéias anteriores que são redescobertas ou vistas sob uma luz diferente, junto com idéias genuinamente novas (WRIGHT, 2004).

Dada uma região factível de solução de um problema de programação linear (ou não-linear), um ponto interior é aquele em que todas as variáveis (coordenadas) se encontram dentro desta região, denominada região de soluções viáveis. A essência dos métodos dos pontos interiores consiste em aplicar o método de Newton às condições de otimalidade de um problema de programação linear

(AZEVEDO, 2002). Considerando a modelagem matemática descrita no item 2.6.1, se um ponto (x, y, z) é ótimo, então os problemas primal e dual, têm como condições de otimalidade a serem satisfeitas:

1. Factibilidade Primal: $b - Ax = 0, x \geq 0$.
2. Factibilidade Dual: $c - A^t y - z = 0, z \geq 0$.
3. Complementaridade: $x_i z_i = 0, i = 1, \dots, n$.

Onde: n é a dimensão dos vetores x e z .

Um conceito, também importante, utilizado para a construção de métodos de pontos interiores é o *gap* (γ). O *gap* é a diferença entre os valores das funções objetivo para o primal e o dual de um mesmo problema. No caso das formulações dadas na modelagem matemática do item 2.6.1, tem-se: $\gamma = c^t x - b^t y$.

A trajetória do ponto interior pode ser representada na Figura 2.

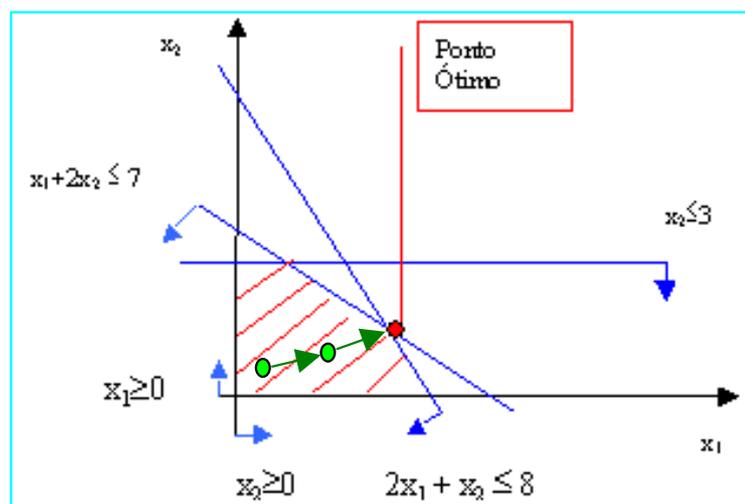


Figura 2 – Representação do ponto interior

O algoritmo de Karmarkar é significativamente diferente do método simplex de George Dantzig, que resolve um PPL começando com um ponto extremo ao longo do limite para, finalmente, chegar a um ponto extremo ótimo. O método projetado por

Karmarkar raramente visita pontos extremos, antes que um ponto ótimo seja alcançado, ou seja, o algoritmo acha soluções viáveis no interior da solução, como verificado na Figura 2.

O MPI tenta encontrar uma solução no centro do polígono, achando uma direção melhor para o próximo movimento, no sentido de obter a solução ótima para o problema. Escolhendo os passos corretamente, uma solução ótima é alcançada depois de algumas iterações.

Embora para encontrar uma direção de movimento, a abordagem de MPI requiera um tempo computacional maior do que o método simplex tradicional, menor número de iterações será requerida pelo MPI para alcançar a solução ótima. Desta forma, a abordagem de MPI tornou-se uma ferramenta competitiva com o método simplex e portanto tem atraído a atenção da comunidade de otimização.

A Figura 3 ilustra como os dois métodos se aproximam da solução ótima. Neste exemplo, o algoritmo de MPI requer aproximadamente a mesma quantidade de iterações que o método simplex. Porém, para um problema de grande porte, este método requereria somente uma fração do número de repetições exigido pelo método simplex, sem contar que o MPI trabalha perfeitamente com não-linearidades.

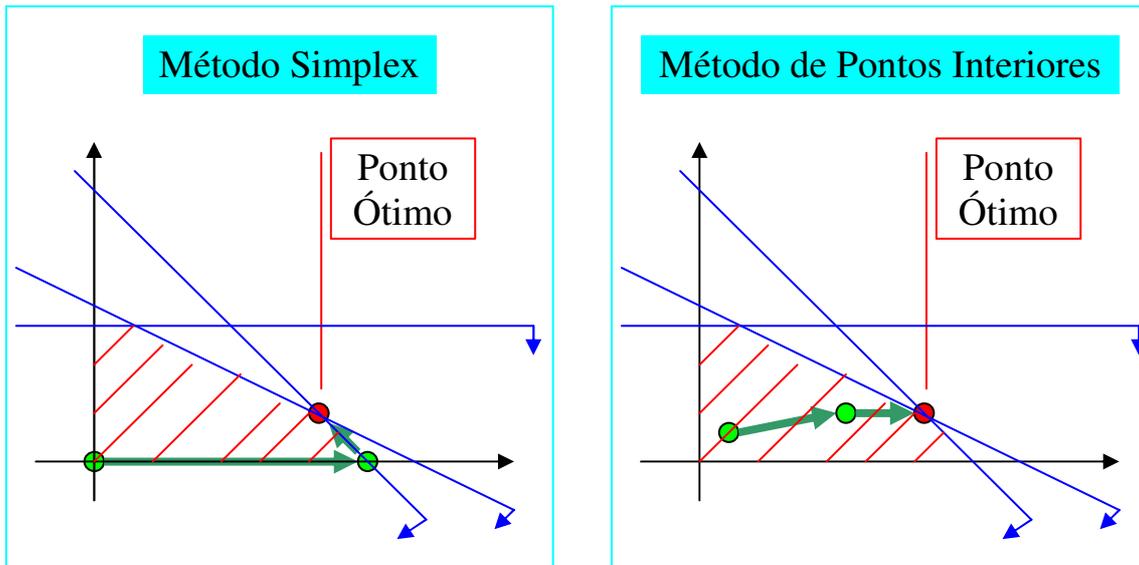


Figura 3 – Método Simplex X Método de Pontos Interiores

O método de Newton, discutido rapidamente a seguir, é utilizado em todos os métodos de pontos interiores que possuem não linearidades, para definir a direção de busca ou do caminhamento da solução.

2.9 O Método de Newton na Construção de Métodos de Pontos Interiores

O método de Newton é um poderoso método iterativo para cálculo de funções e sistemas não lineares e tem uma convergência muito rápida para a solução aproximada do problema, considerando uma tolerância (ε) aceitável. Isto posto, como visto anteriormente, a essência de um método de pontos interiores consiste em se aplicar o método de Newton às condições de otimalidade de um problema de programação linear. Para tanto vem, a seguir, o funcionamento do método de Newton para uma variável e, para várias variáveis, apresentado em (LIMA, 2004).

2.9.1 O Método de Newton para uma variável

Seja $x \in \Omega$ tal que $\phi(x) = 0$. Para encontrar-se o valor de x , utiliza-se sucessivas aproximações da função $\phi(x) = 0$ em torno dos pontos x^0, x^1, \dots, x^k , até que o ponto x^k seja tal que $\phi(x^k) \approx 0$. A aproximação inicial utilizada é dada pela fórmula de Taylor em torno de x^0 :

$$\phi(x) = \phi(x^0) + \phi'(x^0)(x - x^0) + \phi''(x^0)(x - x^0)^2 / 2! + \dots$$

Fazendo a aproximação linear :

$$\phi(x) = \phi(x^0) + \phi'(x^0)(x^1 - x^0) = 0 \Rightarrow x^1 = x^0 - \phi(x^0) / \phi'(x^0)$$

Pode-se aplicar esta fórmula para obter x^1 e depois aplicar o valor de x^k , sucessivamente, de modo a obter novos valores, até que $\phi(x^k) \approx 0$. Constrói-se, assim, o método de Newton para uma variável, como representado no algoritmo:

Dado $x^0 \in \Omega$

Para $k=0, 1, \dots$, faça

$$d^k = -\phi(x^k) / \phi'(x^k)$$

$$x^{k+1} = x^k + d^k$$

Até convergir.

De maneira semelhante, tem-se o método de Newton para duas ou mais variáveis.

2.9.2 O Método de Newton para várias variáveis

Seja $x \in \mathfrak{R}^n$ tal que $\phi_i(x_i) = 0$, para $i = 1, 2, \dots, n$. Novamente, para encontrar-se o valor de x adequado utiliza-se sucessivas aproximações das funções $\phi(x)$, em torno dos pontos x^0, x^1, \dots, x^k , até que o ponto x^k seja tal que $\phi_i(x^k) \approx 0$. A diferença em relação ao método de Newton para uma variável se dá na aproximação fornecida pela fórmula de Taylor para várias variáveis:

$$\phi_i(x) = \phi_i(x^0) + (\nabla \phi_i(x^0))^t (x - x^0) + \dots$$

$$\text{onde } (\nabla \phi_i(x^0))^t = [\delta \phi_i(x^0) / \delta x^1, \dots, \delta \phi_i(x^0) / \delta x^n].$$

Fazendo a aproximação linear, te-se:

$$0 = \phi_i(x^0) + (\nabla \phi_i(x^0))^t (x^1 - x^0), i = 1, \dots, n$$

$$(\nabla \phi_i(x^0))^t (x^1 - x^0) = -\phi_i(x^0), i = 1, \dots, n$$

$$\text{Seja a matriz Jacobiana : } J(x^0) = \begin{pmatrix} (\nabla \phi_1(x^0))^t \\ \dots \\ (\nabla \phi_n(x^0))^t \end{pmatrix} \text{ e } f(x^0) = \begin{pmatrix} \phi_1(x^0) \\ \dots \\ \phi_n(x^0) \end{pmatrix}, \text{ então:}$$

$$J(x^0)(x^1 - x^0) = -f(x^0) = - \begin{pmatrix} \phi_1(x^0) \\ \dots \\ \phi_n(x^0) \end{pmatrix}$$

$$x^1 = x^0 - (J(x^0))^{-1} f(x^0).$$

A matriz Jacobiana é composta pelas equações dos *mismatches*, isto é, cada linha da matriz é composta pelas derivadas parciais de 1ª ordem das equações do sistema.

Constrói-se, assim, o método de Newton para várias variáveis representadas no algoritmo:

Dado $x^0 \in \mathfrak{R}^n$

Para $k=0, 1, \dots$, faça

$$d^k = -(J(x^k))^{-1} f(x^k)$$

$$x^{k+1} = x^k + d^k$$

Até convergir.

Tendo em vista as condições de otimalidade descritas na seção 2.8 e o método de Newton para múltiplas variáveis, o próximo passo consiste em obter um método de pontos interiores.

Considerando o problema com formulação primal e dual, visto na seção 2.6.1, a idéia para se construir um método de pontos interiores consiste em se aplicar o método de Newton ao sistema $f(x, y, z) = 0$ formado pelas condições de otimalidade, resolvendo os problemas primal e dual simultaneamente. Tem-se que $f(x, y, z)$ é dada por:

$$f(x^0, y^0, z^0) = \begin{pmatrix} Ax^0 - b \\ A^t y^0 + z^0 - c \\ X^0 Z^0 e \end{pmatrix} = - \begin{pmatrix} r_p^0 \\ r_d^0 \\ r_a^0 \end{pmatrix} = \begin{pmatrix} f_p \\ f_d \\ f_a \end{pmatrix},$$

onde $X^0 = \text{diag}(x^0)$, $Z^0 = \text{diag}(z^0)$, e representa o vetor unitário e $X^0 Z^0 e = 0$ é equivalente a $x_i^0 z_i^0 = 0, i = 1, \dots, n$.

Dado um ponto (x^0, y^0, z^0) e desconsiderando as restrições $x \geq 0$ e $z \geq 0$, calcula-se o ponto (x, y, z) utilizando o método de Newton para várias variáveis:

$$(x^1, y^1, z^1) = (x^0, y^0, z^0) - j((x^0, y^0, z^0))^{-1} f(x^0, y^0, z^0),$$

$$\text{onde } J(x^0, y^0, z^0) = \begin{pmatrix} \nabla f_p \\ \nabla f_d \\ \nabla f_a \end{pmatrix} = \begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I \\ Z & 0 & X \end{pmatrix}.$$

Assim, d será dado por:

$$d^0 = \begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I \\ Z^0 & 0 & X^0 \end{pmatrix}^{-1} \begin{pmatrix} r_p^0 \\ r_d^0 \\ r_a^0 \end{pmatrix} = \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}.$$

Pode-se, segundo (LIMA, 2004), resumir estes cálculos em um algoritmo como segue:

Método primal-dual afim-escala

Dado um ponto (x^0, y^0, z^0) interior, ou seja, $(x^0, y^0) > 0$.

Para $k=0, 1, \dots$, faça

$$r_p^k = b - Ax^k$$

$$r_d^k = c - A^t y^k - z^k$$

$$r_a^k = -X^k Z^k e$$

$$d^k = \begin{pmatrix} dx^k \\ dy^k \\ dz^k \end{pmatrix} = \begin{pmatrix} A & 0 & 0 \\ 0 & A^t & I \\ Z^k & 0 & X^k \end{pmatrix}^{-1} \begin{pmatrix} r_p^k \\ r_d^k \\ r_a^k \end{pmatrix}$$

Calcula-se o tamanho do passo α_p^k e α_d^k , tal que $x^{k+1} > 0$ e $z^{k+1} > 0$.

$$x^{k+1} = x^k + \alpha_p^k dx^k$$

$$y^{k+1} = y^k + \alpha_d^k dy^k$$

$$z^{k+1} = z^k + \alpha_d^k dz^k$$

Até convergir.

Ou seja, aplica-se o método de Newton às condições de otimalidade e caminha-se na direção encontrada, mantendo as variáveis não-negativas interiores.

Uma observação importante é que dado x^0 e z^0 interiores, o tamanho do passo α é calculado de forma que x^{k+1} e z^{k+1} permaneçam interiores. Com y^{k+1} não é necessário a atenção, pois y trata-se de uma variável livre. Levando em conta essa afirmação, tem-se que os tamanhos dos passos primal e dual são:

$$\alpha_p^k = \min\{1, \tau\rho_p\}$$

$$\alpha_d^k = \min\{1, \tau\rho_p\}$$

onde:

$$\rho_p = \min_{dx_i < 0} \left\{ -\frac{x_i}{dx_i} \right\}$$

$$\rho_p = \min_{dz_i < 0} \left\{ -\frac{z_i}{dz_i} \right\}$$

Adota-se o passo máximo $\alpha = 1$, porque este é o tamanho de passo natural no método de Newton e $\alpha = 1$ pode levar a um ponto factível em uma iteração, como afirma Lima (2004).

2.10 Método de Pontos Interiores Primal-Dual

Na seção anterior, foi apresentado o algoritmo de como obter um método de pontos interiores primal-dual afim-escala. Este método, segundo Lima (2004), tem uma grande desvantagem, pois permite que os pontos calculados (x, z) se aproximem de seus limites muito rapidamente.

Conseqüentemente, as direções calculadas perto destes limites são muito distorcidas, pois o valor de $x_i z_i$ se torna próximo de zero rapidamente e o método progride lentamente, podendo inclusive não convergir.

Par evitar que isto aconteça, acrescenta-se a cada iteração uma perturbação μ^k na condição de complementaridade. No caso do problema estudado na seção 2.9, seria $x_i z_i = 0$, tal que o método primal-dual deverá encontrar a solução do seguinte sistema:

$$\begin{cases} b - Ax = 0 \\ c - A^t y - z = 0 \\ \mu e - XZe = 0 \end{cases}$$

Assim, o desenvolvimento do método de pontos interiores para um problema de programação linear, que tem restrições de igualdade e desigualdade, bem como variáveis canalizadas, pode utilizar uma perturbação na condição de complementaridade de modo a tornar o método mais eficiente. Este assunto é sugestivo para estudos futuros.

Sinteticamente, a seqüência de cálculo, freqüentemente utilizada pelo método de Newton considerando o problema de fluxo de potência ótimo, consiste em:

1. Determinar os valores das potências ativas e reativas que circulam no sistema em cada barra, baseando-se em estimativas para os módulos das tensões e ângulos;
2. Calcular as derivadas parciais das tensões em relação às tensões e ao ângulo, agrupando os valores na matriz Jacobiana;
3. Resolver o sistema composto pelo vetor de potências e pela matriz Jacobiana, encontrando a variação dos ângulos e tensões;

4. Calcular os novos valores de ângulos e tensões, somando os valores obtidos no passo anterior com os valores estimados;
5. Repetir todos os passos até que as variações obtidas no passo 3 se tornem menores que uma precisão (tolerância) pré-determinada.

No capítulo seguinte, será feito um estudo sobre complexidade de algoritmos, levando-se em conta o desempenho dos algoritmos de programação linear considerando a complexidade de tempo e a complexidade de espaço.

Capítulo 3 Complexidade de Algoritmos

3.1 Considerações Iniciais

Um algoritmo é uma seqüência bem definida de procedimentos (passos) que levam uma entrada a ser transformada em uma saída (CORMEN, 2002). É uma ferramenta para resolver um problema computacional bem especificado. O enunciado do problema define em termos gerais o relacionamento entre a entrada e a saída desejada, afirma Cormen (2002). De encontro a este conceito, (GOLDBARG & PACCA, 2005) afirmam que algoritmo é a descrição do cálculo ou avaliação sistemática da função processamento que faz a transformação de dados de entrada em dados de saída.

Uma seqüência de entrada é denominada instância do problema e deve satisfazer quaisquer restrições impostas no enunciado do problema e, ao ser processada pelo algoritmo, obter, caso exista, uma solução para o problema. O tamanho de uma instância é determinado pelo total de códigos necessários à sua identificação, considerando o tipo e a estrutura de dados utilizada.

3.2 Correção dos algoritmos

A verificação do funcionamento adequado de um algoritmo é um dos pontos mais importantes tratado pela literatura. Muitas vezes, um algoritmo é capaz de solucionar somente certo conjunto de instâncias. Para garantir a correção de um algoritmo na solução de certo problema, é necessário garantir seu acerto em todas as instâncias possíveis do mesmo. Basta a exibição de um caso de fracasso para

demonstrar sua ineficiência. Por outro lado, para provar seu acerto serão necessárias árduas demonstrações. Uma forma de reduzir a dificuldade da prova da correção de um algoritmo é estabelecer os domínios de definição, ou as limitações das regiões conhecidas como regiões factíveis.

Então, segundo Leitão (2006), a correção do algoritmo significa trabalhar corretamente quaisquer que sejam os dados de entrada em certo domínio. Assim, por exemplo, uma estrutura, que deve guardar uma coleção de números ordenadamente, nunca deve permitir que sejam guardados números fora de ordem. De igual modo, um algoritmo para listar valores por ordem nunca deve fazer a saída de valores fora dessa ordem.

3.3 Eficiência dos algoritmos

Às vezes, o algoritmo mais imediato está longe de ser razoável em termos de eficiência, afirma Toscani (2001). Um exemplo é o caso da solução de sistemas de equações lineares. O método de Cramer (IEZZI,1993), calculando o determinante através de sua definição, requer dezenas de milhões de anos para resolver um sistema 20×20 . Um sistema como esse pode ser resolvido em tempo razoável pelo método de Gauss. A Tabela 1 mostra segundo (TOSCANI, 2001) o desempenho desses dois algoritmos que calculam o determinante de uma matriz $n \times n$, considerando tempos de operações de um computador real.

TABELA 1
Tamanho do Problema x Tempo de Execução

N	Método de Cramer	Método de Gauss
2	22 μs	50 μs
3	102 μs	159 μs
4	456 μs	353 μs
5	2,35 ms	666 μs
10	1,19 min	4,95 ms
20	15225 séculos	38,63 ms
40	5×10^{33} séculos	0,315 s

Fonte: Toscani, 2001

Algoritmos criados para resolver o mesmo problema muitas vezes diferem de forma drástica em sua eficiência. Essas diferenças podem ser muito mais significativas que as diferenças relativas a hardware e software.

Ser capaz de estabelecer um conjunto de instruções que soluciona certo problema decidível não significa que se tenha encontrado uma estratégia razoável, sob qualquer que seja o ponto de vista. O objetivo maior de um algoritmo é solucionar o problema de uma forma rápida e econômica. Por forma rápida entende-se quanto tempo o algoritmo irá consumir durante sua execução e por forma econômica, quanto espaço de memória irá requerer para manipular os dados de entrada.

3.4 Análise de algoritmos

Analisar um algoritmo, ou analisar a complexidade de um algoritmo significa, segundo Cormen (2002), prever os recursos de que o algoritmo necessitará. Ocasionalmente, recursos como memória, largura de banda de comunicação ou

hardware de computador são as principais preocupações mas, com freqüência, é o tempo de computação que se deseja medir. Em geral, pela análise de vários algoritmos candidatos para um problema, pode-se identificar facilmente um algoritmo eficiente. Essa análise pode indicar mais de um candidato viável, mas vários algoritmos de qualidade inferior em geral são descartados no processo, (CORMEN, 2002).

Mas, por que analisar a complexidade de um algoritmo? Segundo Sedgwick (1996), citado em (BARBOSA, 2001), existem muitas respostas para esta questão, dependendo do contexto. Algumas das possíveis respostas podem ser:

- A finalidade da utilização do algoritmo;
- A importância do algoritmo com relação a outros algoritmos que solucionam o mesmo problema, para que se possa decidir por qual deles optar na implementação da solução do problema;
- A dificuldade da precisão e análise da resposta requerida.

A mais simples razão para se analisar um algoritmo é a de apurar suas características e avaliar a viabilidade de sua utilização prática, ou para poder comparar o algoritmo com outros algoritmos desenvolvidos com o mesmo objetivo do algoritmo em questão. É desejável saber quanto tempo uma implementação de um algoritmo particular irá consumir durante sua execução, ou quanto espaço irá requerer para poder certificar-se da eficiência do algoritmo.

A complexidade de um algoritmo está relacionada com o esforço computacional necessário para a sua execução. Algoritmos podem ser avaliados por uma variedade de critérios de medidas. Os critérios mais freqüentemente utilizados são a taxa de crescimento do tempo ou espaço requerido, como visto anteriormente,

para resolver grandes instâncias de um problema. Tais medidas são denominadas, respectivamente, de complexidade de tempo e complexidade de espaço.

Em relação à complexidade de tempo, segundo Barbosa (2001), pode-se ainda ter a complexidade no Pior Caso ou no Caso Médio. A complexidade no pior caso é geralmente a medida mais empregada na prática. Fixado um tamanho de entrada, a análise no pior caso é feita em relação ao número máximo de operações fundamentais necessárias para a resolução de qualquer problema do tamanho fixado. Seu valor pode ser considerado como um limite de complexidade que não será ultrapassado, sendo, portanto, uma garantia de qualidade mínima do algoritmo.

Se a complexidade é tomada como uma complexidade média sobre todas as entradas do tamanho dado, então a complexidade é denominada de complexidade esperada ou complexidade média. A complexidade média de um algoritmo é usualmente mais difícil de obter do que a complexidade no pior caso. A análise no caso médio é baseada nas distribuições probabilísticas dos dados de entrada do algoritmo. A dificuldade no cálculo do caso médio está justamente nas distribuições probabilísticas dos dados de entrada, que nem sempre são conhecidas. Apesar disto, existem bons exemplos de aplicação deste cálculo, porém utilizando sempre distribuição uniforme. Segundo Ziviani (2000), tem-se ainda um terceiro caso que se refere ao Melhor Caso, que mede o menor tempo de execução sobre todas as entradas de tamanho n .

Antes de analisar um algoritmo, deve-se ter um modelo da tecnologia de implementação usada. Um modelo adotado, desde o final da década de 1960, é o RAM (Random Access Machine – Máquina de acesso aleatório).

O modelo RAM simula o funcionamento de um computador elementar. À memória cabe armazenar os dados do programa. Um programa é constituído por um

conjunto de instruções ou comandos que implementa o algoritmo. O modelo considera que cada instrução i possuirá um tempo associado $t(i)$ para ser operacionalizada em RAM. Então, segundo Goldeberg & Pacca (2005), se para executar um algoritmo codificado em um programa P, com uma certa entrada fixa, são processadas r_1 instruções do tipo i_1 , r_2 instruções do tipo i_2 até r_m instruções do tipo i_m , neste caso o tempo para executar o programa P será dado por:

$$\sum_{j=1}^m r_j t(i_j) \quad (3.1)$$

O estudo da complexidade de um algoritmo poderia ser resolvido através da avaliação do somatório (3.1).

O modelo RAM pode ser expresso de acordo com a Figura 4.

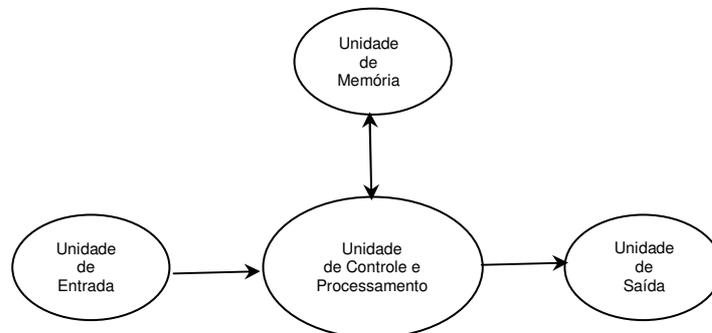


Figura 4 – Modelo RAM.

3.4.1 Notação O (Grande Oh)

A Figura 5 representa a complexidade assintótica na notação O .

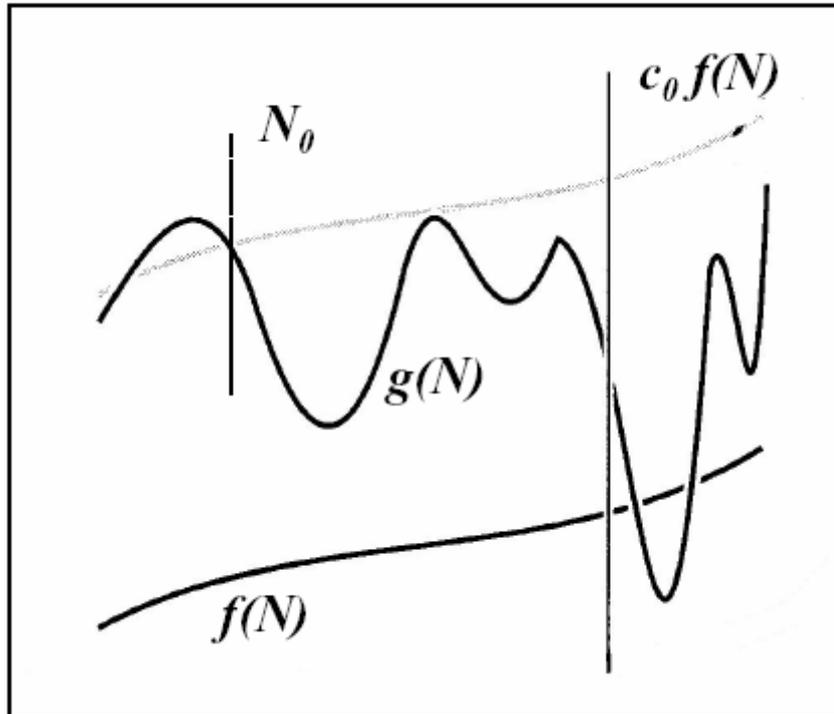


Figura 5 – Notação O – Complexidade assintótica

Uma função $g(n)$ é $O(f(n))$ (lê-se: ordem de $f(n)$) para outra função $f(n)$ se existem as constantes c e N , tal que para $n > N$, tem-se :

$$g(n) \leq c f(n), n > N$$

Isto é, para n suficientemente grande a função $g(n)$ não é maior do que a função $f(n)$ multiplicada por uma constante. A notação O estabelece um limite superior e $g(n)$ pode ser menor ou igual a este limite.

Exemplos:

$$5n^2 + 15 = O(n^2), \text{ pois } 5n^2 + 15 \leq 6n^2, n > 4$$

$$5n^3 + 15 = O(n^3), \text{ pois } 5n^3 + 15 \leq n^3, n > 6$$

Dize-se que a complexidade de tempo ou o tempo de execução de um algoritmo, é $O(f(n))$ e que a complexidade de espaço é $O(f(n))$.

3.4.2 Complexidades mais Comuns

$O(1)$	Constante - independe da entrada	
$O(n)$	Linear	
$O(\log n)$	Logarítmica	
$O(n \log n)$	$n \log n$	
$O(\log \log n)$	Sublogarítmica	
$O(n^2)$	Quadrática	(Polinomial)
$O(n^3)$	Cúbica	(Polinomial)
$O(c^n)$	Exponencial	

$f(n) = O(1)$ (complexidade constante)

O uso do algoritmo independe do tamanho de n . Neste caso, as instruções do algoritmo são executadas um número fixo de vezes.

$f(n) = O(n)$ (complexidade linear)

Em geral, um pequeno trabalho é realizado sobre cada elemento de entrada. Esta é a melhor situação possível para um algoritmo que tem que processar n elementos de entrada, ou produzir n elementos de saída. Cada vez que n dobra de tamanho o tempo de execução dobra.

$f(n) = O(\log n)$ (complexidade logarítmica)

Ocorre tipicamente em algoritmos que resolvem um problema transformando-o em problemas menores. Nestes casos, o tempo de execução pode ser

considerado como sendo menor do que uma constante elevada. Por exemplo, quando n é um milhão, $\log_2 n \gg 20$.

$f(n) = O(n \log n)$

Este tempo de execução ocorre tipicamente em algoritmos que resolvem um problema quebrando-o em problemas menores, resolvendo cada um deles independentemente e depois juntando as soluções. Por exemplo, quando n é um milhão e a base do logaritmo é 2, $n \log_2 n$ é cerca de 20 milhões.

$f(n) = O(n^2)$ (complexidade quadrática)

Algoritmos desta ordem de complexidade ocorrem quando os itens de dados são processados aos pares, muitas vezes em um anel dentro de outro. Por exemplo, quando n é mil, o número de operações é da ordem de 1 milhão. Algoritmos deste tipo somente são úteis para resolver problemas de tamanhos relativamente pequenos.

$f(n) = O(n^3)$ (complexidade cúbica)

Algoritmos de complexidade cúbica, assim como os algoritmos de complexidade quadrática, são úteis apenas para resolver pequenos problemas. Quando n é 100, o número de operações é da ordem de 1 milhão.

$f(n) = O(2^n)$ (complexidade exponencial)

Algoritmos desta ordem de complexidade geralmente não são úteis sob o ponto de vista prático. Eles ocorrem na solução de problemas quando se usa força bruta para resolvê-los. Por exemplo, quando n é vinte, o tempo de execução é cerca

de um milhão. Problemas que somente podem ser resolvidos através de algoritmos exponenciais são ditos *intratáveis*.

Grande parte dos algoritmos, segundo Greve (2004), têm complexidade polinomial variando entre n e n^2 . O algoritmo será dito polinomial para o problema, segundo Gonzaga (1989), se o tempo de execução estiver limitado superiormente por um polinômio em função do tamanho do problema. Neste caso interessa, geralmente, somente o grau do polinômio, pois é ele que caracteriza o crescimento de tempo de computação com o tamanho do problema.

O gráfico 1, (LEITÃO, 2006) mostra as funções características associadas à complexidade temporal de algoritmos.

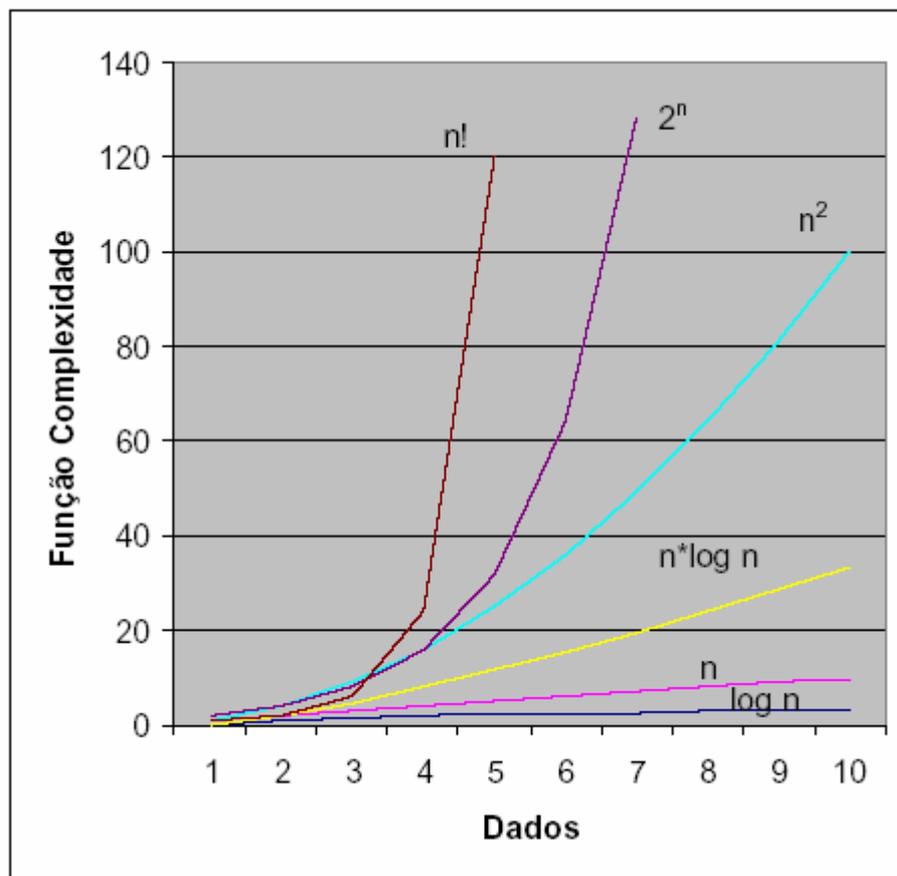


Gráfico 1 – Funções características de complexidade temporal

Fonte: LEITÃO, 2006

3.5 Técnicas de análise de algoritmos

Analisar um algoritmo envolve técnicas de matemática discreta incluindo: manipulação de somatórios, produtórios, permutações, fatoriais, coeficientes binomiais, soluções de recorrência, entre outras.

Dentre os princípios básicos da análise de algoritmos, pode-se destacar:

1. O tempo de execução de um comando de atribuição, de leitura ou escrita pode ser considerado constante, isto é $O(1)$;
2. O tempo de execução de uma seqüência de comandos é determinado pelo maior tempo de execução de qualquer comando da seqüência;
3. O tempo de execução de um comando de decisão é composto pelo tempo de execução dos comandos incluídos dentro do comando condicional, mais o tempo para avaliar a condição que é $O(1)$. Exceto, quando esta avaliação envolve a execução de alguma função cujo tempo de execução não seja constante;
4. O tempo de execução de um laço é a soma do tempo de execução do corpo do laço mais o tempo de avaliar a condição de parada, multiplicado pelo numero de iterações do laço. Geralmente, o tempo para avaliar a condição de parada é $O(1)$;
5. Quando há chamada a procedimentos não recursivos, o tempo de execução de cada procedimento a ser chamado deve ser computado separadamente um a um, iniciando com os procedimentos que não chamam outros procedimentos. A seguir, devem ser avaliados os procedimentos que chamam estes procedimentos “básicos” (cujos tempos

de execução já foram computados). Este procedimento é repetido até chegar ao programa principal;

6. Quando o programa possui procedimentos recursivos, para cada procedimento é associada uma função de complexidade $f(n)$ e, em geral, a análise desta função envolve a solução de uma relação de recorrência.

3.6 Técnicas de projeto de algoritmos

Dentre as técnicas de projeto de algoritmos, pode-se citar:

3.6.1 Dividir para Conquistar (Divide to Conquer)

Esta técnica subdivide o problema de tamanho n em subproblemas de tamanho menor. O ideal, segundo Greve (2004), seria fazer a divisão de maneira balanceada, por exemplo, um problema de tamanho n pode ser subdividido em 2 problemas de tamanho $n/2$.

Exemplos:

1. Determinação do Máximo e Mínimo
2. MergeSort
3. Problema do Skyline
4. QuickSort, etc.

3.6.2 Programação Dinâmica

A técnica de divisão e conquista é adequada quando a soma dos tamanhos dos subproblemas é pequena. Entretanto, se a divisão dá origem a problemas de tamanho elevado, por exemplo, $n-1$, talvez seja mais eficiente utilizar a técnica de programação dinâmica.

Esta técnica é usada, normalmente, quando uma solução imediata do problema possui complexidade exponencial. Isto porque, na tentativa de resolver o problema, tende-se a resolver os mesmos subproblemas diversas vezes. Desta maneira, tenta-se lembrar todas as soluções já encontradas para nunca resolver o mesmo problema duas ou mais vezes.

A essência da programação dinâmica será, portanto, construir amplas tabelas com todos os resultados conhecidos armazenados. Cada entrada será calculada a partir de uma combinação de entradas acima dela ou à sua esquerda. O principal problema é organizar a construção da matriz de maneira eficiente. Normalmente, o tempo de computação é superior a $O(n^2)$.

Exemplos:

1. Multiplicação de n matrizes;
2. Problema do Knapsack., etc.

3.6.3 Algoritmos Gulosos

Alguns problemas podem ser resolvidos pela técnica “gulosa”: A cada ponto, onde tem-se de tomar uma decisão, toma-se aquela que parece ser a melhor neste

momento, sem analisar se ela vai ou não prejudicar no futuro (*caminho do menor esforço*).

É a técnica de utilizar a *optimalidade local* para resolver o problema. Observando o problema como um todo, para cada ponto onde uma decisão é tomada, tem-se um estado do problema. A técnica “gulosa” consiste em só observar o contexto “local” deste estado para escolher-se o próximo estado.

Um exemplo típico de algoritmos gulosos é o algoritmo para a Árvore Expandida de Custo Mínimo de Kruskal. Neste caso, escolhe-se o próximo vértice somente pelo critério “*qual é o vértice livre que está mais próximo ?*”.

Pode-se representar um problema através de uma tupla $\langle a, b, c, \dots, n \rangle$, consistindo de todas as variáveis do problema e de um conjunto possível de estados (espaço de estados), representando todos os valores possíveis deste conjunto de variáveis. A idéia da técnica gulosa é resolver o problema, sempre aplicando o algoritmo de solução, de maneira a tentar ir para um estado de menor valor (ou custo) possível.

Característica da Técnica Gulosa:

Muitos problemas podem ser resolvidos na sua forma mais “branda” utilizando-se uma técnica gulosa. A solução ótima absoluta, porém, não é garantida.

Exemplo: distribuição de tempo de trabalho entre processadores:

- Pode-se minimizar o tempo médio de término de um processo usando a técnica gulosa e fazendo cada processador, assim que está livre, pegar a tarefa menor, mais curta e mais simples.
- Não se pode minimizar o tempo final (*mínimo absoluto* do ponto de vista do custo operacional dos processadores) usando essa técnica, pois o mínimo absoluto é *global* e o critério de otimização é *local*.

3.7 Análise de Complexidade em Programação Linear

3.7.1 A Complexidade Computacional do Método Simplex

A cada iteração do algoritmo simplex, a tarefa mais importante consiste na resolução dos sistemas básicos, que pode ser feito em $O(n^3)$ operações elementares. Isto posto, segundo Sousa (2005), o esforço necessário para resolver um problema de otimização linear, em sua forma padrão, pelo algoritmo simplex, pode ser medido pelo número de iterações.

A questão da eficiência do método simplex sempre foi tema de pesquisa desde a sua publicação. No início, a convergência finita do método, garantida com regras que evitassem ciclos, interpretados como repetições indefinidas de uma solução básica, era suficiente para os pesquisadores, mesmo que o número máximo de iterações pudesse ser muito elevado. Um conjunto de relatos, na maioria informal e não publicado, sobre sua eficiência computacional para se resolver problemas práticos ou gerados de forma aleatória, formou o chamado *folclore* do método simplex, o qual afirma que, na prática, o número de iterações requeridas é um polinômio de grau baixo de n (número de restrições). Então, (SOUSA, 2000) realizou um estudo computacional para o problema de corte de peças em estoque, que envolve milhares de variáveis e observou que o número de iterações é da ordem de n^2 , fornecendo uma classe de problemas práticos que reforça o folclore simplex.

A experiência computacional, com um número grande de problemas resolvidos em vários anos (BAZARAA, 1992), indicou que o número médio de iterações é uma função linear de n , e parece ser menor do que n^3 .

No entanto, Klee & Minty, citados em (SOUZA, 2005), apresentaram um exemplo, para o qual o método simplex, com o critério de Dantzig para escolha da variável a entrar na base (i.é., menor custo relativo), necessita de $2^n - 1$ iterações (onde n é o número de variáveis), percorrendo todos os vértices da região factível do problema, a qual é definida como uma distorção do hipercubo m -dimensional e tem 2^n vértices.

Assim, o algoritmo simplex não pertence à classe de algoritmos com tempo polinomial e sim exponencial. Variações do método simplex, mais tarde, também se mostraram ineficientes, do ponto de vista do estudo do pior caso, necessitando um número exponencial de iterações, como afirma Bertsimas e Tsitsiklis, também citados em (SOUZA, 2005).

Portanto, a questão continua aberta quanto à possibilidade da construção de um método do tipo simplex polinomial, ou a prova definitiva de que é impossível construir um algoritmo do tipo simplex com complexidade polinomial. Segundo Shamir (1987), Terlaky & Zhang (1993), esta é a questão aberta mais desafiante na teoria da otimização linear, como pode ser conferido em (SOUZA, 2005).

3.7.2 A Complexidade do Método dos Pontos Interiores

Como foi mencionado no item 2.8, os métodos dos pontos interiores tiveram seu reconhecimento em 1984, quando Karmarkar propôs um algoritmo polinomial, assegurando que o processo iterativo era da ordem de 50 vezes mais rápido que o método simplex (WRIGHT, 2004).

O método de pontos interiores necessita de um ponto viável interior (relativo) disponível. A partir daí, gera-se novos pontos interiores em uma vizinhança de uma trajetória central até atingir uma certa tolerância para uma solução ótima. Para se atingir uma solução ótima, deve-se realizar um procedimento de purificação de uma solução. A complexidade dos algoritmos de pontos interiores é polinomial, como dito anteriormente.

3.7.3 A Complexidade dos Métodos dos Elipsóides

A idéia geométrica do método dos elipsóides, desenvolvido por Shor (1970), Iudin e Nemirovskii (1976), citado em (BUENO, 2002), é partir de uma bola (elipsóide inicial) que contém ao menos parte do conjunto de soluções do problema de programação linear (PPL), caso exista alguma, e gerar uma seqüência de elipsóides, com volumes cada vez menores, até que o centro de algum deles seja uma solução para o (PPL). Se uma solução não for encontrada, após certo número de elipsóides ser gerado, o método conclui que o (PPL) não tem solução.

Um resultado importante para o método dos elipsóides é o número L . Esse número, definido por Khachiyan (1979), citado em (BUENO, 2002), é a quantidade de *bits* binários necessários para se representar os dados do (PPL). Assim, é muito grande em relação aos dados do (PPL). A importância do algoritmo de Khachiyan está na sua complexidade computacional: polinomial no tamanho L do (PPL), conforme Gács e Lovász (1981), também citados em (BUENO, 2002).

No final da década de 70, Khachiyan utilizou o método dos elipsóides para se ter o primeiro algoritmo polinomial para PL. Desde sua publicação, em 1979, variantes do algoritmo de Khachiyan têm surgido para melhorar sua velocidade de

convergência. A idéia de uma dessas variantes, o algoritmo de cortes profundos, é que os elipsóides sejam construídos de tal forma que seus volumes decresçam mais rapidamente a cada iteração. Isto poderia 'apressar' a chegada a alguma solução do problema, o que nem sempre ocorre, conforme afirma (BUENO, 2002). O algoritmo originalmente apresentado por Khachiyan é também denominado algoritmo de cortes centrais.

3.7.4 Comparativo entre os algoritmos

A Tabela 2, mostra um comparativo, apresentado por (BUENO, 2002), em relação ao número de interações de acordo com tamanho do problema, para o exemplo de Klee e Minty, entre os algoritmos de Khachiyan cortes centrais e sua variante cortes profundos, Simplex e Pontos interiores, aqui denominado Preditor-Corretor.

Tabela 2
Número de iterações para o exemplo de Klee e Minty

n (n. de variáveis)	Numero de iterações			
	Algoritmo de Khachiyan		Algoritmo Simplex	Preditor-Corretor
	Cortes Centrais	Cortes Profundos		
2	156	44	3	9
3	425	122	7	13
4	875	257	15	16
5	1.575	452	31	19
6	2.622	721	63	22
7	3.959	1.068	127	23
8	5.654	1.563	255	24
9	7.731	2.138	511	24
10	10.283	2.824	1.023	24
11	13.658	3.686	2.047	24
12	17.336	4.687	4.095	24
13	21.881	5.827	8.191	25
14	26.742	7.161	16.383	25
15	32.407	8.745	32.767	25
16	39.084	10.566	65.535	25
17	46.261	12.455	131.071	25
18	54.410	14.716	262.143	25
19	63.495	17.151	524.287	25
20	73.978	19.754	1.048.575	25

Fonte: Bueno, 2002

Capítulo 4 Fluxo de Potência Ótimo

4.1 Considerações Iniciais

A energia elétrica tem um importante papel na sociedade, desde a sua utilização para fins domésticos, comerciais e na indústria. Sabendo disso, não é possível conceber a falta deste importante insumo em qualquer atividade. Daí a relevância dos estudos relacionados à melhoria de geração e transmissão dessa energia.

Segundo Lima (2004), até o ano de 1970, a energia final consumida de origem elétrica no Brasil tinha uma participação no consumo final menor que 20%. Após o primeiro choque do petróleo, o percentual de consumo final de energia de origem elétrica chegou a 22% em 1975, atingindo, em 1999, um percentual de 40%. Vale lembrar que as hidrelétricas são responsáveis, atualmente, por cerca de 80% da energia elétrica gerada, como afirma Oliveira (1999), citado em (LIMA, 2004).

Este quadro se deve aos investimentos feitos em hidrelétricas de grande porte. Se, por um lado, estes projetos necessitam de investimentos de grande vulto, por outro lado, o custo de geração resultante tem sido compensador em relação às outras alternativas.

Lima (2004), afirma em seu trabalho que o sistema brasileiro de geração de energia elétrica conta com características que o fazem único no mundo:

1. Predominância hidrelétrica;
2. Grandes extensões geográficas e grandes distâncias entre as fontes geradoras e os principais centros consumidores;
3. Vários potenciais de aproveitamento no mesmo rio;

4. Diversidade de regimes hidrológicos e pluviométricos;
5. Grau de interligação elétrica entre os sistemas (região sul/ sudeste/ centro-oeste) relativamente alto, em comparação com outros países;
6. Grande potencial hidrelétrico inexplorado.

Com estas características, pode-se perceber a importância do planejamento integrado da expansão e utilização do sistema de geração e transmissão para que funcione de maneira otimizada.

4.2 Introdução ao Fluxo de Potência

Na análise de sistemas elétricos (ELGERD, 1978) afirma, em muitas ocasiões, que enfrenta-se o problema de otimizar um critério escalar de custos ou função objetivo, C , função das variáveis de estado, de controle e/ou perturbação do sistema em questão, observando, simultaneamente, certas restrições de igualdade e/ou desigualdades para essas mesmas variáveis. Matematicamente, o problema consiste em otimizar a função escalar:

$$C = C(x, u, p) \quad (4.1)$$

onde:

$$x = \begin{bmatrix} \delta_1 \\ |V_1| \\ \dots \\ \delta_n \\ |V_n| \end{bmatrix}$$

Representa o vetor das n variáveis de estado, identificado no sistema pelo módulo da tensão da barra _{i} $|V_i|$ e pelo ângulo de potência δ_i , isto é, o ângulo da fase entre as tensões das barras.

$u = \begin{bmatrix} P_{G1} \\ Q_{G1} \\ \dots \\ P_{Gm} \\ Q_{Gm} \end{bmatrix}$

Representa o vetor das m variáveis de controle, identificado no sistema pelas “saídas” do gerador, isto é, potência ativa e potência reativa gerada.

$p = \begin{bmatrix} P_{D1} \\ Q_{D1} \\ \dots \\ P_{Dp} \\ Q_{Dp} \end{bmatrix}$

Representa o vetor das p variáveis não controladas ou de perturbações. São as demandas de potência.

Segundo Ribeiro (2005), na resolução do problema de fluxo de potência, duas variáveis possuem seu valor conhecido e duas são incógnitas. De acordo com quais variáveis são incógnitas, define-se três tipos de barras:

- barra de carga (PQ), onde P_i e Q_i são conhecidos e V_i e δ_i são calculados;
- barra de geração (PV), onde P_i e V_i são conhecidos e Q_i e δ_i são calculados;
- barra de referência ($V \delta$), onde V_i e δ_i são conhecidos e P_i e Q_i são calculados.

A função escalar deve satisfazer, simultaneamente, as restrições representadas pelas equações e/ou inequações:

$$g(x, u, p) = 0 \quad (4.2)$$

$$h(x, u, p) \leq 0 \quad (4.3)$$

O número de restrições, isto é, as dimensões dos vetores g e h , não necessita, necessariamente, ser relacionado com o número de variáveis de estado e de controle.

O conjunto de restrições de igualdade representado pela equação (4.2), é composto por duas equações para cada barra:

$$\sum_{j \in \Omega_i} P_{ij} = P_i \quad (4.4)$$

$$\sum_{j \in \Omega_i} Q_{ij} = Q_i - V_i^2 b_{shi} \quad (4.5)$$

Onde:

Ω_i conjunto das barras ligadas à barra i ;

P_{ij} fluxo de potência ativa no circuito i - j ;

Q_{ij} fluxo de potência reativa no circuito i - j ;

b_{shi} susceptância shunt na barra i .

As expressões gerais dos fluxos de potência ativa e reativa em linhas de transmissão, transformadores em fase e defasadores são:

$$P_{ij} = a_{ij}^2 \cdot V_i^2 \cdot g_{ij} - a_{ij} \cdot V_i \cdot V_j [g_{ij} \cdot \cos(\delta_{ij} + \varphi_{ij}) + b_{ij} \cdot \text{sen}(\delta_{ij} + \varphi_{ij})] \quad (4.6)$$

$$P_{ji} = V_j^2 \cdot g_{ij} - a_{ij} \cdot V_i \cdot V_j [g_{ij} \cdot \cos(\delta_{ij} + \varphi_{ij}) - b_{ij} \cdot \text{sen}(\delta_{ij} + \varphi_{ij})] \quad (4.7)$$

$$Q_{ij} = -a_{ij}^2 \cdot V_i^2 \cdot (b_{ij} + b_{shij}) - a_{ij} \cdot V_i \cdot V_j [g_{ij} \cdot \text{sen}(\delta_{ij} + \varphi_{ij}) - b_{ij} \cdot \cos(\delta_{ij} + \varphi_{ij})] \quad (4.8)$$

$$Q_{ji} = -V_j^2 \cdot (b_{ij} + b_{shij}) + a_{ij} \cdot V_i \cdot V_j [g_{ij} \cdot \text{sen}(\delta_{ij} + \varphi_{ij}) + b_{ij} \cdot \cos(\delta_{ij} + \varphi_{ij})] \quad (4.9)$$

onde:

a_{ij} tap do transformador i-j

δ_{ij} diferença angular $\delta_i - \delta_j$

φ_{ij} ângulo de defasamento no circuito i-j

g_{ij} condutância série no circuito i-j

b_{ij} susceptância série no circuito i-j

b_{shij} metade da susceptância shunt no circuito i-j

Utiliza-se $a_{ij} = 0$ e $\varphi_{ij} = 0$ para a representação de linhas de transmissão, $b_{shij} = 0$ e $\varphi_{ij} = 0$ para transformadores em fase, $b_{shij} = 0$ e $a_{ij} = 0$ para defasadores puros e $b_{shij} = 0$ para defasadores.

O conjunto de restrições de desigualdade, representado pela inequação $h(x, u, p) \leq 0$, contém as restrições operacionais de tensão, de injeção de potência ativa e reativa do sistema, conforme:

$$V_i^{\min} \leq V_i \leq V_i^{\max} \quad (4.10)$$

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (4.11)$$

$$Q_i^{\min} \leq Q_i \leq Q_i^{\max} \quad (4.12)$$

Se a função custo, aqui identificada como função objetivo e, todas as equações e/ou inequações que compõem as restrições forem lineares, então o trata-se de um problema de programação linear. Em problemas de fluxo de potência, a função objetivo e as restrições são, quase sempre, em sua totalidade, funções não

lineares tratando-se assim, de um problema de programação não linear. Para a solução do mesmo, usa-se métodos numéricos iterativos de solução de equações não lineares.

O problema de fluxo de potência pode ser dividido em três categorias:

1. Despacho econômico ótimo, desprezando-se as perdas na linha;
2. Despacho econômico ótimo, considerando-se as perdas na linha;
3. Funcionamento ótimo, ou fluxo de potência ótimo.

Neste sentido, este estudo dará atenção ao terceiro caso, isto é, a um problema de otimização da função objetivo, aqui representando o custo. Este deverá ser minimizado sem prejudicar o perfeito funcionamento do sistema, e atender as demandas de potência. Mas a função objetivo pode ser otimizada em função de outros critérios ou aspectos.

4.3 O Problema do Fluxo de Potência Ótimo

Existe, segundo Araújo (2005), diversos pontos factíveis para um correto funcionamento de um sistema elétrico de potência (SEP), mas alguns pontos de operação são mais vantajosos do que outros dependendo dos aspectos avaliados. Como exemplo, para diminuir as perdas do sistema pode-se distribuir a geração uniformemente pelos geradores do sistema; por outro lado, para minimizar o custo de geração, é vantagem que esta distribuição deixe de ser uniforme e passe a se concentrar nos geradores de menor custo.

Para resolver este problema, é comumente utilizado o fluxo de potência ótimo (FPO) onde, por meio de uma função objetivo, procura-se encontrar um ponto ótimo

de funcionamento para satisfazer um ou mais objetivos, estando o sistema sujeito às restrições físicas, funcionais, de confiabilidade, entre outras.

Segundo Baptista (2004), o fluxo de potência ótimo é um problema de otimização não-linear, não convexo, de grande porte, que calcula um conjunto de variáveis ótimas de estado e controle da rede, a partir dos dados de carga e dos parâmetros do sistema. O problema de FPO otimiza uma função objetivo, enquanto satisfaz um conjunto de restrições físicas e operacionais impostas pelas limitações dos equipamentos e exigências de segurança, como visto anteriormente.

Baptista (2004), afirma ainda, que o problema de FPO foi proposto por Carpentier no início da década de 60, a partir do problema de despacho econômico (DE). Historicamente, o problema de DE, resolvido pelo método dos custos incrementais iguais, foi o precursor do problema de fluxo de potência ótimo, o qual marcou o fim do período clássico do DE, que tinha sido estudado e desenvolvido ao longo de 30 anos. Assim, o problema de DE passou a ser abordado como um caso particular do FPO. Desde então, muitos trabalhos foram apresentados na tentativa de resolvê-lo.

Segundo Terra (1989), os métodos para solução do FPO podem ser reunidos em quatro grandes famílias: Programação Linear (PL), Kuhn-Tucker (KT), Gradiente (GR) e variáveis métricas (VM). Nas últimas três décadas, as soluções do problema utilizaram estas diferentes técnicas de programação matemática. No início da década de 90, o interesse na aplicação de métodos de pontos interiores, em sistemas elétricos de potência, teve um grande aumento devido ao seu desempenho e propriedades de convergência (BAPTISTA, 2004).

Oliveira e Filho (2003), afirmam que o FPO tem aplicação em diversos problemas de análise e operação de sistemas de potência, tais como a análise de

confiabilidade de geração e transmissão, análise de segurança, planejamento da expansão da geração e transmissão, e programação da geração à curto prazo.

Na grande maioria dessas aplicações, segundo Oliveira e Filho (2003), a representação linearizada (DC) do fluxo de potência tem sido adotada, devido à sua maior simplicidade e ao grau de precisão satisfatório de seus resultados. Na Figura 6 apresenta-se a estrutura funcional dos SEPs.

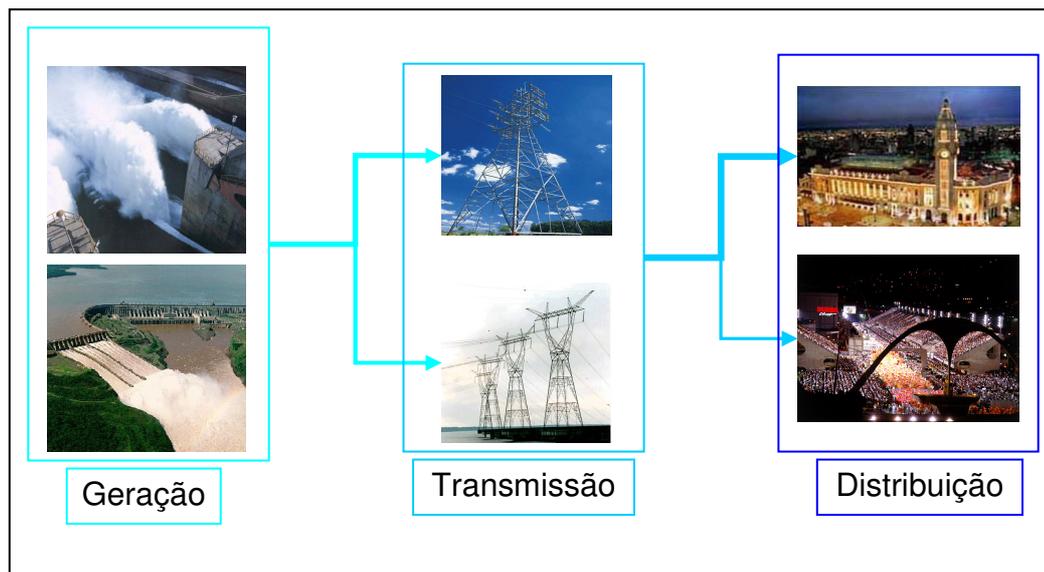


Figura 6 – Estrutura Funcional dos SEPs.

Segundo Falcão (2003), os componentes da estrutura funcional, apresentados na Figura 6 são:

- Geração: composto pelas usinas ou centrais geradoras. Essas centrais podem ser do tipo hidrelétrica, térmica (carvão, óleo ou gás natural), ou nuclear. As centrais hidrelétricas, em geral, são localizadas em pontos distantes dos centros de consumo, exigindo sistemas de transmissão complexos e em tensão elevada;
- Transmissão: constituído pelas linhas de transmissão e equipamentos auxiliares necessários para transmitir a potência produzida nas centrais

geradoras até os centros de consumo. Os sistemas de transmissão podem ser em corrente alternada (CA) ou em corrente contínua (CC);

- Distribuição: constituído pelas subestações e alimentadores responsáveis pela distribuição da energia elétrica aos consumidores industriais, comerciais e residenciais.

O modelo matemático do problema do FPO é representado por um problema de otimização formulado na próxima seção.

4.4 Formulação do Problema de Fluxo de Potência Ótimo

O problema de Fluxo de Potência Ótimo, como visto anteriormente, consiste na determinação do estado de uma rede elétrica. Maximiza ou minimiza uma função objetivo enquanto satisfaz um conjunto de restrições físicas e operacionais.

As restrições de igualdade correspondem às equações de balanço de potência ativa e reativa em cada barra da rede. As desigualdades são restrições funcionais, como o monitoramento do fluxo em linhas e limites físicos e operacionais do sistema.

O problema de Fluxo de Potência Ótimo pode ser formulado matematicamente e, genericamente, por:

$$\begin{aligned}
 \min. \quad & f(x, u, p) \\
 \text{s.a.} \quad & g_i(x, u, p) = 0, i = 1, \dots, m \\
 & h_j(x, u, p) \leq 0, j = 1, \dots, r \\
 & x_{\min} \leq x \leq x_{\max}, u_{\min} \leq u \leq u_{\max}
 \end{aligned} \tag{4.12a}$$

onde: $(x, u, p) \in R^n$ representa as variáveis definidas anteriormente; $f(x, u, p)$ representa o índice de desempenho do sistema; $g(x, u, p) = 0$ representa as

equações do fluxo de potência; $h(x, u, p) \leq 0$ representa as restrições funcionais, isto é, limites de potência ativa e reativa nas linhas de transmissão e transformadores, limites de injeção de potência reativa nas barras de controle de tensão e injeção de potência ativa na barra de referência ; $x_{\min} \leq x \leq x_{\max}$ e $u_{\min} \leq u \leq u_{\max}$ representam limites nas variáveis de estado e de controle, respectivamente.

4.4.1 Restrições de Igualdade

As principais restrições de igualdades utilizadas em problemas de Fluxo de Potência Ótimo, organizadas por (ARAUJO, 2005), são reproduzidas a seguir em sua forma geral.

4.4.1.1 Equações de Balanço de Potência Ativa

$$\sum_{j \in \Omega_i} P_{ij} = PG_i - FC_i \cdot (1 - A_i - B_i + A_i \cdot V_i + B_i \cdot V_i^2) \cdot PL_i + PA_i \quad (4.13)$$

Onde:

Ω_i conjunto de barras ligadas à barra i ;

P_{ij} fluxo ativo no circuito $i - j$;

PG_i potência ativa gerada na barra i ;

FC_i fator de carga (em pu) na barra i ;

PL_i carga ativa na barra i ;

A_i fator de carga (em pu) da variação linear da carga ativa em relação à tensão;

B_i fator de carga (em pu) da variação quadrática da carga ativa em relação à tensão;

V_i módulo de tensão na barra i ;

PA_i injeção de potência ativa na barra i .

As expressões dos fluxos P_{ij} e P_{ji} correspondem às equações (4.6) e (4.7), respectivamente. Nas equações apresentadas, é incluído um fator de variação das cargas em relação à tensão. Não considerar esta hipótese é equivalente a declarar $A_i = B_i = C_i = D_i = 0$ em cada barra da rede.

4.4.1.2 Equações de Balanço de Potência Reativa

$$\sum_{j \in \Omega_i} Q_{ij} = QG_i + QC_i - QI_i + V_i^2 \cdot b_{shi} - FC_i \cdot (1 - C_i - D_i + C_i \cdot V_i + D_i \cdot V_i^2) \cdot QL_i \quad (4.14)$$

Onde:

Ω_i conjunto de barras ligadas à barra i ;

Q_{ij} fluxo reativo no circuito i - j ;

QG_i potência reativa gerada na barra i ;

QC_i injeção de potência reativa capacitiva na barra i ;

QI_i injeção de potência reativa indutiva na barra i ;

V_i módulo de tensão na barra i ;

b_{shi} shunt na barra i ;

FC_i fator de carga (em pu) da barra i ;

QL_i carga reativa da barra i ;

C_i fator de carga (em pu) da variação linear da carga reativa em relação à tensão;

D_i fator de carga (em pu) da variação quadrática da carga reativa em relação à tensão.

As expressões dos fluxos Q_{ij} e Q_{ji} correspondem às equações (4.8) e (4.9), respectivamente.

4.4.1.3 Intercâmbio Líquido entre Áreas

$$IT_k = \sum_{I_1} P_{ij} + \sum_{I_2} P_{ji} - \sum_{I_3} P_{ij} - \sum_{I_4} P_{ji} \quad (4.15)$$

onde:

IT_k intercâmbio líquido na área k;

P_{ij} fluxo ativo no circuito i-j;

I_1 conjunto de circuitos de interligação i-j tal que

1. a medição é realizada no nó i
2. o nó i pertence a área k

I_2 conjunto de circuitos de interligação i-j tal que

1. a medição é realizada no nó j
2. o nó j pertence a área k

I_3 conjunto de circuitos de interligação i-j tal que

1. a medição é realizada no nó i
2. o nó i não pertence a área k

I_4 conjunto de circuitos de interligação i-j tal que

1. a medição é realizada no nó j
2. o nó j não pertence a área k

4.4.2 Restrições de Desigualdade

As restrições de desigualdade são as limitações impostas a uma variável, ou a um conjunto de variáveis do sistema. Em relação à sua função, elas podem ser classificadas em três grandes grupos:

4.4.2.1 Restrições Físicas

São incluídas neste grupo as restrições impostas pelas limitações de capacidade dos componentes do sistema. Exemplos destas limitações podem ser: limites máximo e mínimo de geração de potência ativa e reativa das unidades geradoras, limites nos valores dos “tapes”, limites de fluxo nas linhas, etc.

4.4.2.2 Restrições Operacionais

A operação do sistema impõe limites que devem ser considerados na modelagem. Alguns exemplos destas restrições são: limites mínimos e máximos da magnitude da tensão nas barras, limites originados pela taxa de tomada de carga das unidades geradoras, defasamento angular máximo entre barras, etc.

4.4.2.3 Restrições de Segurança

As restrições de segurança representam um grupo de restrições relacionadas a um conjunto de contingências determinadas pela análise de segurança. Em

relação à sua representação matemática no problema de otimização, as restrições podem ser divididas em duas classes: Restrições simples e funcionais.

4.4.2.4 Restrições Simples ou de Canalizações

Este tipo de restrição consiste de limites nas variáveis dependentes, representadas por $x_{\min} \leq x \leq x_{\max}$ e nas variáveis independentes, representadas por

$$u_{\min} \leq u \leq u_{\max}$$

4.4.2.5 Restrições Funcionais

Este tipo de restrição é modelado como uma função das variáveis de otimização, representados por $h(x, u, p) \leq h_{\max}$. Em geral, este tipo de restrição impõe condições mais severas à convergência dos métodos de otimização do que as restrições simples.

As restrições funcionais podem ser transformadas em uma restrição de igualdade e uma restrição de canalização, com o objetivo de facilitar a implementação computacional.

Como exemplo, a restrição $h(x, u, p) \leq h_{\max}$, pode ser transformada em uma restrição do tipo $h_{\min} \leq h(x, u, p) \leq h_{\max}$.

Então cria-se uma variável auxiliar y de modo que $y = h(x, u, p)$, e as novas equações são:

$$y - h(x, u, p) = 0 \quad \text{Restrição de Igualdade} \quad (4.16)$$

$$h_{\min} \leq y \leq h_{\max} \quad \text{Restrição de Canalização}$$

4.4.3 Função Objetivo

A função objetivo representa a variável ou conjunto de variáveis que se deseja otimizar. As classes de funções objetivo podem incluir funções lineares e não-lineares. Dependendo do tipo de aplicação, o problema pode ser formulado combinando uma ou mais funções objetivo ao mesmo tempo. A modelagem matemática das funções objetivo mais utilizadas, também relatada em Araújo (2005), é reproduzida a seguir.

4.4.3.1 Mínimo Custo de Geração Ativa

$$f = \sum_{i \in I_G} CP_i \cdot PG_i \quad (4.17)$$

onde:

I_G conjunto de geradores controláveis de potência ativa;

CP_i custo de geração ativa do gerador i ;

PG_i geração ativa do gerador i .

4.4.3.2 Mínima Injeção de Potência Reativa

$$f = \sum_{i \in IQ_c} QC_i + \sum_{i \in IQ_i} QI_i \quad (4.18)$$

Onde:

IQ_c conjunto de barras candidatas à injeção de potência reativa capacitiva;

QC_i potência reativa capacitiva injetada na barra i ;

IQ_i conjunto de barras candidatas à injeção de potência reativa indutiva;

QI_i potência reativa indutiva injetada na barra i .

4.4.3.3 Mínima Injeção de Potência Ativa

$$f = \sum_{i \in I_p} PA_i \quad (4.19)$$

Onde:

I_p conjunto de barras candidatas à injeção de potência ativa;

PA_i potência ativa injetada na barra i .

4.4.3.4 Mínima Perda Ativa

$$f = \sum_{i,j \in I_C} (P_{ij} + P_{ji}) \quad (4.20)$$

Onde:

I_C conjunto de circuitos do sistema;

P_{ij}, P_{ji} fluxo ativo nos circuitos $i-j, j-i$.

Note que $P_{ij} + P_{ji}$ é igual a perda no circuito $i-j$. As expressões dos fluxos P_{ij} e P_{ji} são relativas às fórmulas (4.6) e (4.7), respectivamente.

4.4.3.5 Mínimo Corte de Carga

$$f = \sum_{i \in I_C} (1 - FC_i) \cdot PL_i \quad (4.21)$$

Onde:

I_C conjunto de barras de carga;

FC_i fração de carga efetiva na barra i (em pu);

PL_i carga original da barra i .

Observar que $FC_i \cdot PL_i$ representa a carga efetiva na barra i , enquanto que $(1 - FC_i) \cdot PL_i$ representa o corte de carga nesta barra.

4.4.3.6 Mínimo Desvio de Potência Ativa Gerada

$$f = \frac{1}{2} \sum_{i \in I_G} \rho \cdot (PG_i - \overline{PG_i})^2 \quad (4.22)$$

Onde:

I_G conjunto de geradores controláveis de potência ativa;

ρ peso associado ao desvio de potência ativa;

PG_i geração de potência ativa do gerador i ;

$\overline{PG_i}$ geração de potência ativa inicial no gerador i .

4.4.3.7 Mínimo Desvio de Ângulo de Defasamento

$$f = \frac{1}{2} \sum_{i,j \in I_\varphi} \rho \cdot (\varphi_{ij} - \overline{\varphi_{ij}})^2 \quad (4.23)$$

Onde:

I_φ conjunto de circuitos com controle de ângulo de defasamento;

ρ peso associado ao desvio de ângulo de defasamento;

φ_{ij} ângulo de defasamento no circuito i - j ;

$\overline{\varphi_{ij}}$ ângulo de defasamento inicial no circuito i - j .

4.4.3.8 Mínimo Desvio de Tensão

$$f = \frac{1}{2} \sum_{i \in I} \rho \cdot (V_i - \bar{V}_i)^2 \quad (4.24)$$

Onde:

- I conjunto de barras do sistema;
- ρ peso associado ao desvio de tensão;
- V_i tensão da barra i ;
- \bar{V}_i tensão inicial da barra i .

4.4.3.9 Mínimo Desvio de Tap

$$f = \frac{1}{2} \sum_{i,j \in I_T} \rho \cdot (a_{ij} - \bar{a}_{ij})^2 \quad (4.25)$$

onde:

- I_T conjunto de transformadores controláveis;
- ρ peso associado ao desvio de tap;
- a_{ij} tap do transformador i - j ;
- \bar{a}_{ij} tap inicial do transformador i - j .

4.4.3.10 Mínimo Desvio de Intercâmbio

$$f = \frac{1}{2} \sum_{i \in J_I} \rho \cdot (IT_i - \bar{IT}_i)^2 \quad (4.26)$$

onde:

- J_I conjunto de áreas de intercâmbio;
- ρ peso associado ao desvio de intercâmbio entre áreas;
- IT_i intercâmbio da área i ;

\overline{IT}_i intercâmbio inicial da área i .

4.4.3.11 Mínimo Desvio do Ponto de Operação

Esta função objetivo é uma combinação das funções objetivo de desvio apresentadas anteriormente.

4.4.4 Função Lagrangeana

A função Lagrangeana é um artifício matemático utilizado para transformar um problema de otimização, sujeito a apenas restrições de igualdade, em um problema de otimização irrestrita. Este artifício consiste em adicionar as restrições de igualdade à função objetivo, multiplicada por um valor λ (multiplicador de Lagrange), formando uma nova função objetivo, a qual é denominada de função Lagrangeana.

Seja um problema de otimização com apenas restrições de igualdade:

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

Transformando as restrições de desigualdade em restrições de igualdade, fazendo uso do vetor de variáveis auxiliares y e x .

$$\begin{aligned}
& \text{min. } f(x) \\
& \text{s.a. } g(x) = 0 \\
& \quad y - h(x) = 0 \\
& \quad y_{\min} \leq y \leq y_{\max} \\
& \quad x_{\min} \leq x \leq x_{\max}
\end{aligned}$$

Fazendo $x = \begin{bmatrix} x \\ y \end{bmatrix}$ e acrescentando $y - h(x) = 0$ no conjunto das restrições

$h(x) = 0$ tem-se:

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

Transformando as restrições de canalização em restrições de igualdade pelo uso dos vetores de variáveis de folga S_{low} e S_{up} :

$$\begin{aligned}
& \text{min. } f(x) \\
& \text{s.a. } g(x) = 0 \\
& \quad x - S_{low} - x_{\min} = 0 \\
& \quad x + S_{up} - x_{\max} = 0 \\
& \quad S_{low} \geq 0 \\
& \quad S_{up} \geq 0
\end{aligned}$$

A função lagrangeana deste problema é definida pela equação:

$$L(x, \lambda, \pi, s) = \begin{cases} f(x) + \lambda^t g(x) + \pi_{low} (x - S_{low} - x_{\min}) + \pi_{up} (x + S_{up} - x_{\max}) \\ \mu \log(S_{low}) + \mu \log(S_{up}) \end{cases} \quad (4.27)$$

4.4.5 Funções Penalidade

Também conhecido como método das funções de penalidade externa, este é um procedimento que visa aproximar problemas de otimização com restrições, por problemas de otimização sem restrições. Essa aproximação é obtida, adicionando-se à função objetivo uma parcela que estabelece uma grande penalidade pela violação das restrições. Esta parcela está associada a um parâmetro μ que determina quão severa é a penalidade, se as restrições forem violadas.

4.4.6 As condições de Otimalidade

As condições de otimalidade citadas no capítulo 2 descrevem as equações de otimalidade de primeira ordem, conhecidas como condições de Karush-Kuhn-Tucher (KKT) como afirma Wright (1997), citada em (ARAUJO, 2005). Estas condições devem ser satisfeitas em qualquer ponto ótimo restrito, local ou global, dos problemas de programação linear e da maioria dos problemas de programação não-linear. As condições de KKT formam a base para o desenvolvimento de muitos algoritmos computacionais e são utilizados como critérios de convergência de vários métodos. Estas equações dão as condições necessárias para que o ponto seja considerado candidato a ótimo, ou mínimo local, como afirma Castronouvo (2001), também citado por (ARAUJO, 2005).

Para o problema de otimização não-linear representado pela função Lagrangeana do item 4.3.5, a qual apresenta restrições de igualdade e desigualdade, as condições de KKT degeneradas ($\mu \neq 0$ e $\mu \rightarrow 0$ no processo

iterativo) têm como objetivo evitar problemas numéricos no método de Newton para a resolução do sistema linear.

4.5 Modelagem do Fluxo de Potência Linearizado

O modelo a ser considerado, apresentado por Monticelli (1983), leva em conta a alocação de potência ativa, com restrições de geração e transmissão. Utiliza-se, neste trabalho, um modelo de fluxo de potência linearizado DC, isto é, fluxo de carga com corrente contínua (CC), com limites no fluxo de carga ativa.

Na construção do modelo, são considerados os elementos existentes no sistema basicamente os geradores, as cargas e as linhas de transmissão que permitem o transporte de energia.

Segundo Rodrigues (2002), o fluxo de potência linearizado é baseado no acoplamento entre os ângulos das tensões nodais e as potências ativas injetadas nas barras e apresenta resultados melhores para sistemas com tensões elevadas, nos quais as quedas de tensão não são muito relevantes.

O modelo DC não leva em conta as magnitudes das tensões das barras, as potências reativas e os *taps* dos transformadores. Também não são considerados os elementos *shunt* das linhas.

Este modelo só é válido nos casos em que as seguintes aproximações são válidas, pois, caso contrário, os valores obtidos não seriam úteis.

$$R \ll X \quad (4.28)$$

$$R \sim 0 \quad (4.29)$$

onde:

- R Representa a resistência da linha;
 X Representa a reatância.

4.5.1 Formulação Matricial

O modelo linearizado (MONTICELLI, 1983) pode ser expresso matricialmente por uma equação do tipo:

$$I = Y.E \quad (4.30)$$

onde:

- I Representa o vetor de injeção de corrente, cujas componentes são $I_i (i = 1, n)$;
- Y Matriz admitância nodal;
- E Vetor das tensões nodais, cujas componentes são $E_i = V_i e^{j\theta_k}$

Para uma rede de transmissão sem transformadores em fase ou defasadores, os fluxos de potência ativa nos ramos da rede são dados por:

$$P_{ik} = x_{ik}^{-1} \theta_{ik} \quad (4.31)$$

em que x_{ik} é a reatância equivalente de todas as linhas em paralelo que existem no ramo $(i - k)$.

A injeção de potência ativa na barra i é igual à soma dos fluxos que saem da barra, ou seja:

$$P_i = \sum_{k \in \Omega_i} x_{ik}^{-1} \theta_{ik} \quad (i = 1, n) \quad (4.32)$$

sendo n o número de barras do sistema.

A expressão 4.32 pode ser colocada na forma:

$$P_i = \left(\sum_{k \in \Omega_i} x_{ik}^{-1} \theta_i \right) + \left(\sum_{k \in \Omega_i} x_{ik}^{-1} \theta_k \right) \quad (4.33)$$

que, por sua vez, admite uma representação matricial da forma:

$$P = B' \theta \quad (4.34)$$

em que:

θ é o vetor dos ângulos das tensões nodais;

P é o vetor das injeções líquidas de potência ativa;

B' é a matriz do tipo admitância nodal, cujos elementos são:

$$B'_{ik} = -x_{ik}^{-1} \quad (4.35)$$

$$B'_{ii} = \sum_{k \in \Omega_i} x_{ik}^{-1} \quad (4.36)$$

A matriz B' que aparece na equação 4.34 é singular, pois a soma das componentes de P é nula, uma vez que as perdas de transmissão foram desprezadas, ou seja, a injeção de potência de uma barra qualquer poder ser obtida a partir da soma algébrica das demais. Para que este problema seja resolvido, adota-se uma barra como referência angular ($\theta_i = 0$). Desta forma, o sistema passa a ser não-singular com dimensão $(n-1)$ e os ângulos das $(n-1)$ barras restantes podem ser determinados a partir das injeções de potência especificadas nessas $(n-1)$ barras.

Uma vez escolhida a barra de referência, elimina-se as respectivas linha e coluna da matriz admitância e a respectiva linha dos vetores de ângulos e de potências.

Os ângulos nodais, em radianos, são obtidos da seguinte forma:

$$\theta = B^{-1}.P \quad (4.37)$$

Obtidos os valores dos ângulos das barras e conhecendo-se as reatâncias das linhas, os fluxos de potência ativa nas linhas podem ser calculados pela equação 4.31.

4.5.2 Representação das Perdas no Modelo DC

A representação da injeção de potência complexa nas barras de uma linha no modelo DC, apresentada por Monticelli (1983), é resumida a seguir:

$$S_i^* = P_i - jQ_i = E_i^* I_i \quad (4.38)$$

Separando as partes real e imaginária, têm-se as injeções de potência ativa e reativa nas barras, como foram representadas nas equações 4.6 e 4.8.

4.5.3 Matriz de Conexão de um Sistema Elétrico

Segundo Pereira e Pinto (1985), citado em (RODRIGUES, 2002), a matriz de conexão de um sistema dá idéia de como este sistema responde às variações das potências injetadas nas barras e pode ser muito útil quando do planejamento da expansão do sistema.

No sistema de três barras da figura 7, as quedas de tensão sobre os elementos são dadas por:

$$v_{b1} = V_1 - V_2 \quad (4.39)$$

$$v_{b2} = V_1 - V_3 \quad (4.40)$$

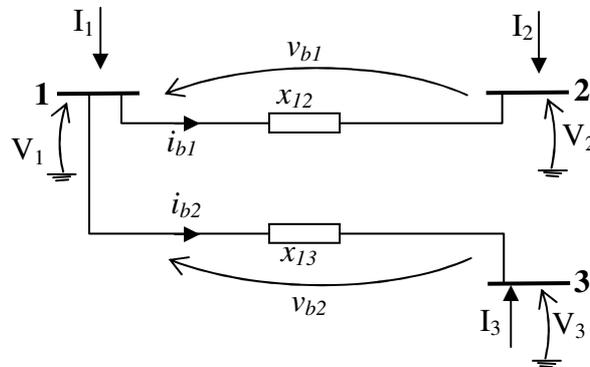


Figura 7 – Sistema de três barras.

A tensão em cada ramo pode ser então, representada, por:

$$v_b = AV_{bus} \quad (4.41)$$

onde A é a matriz de incidência ramos-barra e, para esta rede, é dada por:

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad (4.42)$$

em que na sua construção tem-se:

- 1 se a barra de origem do elemento l é a barra i ;
- 1 se a barra de chegada do elemento l é a barra i ;
- 0 se o elemento l não incidir na barra i ;

e V_{bus} é o vetor das tensões nodais representado por:

$$V_{bus} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \quad (4.43)$$

As correntes nos ramos são:

$$i_{b1} = \frac{V_1 - V_2}{x_{12}} \quad (4.44)$$

$$i_{b2} = \frac{V_1 - V_3}{x_{13}} \quad (4.45)$$

O que pode ser genericamente escrito como:

$$i_b = \frac{1}{x} v_b \quad (4.46)$$

ou:

$$i_b = b_{prim} v_b \quad (4.47)$$

em que b_{prim} é a matriz admitância primitiva da rede e é uma matriz diagonal cujos elementos são as admitâncias das linhas. A dimensão desta matriz é $b \times b$, sendo b o número de linhas da rede.

Desta forma conclui-se que as correntes injetadas nas barras são:

$$I_{bus} = Y_{bus} V_{bus} \quad (4.48)$$

Substituindo 4.41 em 4.47, tem se:

$$i_b = b_{prim} A V_{bus} \quad (4.49)$$

Lembrando que:

$$V_{bus} = Z_{bus} I_{bus} \quad (4.50)$$

As correntes nos ramos podem ser calculadas através de:

$$i_b = (b_{prim} A Z_{bus}) I_{bus} \quad (4.51)$$

O produto $(b_{prim} A Z_{bus})$ é chamado matriz de conexão da rede, ou seja:

$$C = b_{prim} A Z_{bus} \quad (4.52)$$

Analisando-se a equação 4.50, nota-se que:

- O vetor i_b é análogo ao vetor de fluxos de potência nas linhas;
- O vetor I_{bus} é análogo ao vetor de potências injetadas nas barras.

Deve-se, entretanto, lembrar que a matriz Z_{bus} é singular e, por isso, não possui inversa; sendo assim, é necessário reduzir a sua ordem para que seja possível a inversão.

Observando as analogias anteriores relacionadas, observa-se que os fluxos de potência ativa nos ramos podem ser escritos conforme a expressão:

$$P_b = C.P_{bus} \quad (4.53)$$

Visando a melhor compreensão de cada etapa do processo de construção, faz-se uso de um pequeno exemplo de uma rede elétrica.

Exemplo 4.1 Seja uma rede elétrica com quatro barras, com três barras de geração que podem representar usinas hidrelétricas ou termoelétricas e uma barra de carga que pode representar o consumo das cidades ou de grandes indústrias. Uma possível representação deste problema é dada na Figura 8.

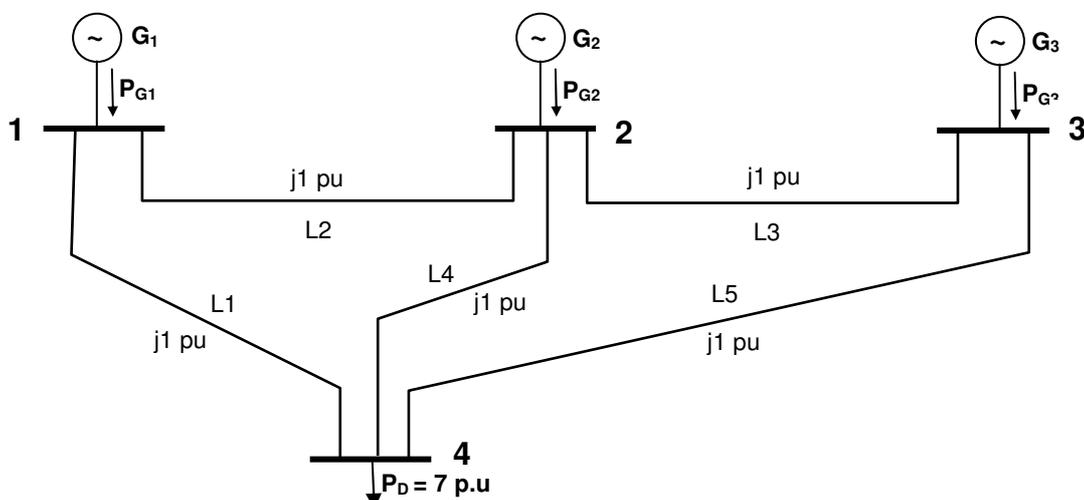


Figura 8 – Representação da rede elétrica do Exemplo 4.1.

Neste problema, pretende-se efetuar o despacho ótimo dos três geradores para atender à carga na barra 4. Deve-se observar, entretanto, que as linhas, neste exemplo, têm resistência nula, o que significa que as perdas de transmissão não são consideradas.

Dados do sistema:

- Equações de custo dos geradores:

$$C_1 = 0,25P_{G1}^2 + 2P_{G1} + 1 \quad (4.54)$$

$$C_2 = 0,5P_{G2}^2 + 3P_{G2} + 2 \quad (4.55)$$

$$C_3 = 0,75P_{G3}^2 + 4P_{G3} + 3 \quad (4.56)$$

- Limites dos geradores:

$$0,8 \leq P_{G1} \leq 6 \text{ p.u.} \quad (4.57)$$

$$0,4 \leq P_{G2} \leq 4 \text{ p.u.} \quad (4.58)$$

$$0,6 \leq P_{G3} \leq 5 \text{ p.u.} \quad (4.59)$$

- Fluxos máximos nas linhas:

$$-3,6 \leq F_l \leq 3,6 \text{ p.u.} \quad (4.60)$$

As equações de custo apresentadas em 4.54, 4.55 e 4.56 são equações quadráticas e o gráfico das curvas é apresentado como uma função polinomial de grau 2. Modelos lineares (retas) são mais simples e flexíveis e permitem, com boa qualidade de resultados, o uso da Programação Linear como visto no capítulo 2. Para linearizar as curvas usa-se o método *piecewise*, que consiste em desdobrar o gerador original em vários novos geradores.

Desdobrando cada um dos geradores em três novos geradores, com o objetivo de linearizar, tem-se as curvas de custo mostradas no Gráfico 2. Cada uma das curvas foi dividida em três partes e linearizada. Os geradores, cujas curvas eram quadráticas, representados pelas equações anteriores, terão, agora novas equações de custo, sendo elas lineares.

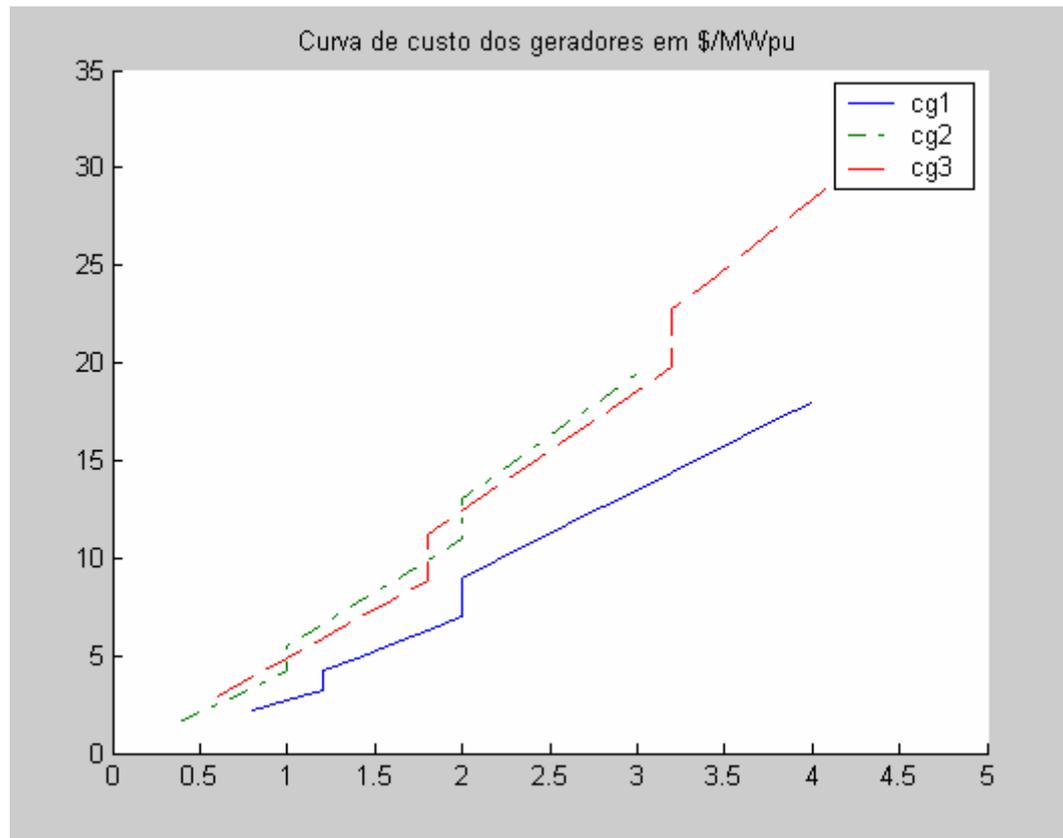


Gráfico 2 – Curvas de custo dos geradores desdobrados.

As novas curvas de custo serão representadas pelas equações:

- Gerador 1:

$$C_1^A = 2,7P_{G1}^A \quad (4.61)$$

$$C_1^B = 3,5P_{G1}^B \quad (4.62)$$

$$C_1^C = 4,5P_{G1}^C \quad (4.63)$$

- Gerador 2:

$$C_2^A = 4,2P_{G2}^A \quad (4.64)$$

$$C_2^B = 5,5P_{G2}^B \quad (4.65)$$

$$C_2^C = 6,5P_{G2}^C \quad (4.66)$$

- Gerador 3:

$$C_3^A = 4,9P_{G3}^A \quad (4.67)$$

$$C_3^B = 6,2P_{G3}^B \quad (4.68)$$

$$C_3^C = 7,1P_{G3}^C \quad (4.69)$$

Os limites para os geradores desdobrados serão:

- Gerador 1:

$$0 \leq P_{G1}^A \leq 1,2 \text{ p.u.} \quad (4.70)$$

$$0 \leq P_{G1}^B \leq 2,0 \text{ p.u.} \quad (4.71)$$

$$0 \leq P_{G1}^C \leq 2,0 \text{ p.u.} \quad (4.72)$$

- Gerador 2:

$$0 \leq P_{G2}^A \leq 1,6 \text{ p.u.} \quad (4.73)$$

$$0 \leq P_{G2}^B \leq 1,0 \text{ p.u.} \quad (4.74)$$

$$0 \leq P_{G2}^C \leq 1,0 \text{ p.u.} \quad (4.75)$$

- Gerador 3:

$$0 \leq P_{G3}^A \leq 1,8 \text{ p.u.} \quad (4.76)$$

$$0 \leq P_{G3}^B \leq 1,3 \text{ p.u.} \quad (4.77)$$

$$0 \leq P_{G3}^C \leq 1,5 \text{ p.u.} \quad (4.78)$$

Conforme descrito anteriormente, a matriz de conexão da rede pode ser facilmente obtida pela equação $C = b_{prim}AZ_{bus}$ e, neste caso exemplo, tem-se que:

- Matriz de incidência (A)

$$A = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (4.79)$$

Matriz de impedância (Z_{prim})

$$Z = \begin{bmatrix} 0,3146 & 0 & 0 & 0 & 0 \\ 0 & 0,2372 & 0 & 0 & 0 \\ 0 & 0 & 0,1823 & 0 & 0 \\ 0 & 0 & 0 & 0,3213 & 0 \\ 0 & 0 & 0 & 0 & 0,2806 \end{bmatrix} \quad (4.80)$$

Matriz de admitância ($b_{prim} = Z_{prim}^{-1}$)

$$b_{prim} = \begin{bmatrix} 3,1786 & 0 & 0 & 0 & 0 \\ 0 & 4,2159 & 0 & 0 & 0 \\ 0 & 0 & 5,4855 & 0 & 0 \\ 0 & 0 & 0 & 3,1124 & 0 \\ 0 & 0 & 0 & 0 & 3,5638 \end{bmatrix} \quad (4.81)$$

Isto posto, fazendo alguns ajustes nas matrizes anteriores, de modo que o produto seja possível, tem-se a matriz de conexão dada por:

$$C = \begin{bmatrix} 0,5757 & 0,2558 & 0,1551 \\ 0,4243 & -0,2558 & -0,1551 \\ 0,1738 & 0,3049 & -0,4213 \\ 0,2505 & 0,4393 & 0,2663 \\ 0,1738 & 0,3049 & 0,5787 \end{bmatrix} \quad (4.82)$$

De posse da matriz de conexão de rede, é possível montar o conjunto de restrições, composto de duas condições:

- Balanço de potência ativa:

$$P_{G1}^A + P_{G1}^B + P_{G1}^C + P_{G1}^{\min} + P_{G2}^A + P_{G2}^B + P_{G2}^C + P_{G2}^{\min} + P_{G3}^A + P_{G3}^B + P_{G3}^C + P_{G3}^{\min} = L \quad (4.83)$$

- Limites de fluxo nas linhas, fornecidos por:

$$F_l^{\min} \leq CP_G \leq F_l^{\max} \quad (4.84)$$

O problema de Programação Linear será:

$$\min C_T = 2,7P_{G1}^A + 3,5P_{G1}^B + 4,5P_{G1}^C + 4,3P_{G2}^A + 5,5P_{G2}^B + 6,5P_{G2}^C + 4,9P_{G3}^A + 6,2P_{G3}^B + 7,1P_{G3}^C$$

sujeito a

$$\begin{aligned}
 & P_{G1}^A + P_{G1}^B + P_{G1}^C + 0,8 + P_{G2}^A + P_{G2}^B + P_{G2}^C + 0,4 + P_{G3}^A + P_{G3}^B + P_{G3}^C + 0,6 = 7,0 \\
 & -3,6 \leq 0,5757(P_{G1}^A + P_{G1}^B + P_{G1}^C + 0,8) + 0,2558(P_{G2}^A + P_{G2}^B + P_{G2}^C + 0,4) + 0,1551(P_{G3}^A + P_{G3}^B + P_{G3}^C + 0,6) \leq 3,6 \\
 & -3,6 \leq 0,42431(P_{G1}^A + P_{G1}^B + P_{G1}^C + 0,8) - 0,2558(P_{G2}^A + P_{G2}^B + P_{G2}^C + 0,4) - 0,1551(P_{G3}^A + P_{G3}^B + P_{G3}^C + 0,6) \leq 3,6 \\
 & -3,6 \leq 0,1738(P_{G1}^A + P_{G1}^B + P_{G1}^C + 0,8) + 0,3049(P_{G2}^A + P_{G2}^B + P_{G2}^C + 0,4) - 0,4213(P_{G3}^A + P_{G3}^B + P_{G3}^C + 0,6) \leq 3,6 \\
 & -3,6 \leq 0,2505(P_{G1}^A + P_{G1}^B + P_{G1}^C + 0,8) + 0,4393(P_{G2}^A + P_{G2}^B + P_{G2}^C + 0,4) + 0,2663(P_{G3}^A + P_{G3}^B + P_{G3}^C + 0,6) \leq 3,6 \\
 & -3,6 \leq 0,1738(P_{G1}^A + P_{G1}^B + P_{G1}^C + 0,8) + 0,3049(P_{G2}^A + P_{G2}^B + P_{G2}^C + 0,4) + 0,5787(P_{G3}^A + P_{G3}^B + P_{G3}^C + 0,6) \leq 3,6 \\
 & 0 \leq P_{G1}^A \leq 1,2 \\
 & 0 \leq P_{G1}^B \leq 2,0 \\
 & 0 \leq P_{G1}^C \leq 2,0 \\
 & 0 \leq P_{G2}^A \leq 1,6 \\
 & 0 \leq P_{G2}^B \leq 1,0 \\
 & 0 \leq P_{G2}^C \leq 1,0 \\
 & 0 \leq P_{G3}^A \leq 2,0 \\
 & 0 \leq P_{G3}^B \leq 1,0 \\
 & 0 \leq P_{G3}^C \leq 2,0
 \end{aligned}$$

Resolvendo o Problema de Programação Linear (PPL) obtêm-se os seguintes valores para as potências a serem fornecidas pelos geradores:

- Gerador 1:

$$P_{G1}^A = 1,20 \text{ p.u.} \quad (4.85)$$

$$P_{G1}^B = 0,00 \text{ p.u.} \quad (4.86)$$

$$P_{G1}^C = 2,00 \text{ p.u.} \quad (4.87)$$

- Gerador 2:

$$P_{G2}^A = 1,60 \text{ p.u.} \quad (4.88)$$

$$P_{G2}^B = 0,00 \text{ p.u.} \quad (4.89)$$

$$P_{G2}^C = 0,00 \text{ p.u.} \quad (4.90)$$

- Gerador 3:

$$P_{G3}^A = 0,40 \text{ p.u.} \quad (4.91)$$

$$P_{G3}^B = 0,00 \text{ p.u.} \quad (4.92)$$

$$P_{G3}^C = 0,00 \text{ p.u.} \quad (4.93)$$

Sendo a potência de cada gerador dada pela equação:

$$P_G = P_G^{\min} + P_G^A + P_G^B + P_G^C \quad (4.94)$$

As potências que cada gerador deverá fornecer, neste caso, serão:

$$\begin{aligned} P_{G1} &= 4,00 \text{ p.u.} \\ P_{G2} &= 2,00 \text{ p.u.} \\ P_{G3} &= 1,00 \text{ pu} \end{aligned} \quad (4.95)$$

Desta forma, os fluxos nas linhas podem ser calculados pela equação 4.52:

$$F_l = \begin{bmatrix} 0,5757 & 0,2558 & 0,1551 \\ 0,4243 & -0,2558 & -0,1551 \\ 0,1738 & 0,3049 & -0,4213 \\ 0,2505 & 0,4393 & 0,2663 \\ 0,1738 & 0,3049 & 0,5787 \end{bmatrix} \cdot \begin{bmatrix} 4,00 \\ 2,00 \\ 1,00 \end{bmatrix} \quad (4.96)$$

Os fluxos serão:

$$\begin{aligned} F_{14} &= 2,9694 \text{ p.u.} \\ F_{12} &= 1,0306 \text{ p.u.} \\ F_{23} &= 0,8839 \text{ p.u.} \\ F_{24} &= 2,1467 \text{ p.u.} \\ F_{34} &= 1,8839 \text{ p.u.} \end{aligned} \quad (4.97)$$

Substituindo os valores encontrados para os geradores na função-objetivo do problema, o custo total para este despacho será:

$$C = 21,08 \quad (4.98)$$

O custo total mínimo de operação dos geradores para atender a carga é, portanto, de 21,08 unidades monetárias.

Capítulo 5 Programação Matemática Aplicada ao Problema de Fluxo de Potência Ótimo DC

5.1 Considerações Iniciais

Um sistema de energia elétrica tem uma série de dispositivos de controle que influenciam diretamente nas condições de operação e, portanto, devem ser incluídos na modelagem do sistema para que se possa simular corretamente seu desempenho.

No capítulo anterior, foi dado um exemplo que mostrou uma rede elétrica com quatro barras, com três barras de geração que representam usinas hidrelétricas ou termoelétricas e uma quarta barra de carga representando o consumo das cidades ou de grandes indústrias. No problema, propôs-se efetuar o despacho ótimo de três geradores para atender à demanda da quarta barra com o objetivo de minimizar o custo de geração.

Neste capítulo, pretende-se mostrar a eficiência da Programação Linear em redes de energia elétrica. São utilizadas três redes: uma rede de 5 barras e 7 linhas (ELGERD, 1978), uma rede de 6 barras e 7 linhas, e a rede padrão IEEE14bus com 14 barras e 20 linhas.

Os estudos são realizados com os dados da rede de energia elétrica descritos nas Tabelas 4 e 5. A Tabela 3 relaciona cada caso investigado, a rede utilizada, o problema em questão, a função objetivo, as restrições do problema e o método de otimização utilizado.

TABELA 3
Síntese dos casos e serem investigados

Caso	Rede	Problema Investigado	Função Objetivo	Restrições	Método de Otimização
IA	5 barras 7 linhas	Despacho Econômico com restrições de desigualdades	Mínimo Custo de Geração Ativa (4.17 – pág. 75)	Fluxos nas linhas	LINDO - Simplex
IB	5 barras 7 linhas	Despacho Econômico com restrições de desigualdades	Mínimo Custo de Geração Ativa (4.17 – pág. 75)	Fluxos nas linhas	Pontos Interiores LIPSOL
IC	5 barras 7 linhas	Despacho Econômico com restrições de desigualdades	Mínimo Custo de Geração Ativa (4.17 – pág. 75)	Fluxos nas linhas	MatLab (LinProg)
II	6 barras 7 linhas	Despacho Econômico	Mínimo Custo de Geração Ativa (4.17 – pág. 75)	—	MatLab (LinProg)
III	6 barras 7 linhas	Gerenciamento de congestionamento	Mínimo Custo de Geração Ativa (4.17 – pág.75)	Fluxos nas linhas	MatLab (LinProg)
IV	6 barras 7 linhas	Mínima Injeção de potencia ativa	Mínima Injeção de Potência Ativa (4.19 – pág. 76)	Fluxos nas linhas	MatLab (LinProg)
V	6 barras 7 linhas	Gerenciamento de congestionamento	Mínimo Corte de Carga (4.21 – pág. 76)	Fluxos nas linhas	MatLab (LinProg)
VI	14 barras 20 linhas	Gerenciamento de congestionamento	Mínimo Custo de Geração Ativa (4.17 – pág. 75)	Fluxos nas linhas	MatLab (LinProg)

O Caso I é analisado e descrito detalhadamente com o objetivo de ilustrar as particularidades da formulação do problema e do método de resolução. Para os demais casos, a formulação do problema, método de resolução e resultados são apresentados sinteticamente.

5.2 Despacho Econômico com Restrições de Desigualdades – Casos

IA, IB e IC

Com o objetivo de conferir a complexidade de alguns algoritmos da Programação Linear, o PPL será resolvido três vezes, usando, conseqüentemente,

três algoritmos diferentes correspondentes aos casos IA, IB e IC. No caso IA, será usado o algoritmo do método Simplex através do *software* LINDO¹. No caso IB, será usado o algoritmo do método de pontos interiores desenvolvido por (ZHANG, 1995). No caso IC, será usado o algoritmo do *Toolbox* do MatLab².

5.2.1 Caso IA – Método Simplex

Os dados para tratamento do problema partem do exercício 7.9, página 295, tratado em (ELGERD, 1978). Neste problema admite-se que todas as linhas sejam caracterizadas por uma impedância em série de $0,111 + j0,851$ ohm/milha e que as sete linhas são de 138 kV. O fluxo de potência máximo em cada linha é de 4,5 pu. A Figura 9 exibe o sistema de cinco barras e sete linhas que servirá como ilustração para o problema proposto. Na Tabela 4 são apresentados, após cálculos, os dados da rede.

Tabela 4
Dados da Rede

Legenda das Linhas	Comprimento (Milhas)	Barra Inicial	Barra Final	X^* (pu)	Fluxos Ótimos (pu)
L1	70,4	1	2	0,3146	0.8152
L2	53,1	1	3	0,2372	4.2040
L3	40,8	2	3	0,1823	4.0632
L4	71,9	1	4	0,3213	3.9383
L5	62,8	2	5	0,2806	1.5520
L6	30,6	3	5	0,1367	-2.2328
L7	98,4	4	5	0,4397	1.3042

$$Z_{base} = \left[\frac{(138kV)^2}{100MVA} \right] \Omega \quad * X = (0,851 * Comprimento) / Z_{base}$$

Fonte: Elgerd, 1978

¹ LINDO – Linear Interactive Discrete Optimizer ®

² MatLab – Matrix Laboratory

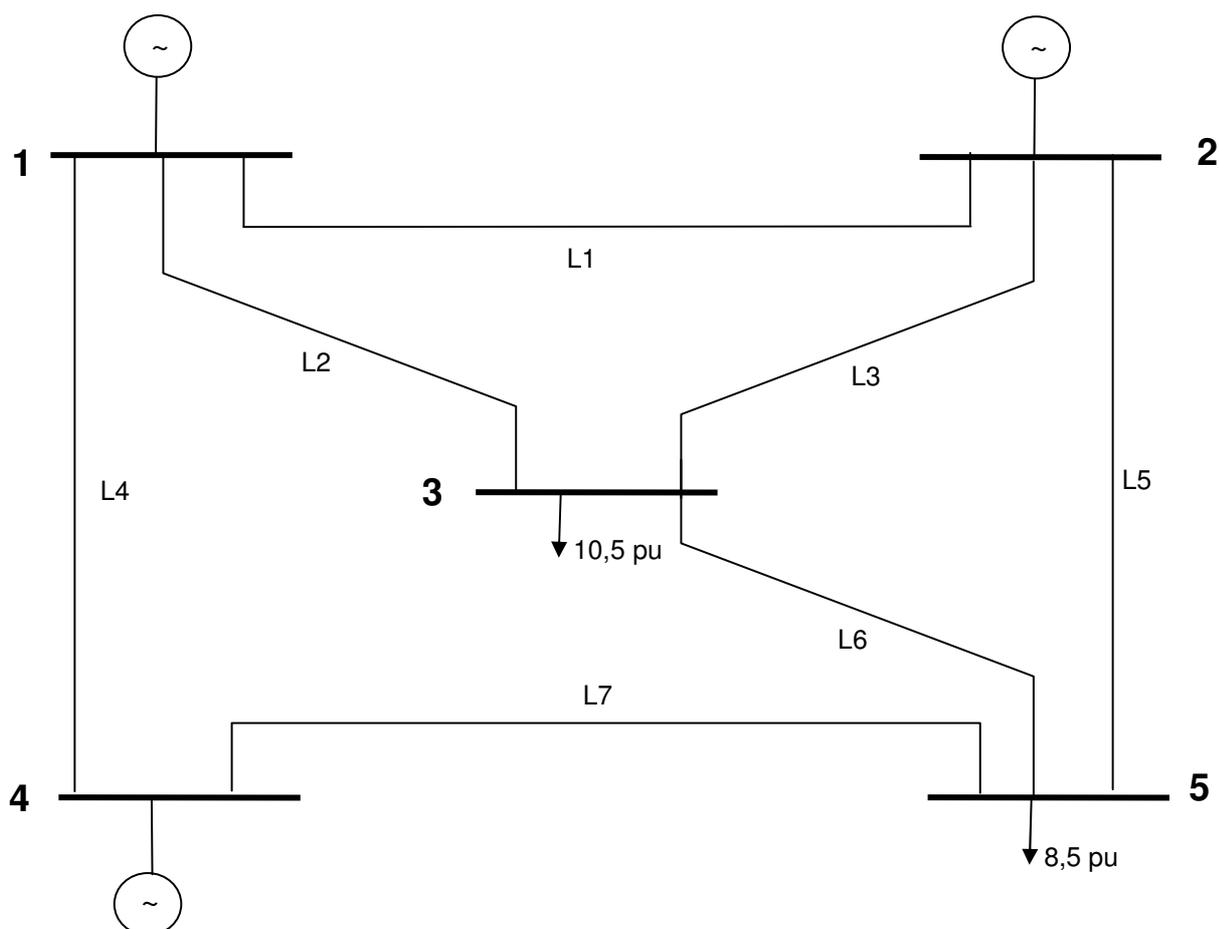


Figura 9 – Sistema de 5 Barras e 7 linhas.

Na Tabela 5, são apresentados os dados de barra.

Tabela 5
Dados da Barra

Barra	Barra Tipo		(+) P_G (pu) (-) P_D (pu)	$P_{G \min}$ (pu)	$P_{G \max}$ (pu)	Custo unitário (\$ / pu)
1	1	PV	8,9575	1,0	10,0	36,2324
2	1	PV	4,8000	1,4	9,5	21,4200
3	2	PQ	-10,5000	0	0	0
4	1	PV	5,2424	1,6	13,0	32,6004
5	3	Ref.	-8,5000	0	0	0

Neste problema, pretende-se efetuar, como o exemplo 4.1 do capítulo anterior, o despacho ótimo dos três geradores, para atender à carga na barra 3 que

consiste em 10.5 p.u. e da barra 5 que demanda uma carga de 8.5 p.u.

Desconsidera-se, neste caso, a perda de potência ativa no sistema. Com a intenção de promover os cálculos, considerou-se os mesmos custos de geração do exemplo 4.1.

Dados do sistema:

- Equações de custo dos geradores:

$$C_1 = 0,25P_{G1}^2 + 2P_{G1} + 1 \quad (5.1)$$

$$C_2 = 0,5P_{G2}^2 + 3P_{G2} + 2 \quad (5.2)$$

$$C_4 = 0,75P_{G4}^2 + 4P_{G4} + 3 \quad (5.3)$$

- Limites dos geradores:

$$1,0 \leq P_{G1} \leq 10 \text{ p.u.} \quad (5.4)$$

$$1,4 \leq P_{G2} \leq 9,5 \text{ p.u.} \quad (5.5)$$

$$1,6 \leq P_{G4} \leq 13 \text{ p.u.} \quad (5.6)$$

- Fluxos máximos nas linhas:

$$-4,5 \leq F_l \leq 4,5 \text{ p.u.} \quad (5.7)$$

Como visto no capítulo anterior, para linearizar as curvas de custo dos geradores usa-se o método *piecewise*, que consiste em desdobrar o gerador original em vários novos geradores. Têm-se, então, as novas curvas linearizadas representadas pelas equações:

- Gerador 1:

$$C_1^A = 3,25P_{G1}^A \quad (5.8)$$

$$C_1^B = 4,75P_{G1}^B \quad (5.9)$$

$$C_1^C = 6,25P_{G1}^C \quad (5.10)$$

- Gerador 2:

$$C_2^A = 6,3P_{G2}^A \quad (5.11)$$

$$C_2^B = 10,0P_{G2}^B \quad (5.12)$$

$$C_2^C = 13,2P_{G2}^C \quad (5.13)$$

- Gerador 3:

$$C_4^A = 8,95P_{G4}^A \quad (5.14)$$

$$C_4^B = 14,5P_{G4}^B \quad (5.15)$$

$$C_4^C = 20,5P_{G4}^C \quad (5.16)$$

Os limites para os geradores desdobrados serão:

- Gerador 1:

$$1,0 \leq P_{G1}^A \leq 3,0 \text{ p.u.} \quad (5.17)$$

$$0 \leq P_{G1}^B \leq 3,0 \text{ p.u.} \quad (5.18)$$

$$0 \leq P_{G1}^C \leq 3,0 \text{ p.u.} \quad (5.19)$$

- Gerador 2:

$$1,4 \leq P_{G2}^A \leq 3,4 \text{ p.u.} \quad (5.20)$$

$$0 \leq P_{G2}^B \leq 4,0 \text{ p.u.} \quad (5.21)$$

$$0 \leq P_{G2}^C \leq 3,0 \text{ p.u.} \quad (5.22)$$

- Gerador 3:

$$1,6 \leq P_{G4}^A \leq 5,0 \text{ p.u.} \quad (5.23)$$

$$0 \leq P_{G4}^B \leq 4,0 \text{ p.u.} \quad (5.24)$$

$$0 \leq P_{G4}^C \leq 4,0 \text{ p.u.} \quad (5.25)$$

A matriz de conexão da rede pode ser facilmente obtida pela equação

$C = b_{prim} A X_{bus}$, como visto anteriormente, tem-se que:

- Matriz de incidência (A)

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad (5.26)$$

- Matriz de impedância (X_{prim})

$$X_{prim} = \begin{bmatrix} 0,3146 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,2372 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,1823 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,3213 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,2806 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,1367 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,4397 \end{bmatrix} \quad (5.27)$$

- Matriz de admitância ($b_{prim} = X_{prim}^{-1}$)

$$b_{prim} = \begin{bmatrix} 3,1786 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4,2159 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5,4855 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3,1124 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3,5638 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7,3153 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2,2743 \end{bmatrix} \quad (5.28)$$

Isto posto, fazendo alguns ajustes nas matrizes anteriores, de modo que o produto seja possível, tem-se a matriz de conexão dada por:

$$C = \begin{bmatrix} 0,3160 & -0,1668 & 0,0245 & -0,1826 \\ 0,4533 & 0,0667 & -0,1139 & -0,2619 \\ 0,0445 & 0,2747 & -0,1905 & -0,0257 \\ 0,2307 & 0,1001 & 0,0894 & 0,4445 \\ 0,2715 & 0,4585 & 0,2150 & -0,1568 \\ 0,4978 & 0,4414 & 0,6955 & -0,2876 \\ -0,2307 & -0,1001 & -0,0894 & 0,5555 \end{bmatrix} \quad (5.29)$$

De posse da matriz de conexão de rede, é possível montar o conjunto de restrições, composto de duas condições:

- Balanço de potência ativa:

$$P_{G1}^A + P_{G1}^B + P_{G1}^C + P_{G1}^{\min} + P_{G2}^A + P_{G2}^B + P_{G2}^C + P_{G2}^{\min} + P_{G4}^A + P_{G4}^B + P_{G4}^C + P_{G4}^{\min} = L \quad (5.30)$$

- Limites de fluxo nas linhas, são fornecidos por:

$$F_l^{\min} \leq CP_G \leq F_l^{\max} \quad (5.31)$$

O problema de Programação Linear será:

$$\min C_T = 3,25P_{G1}^A + 4,75P_{G1}^B + 6,25P_{G1}^C + 6,3P_{G2}^A + 10,0P_{G2}^B + 13,2P_{G2}^C + 8,95P_{G4}^A + 14,5P_{G4}^B + 20,5P_{G4}^C$$

sujeito a

$$P_{G1}^A + P_{G1}^B + P_{G1}^C + 1,0 + P_{G2}^A + P_{G2}^B + P_{G2}^C + 1,4 + P_{G4}^A + P_{G4}^B + P_{G4}^C + 1,6 = 19,0$$

$$\begin{bmatrix} -4,5 \\ -4,5 \\ -4,5 \\ -4,5 \\ -4,5 \\ -4,5 \\ -4,5 \end{bmatrix} \leq \begin{bmatrix} 0,3160 & -0,1668 & 0,0245 & -0,1826 \\ 0,4533 & 0,0667 & -0,1139 & -0,2619 \\ 0,0445 & 0,2747 & -0,1905 & -0,0257 \\ 0,2307 & 0,1001 & 0,0894 & 0,4445 \\ 0,2715 & 0,4585 & 0,2150 & -0,1568 \\ 0,4978 & 0,4414 & 0,6955 & -0,2876 \\ -0,2307 & -0,1001 & -0,0894 & 0,5555 \end{bmatrix} \cdot \begin{bmatrix} P_{G1}^A + P_{G1}^B + P_{G1}^C + 1,0 \\ P_{G2}^A + P_{G2}^B + P_{G2}^C + 1,4 \\ -10,5 \\ P_{G4}^A + P_{G4}^B + P_{G4}^C + 1,6 \end{bmatrix} \leq \begin{bmatrix} 4,5 \\ 4,5 \\ 4,5 \\ 4,5 \\ 4,5 \\ 4,5 \\ 4,5 \end{bmatrix}$$

$$\begin{bmatrix} 1,0 \\ 0,0 \\ 0,0 \\ 1,4 \\ 0,0 \\ 0,0 \\ 1,6 \\ 0,0 \\ 0,0 \end{bmatrix} \leq \begin{bmatrix} P_{G1}^A \\ P_{G1}^B \\ P_{G1}^C \\ P_{G2}^A \\ P_{G2}^B \\ P_{G2}^C \\ P_{G4}^A \\ P_{G4}^B \\ P_{G4}^C \end{bmatrix} \leq \begin{bmatrix} 3,0 \\ 3,0 \\ 3,0 \\ 3,4 \\ 4,0 \\ 3,0 \\ 5,0 \\ 4,0 \\ 4,0 \end{bmatrix}$$

O PPL pode, então, ser reescrito da forma a seguir :

$$\min C_T = 3,25P_{G1}^A + 4,75P_{G1}^B + 6,25P_{G1}^C + 6,3P_{G2}^A + 10,0P_{G2}^B + 13,2P_{G2}^C + 8,95P_{G4}^A + 14,5P_{G4}^B + 20,5P_{G4}^C$$

sujeito a

$$P_{G1}^A + P_{G1}^B + P_{G1}^C + P_{G2}^A + P_{G2}^B + P_{G2}^C + P_{G3}^A + P_{G3}^B + P_{G3}^C = 15,00$$

$$0,3160(P_{G1}^A + P_{G1}^B + P_{G1}^C) - 0,1668(P_{G2}^A + P_{G2}^B + P_{G2}^C) - 0,1826(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 4,9669$$

$$0,4533(P_{G1}^A + P_{G1}^B + P_{G1}^C) + 0,0667(P_{G2}^A + P_{G2}^B + P_{G2}^C) - 0,2619(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 3,1764$$

$$0,0445(P_{G1}^A + P_{G1}^B + P_{G1}^C) + 0,2747(P_{G2}^A + P_{G2}^B + P_{G2}^C) - 0,0257(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 1,9718$$

$$0,2307(P_{G1}^A + P_{G1}^B + P_{G1}^C) + 0,1001(P_{G2}^A + P_{G2}^B + P_{G2}^C) + 0,4445(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 4,3597$$

$$0,2715(P_{G1}^A + P_{G1}^B + P_{G1}^C) + 0,4585(P_{G2}^A + P_{G2}^B + P_{G2}^C) - 0,1568(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 6,0950$$

$$0,4978(P_{G1}^A + P_{G1}^B + P_{G1}^C) + 0,4414(P_{G2}^A + P_{G2}^B + P_{G2}^C) - 0,2876(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 11,1472$$

$$-0,2307(P_{G1}^A + P_{G1}^B + P_{G1}^C) - 0,1001(P_{G2}^A + P_{G2}^B + P_{G2}^C) + 0,5555(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 3,0433$$

$$-0,3160(P_{G1}^A + P_{G1}^B + P_{G1}^C) + 0,1668(P_{G2}^A + P_{G2}^B + P_{G2}^C) + 0,1826(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 4,0331$$

$$-0,4533(P_{G1}^A + P_{G1}^B + P_{G1}^C) - 0,0667(P_{G2}^A + P_{G2}^B + P_{G2}^C) + 0,2619(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 5,8236$$

$$-0,0445(P_{G1}^A + P_{G1}^B + P_{G1}^C) - 0,2747(P_{G2}^A + P_{G2}^B + P_{G2}^C) + 0,0257(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 7,0282$$

$$-0,2307(P_{G1}^A + P_{G1}^B + P_{G1}^C) - 0,1001(P_{G2}^A + P_{G2}^B + P_{G2}^C) - 0,4445(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 4,6433$$

$$-0,2715(P_{G1}^A + P_{G1}^B + P_{G1}^C) - 0,4585(P_{G2}^A + P_{G2}^B + P_{G2}^C) + 0,1568(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 2,9050$$

$$-0,4978(P_{G1}^A + P_{G1}^B + P_{G1}^C) - 0,4414(P_{G2}^A + P_{G2}^B + P_{G2}^C) + 0,2876(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq -2,1471$$

$$+0,2307(P_{G1}^A + P_{G1}^B + P_{G1}^C) + 0,1001(P_{G2}^A + P_{G2}^B + P_{G2}^C) - 0,5555(P_{G4}^A + P_{G4}^B + P_{G4}^C) \leq 5,9567$$

$$1,0 \leq P_{G1}^A \leq 3,0$$

$$0 \leq P_{G1}^B \leq 3,0$$

$$0 \leq P_{G1}^C \leq 3,0$$

$$1,4 \leq P_{G2}^A \leq 3,4$$

$$0 \leq P_{G2}^B \leq 4,0$$

$$0 \leq P_{G2}^C \leq 3,0$$

$$1,6 \leq P_{G4}^A \leq 5,0$$

$$0 \leq P_{G4}^B \leq 4,0$$

$$0 \leq P_{G4}^C \leq 4,0$$

Resolvendo, então, o Problema de Programação Linear usando o LINDO, obtém-se os seguintes valores para as potências a serem fornecidas pelos geradores:

- Gerador 1:

$$P_{G1}^A = 3,0000 \text{ p.u.} \quad (5.32)$$

$$P_{G1}^B = 3,0000 \text{ p.u.} \quad (5.33)$$

$$P_{G1}^C = 2,3720 \text{ p.u.} \quad (5.34)$$

- Gerador 2:

$$P_{G2}^A = 3,4000 \text{ p.u.} \quad (5.35)$$

$$P_{G2}^B = 0,0000 \text{ p.u.} \quad (5.36)$$

$$P_{G2}^C = 0,0000 \text{ p.u.} \quad (5.37)$$

- Gerador 3:

$$P_{G4}^A = 3,2280 \text{ p.u.} \quad (5.38)$$

$$P_{G4}^B = 0,0000 \text{ p.u.} \quad (5.39)$$

$$P_{G4}^C = 0,0000 \text{ p.u.} \quad (5.40)$$

Sendo a potência de cada gerador dada pela equação:

$$P_G = P_G^{\min} + P_G^A + P_G^B + P_G^C \quad (5.41)$$

As potências para cada gerador, serão:

$$\begin{aligned}
 P_{G1} &= 9,3720 \text{ p.u.} \\
 P_{G2} &= 4,8000 \text{ p.u.} \\
 P_{G4} &= 4,8280 \text{ pu}
 \end{aligned}
 \tag{5.42}$$

Com isso, os fluxos nas linhas podem ser calculados pela equação 4.53:

$$C = \begin{bmatrix} 0,3160 & -0,1668 & 0,0245 & -0,1826 \\ 0,4533 & 0,0667 & -0,1139 & -0,2619 \\ 0,0445 & 0,2747 & -0,1905 & -0,0257 \\ 0,2307 & 0,1001 & 0,0894 & 0,4445 \\ 0,2715 & 0,4585 & 0,2150 & -0,1568 \\ 0,4978 & 0,4414 & 0,6955 & -0,2876 \\ -0,2307 & -0,1001 & -0,0894 & 0,5555 \end{bmatrix} \begin{bmatrix} 9,3720 \\ 4,800 \\ -10,500 \\ 4,8280 \end{bmatrix}
 \tag{5.43}$$

Os fluxos serão:

$$\begin{aligned}
 F_{12} &= 1,0218 \text{ p.u.} \\
 F_{13} &= 4,5005 \text{ p.u.} \\
 F_{23} &= 4,0923 \text{ p.u.} \\
 F_{14} &= 3,8497 \text{ p.u.} \\
 F_{25} &= 1,7295 \text{ p.u.} \\
 F_{35} &= -1,9073 \text{ p.u.} \\
 F_{45} &= 0,9783 \text{ p.u.}
 \end{aligned}
 \tag{5.44}$$

Substituindo os valores encontrados para os geradores na função objetivo do problema, o custo total para este despacho será:

$$C = 89,14$$

O custo total mínimo de operação dos geradores para atender a carga é, portanto, de 89,14 unidades monetárias.

A geração e carga, em cada uma das barras do sistema, são dadas por:

$$P_{bus} = \begin{bmatrix} 9,3720 \\ 4,8000 \\ -10,5000 \\ 4,8280 \\ -8,5000 \end{bmatrix} p.u. \quad (5.45)$$

O Quadro 1 mostra o relatório de análise de sensibilidade fornecido pelo programa LINDO. O coeficiente *slack or surplus* informa os ajustes que poderão ser feitos em cada variável de controle sem alteração do custo final da função objetivo. Por exemplo: a variável X1 que representa a potência ativa gerada pelo gerador 1, ao custo unitário de 3,25 \$/pu, pode variar de 3,00000 a 3,477897 unidades, para o mesmo custo total da função objetivo. É apresentado, também, o número de iterações gastas pelo programa para obter a solução do problema.

LP OPTIMUM FOUND AT STEP 5			
OBJECTIVE FUNCTION VALUE			
1)	89.13558		
VARIABLE	VALUE	REDUCED COST	
FO	0.000000	1.000000	
X1	3.000000	-3.000000	
X2	3.000000	-1.500000	
X3	2.372008	0.000000	
X4	3.400000	-1.409480	
X5	0.000000	2.290520	
X6	0.000000	5.490520	
X7	3.227992	0.000000	
X8	0.000000	5.550000	
X9	0.000000	11.550000	
ROW	SLACK OR SURPLUS	DUAL PRICES	
2)	3.477897	0.000000	
3)	0.000000	3.775168	
4)	0.408225	0.000000	

5)	0.653095	0.000000
6)	2.769249	0.000000
7)	6.407225	0.000000
8)	3.521913	0.000000
9)	5.522103	0.000000
10)	9.000100	0.000000
11)	8.591775	0.000000
12)	8.349905	0.000000
13)	6.230751	0.000000
14)	2.592875	0.000000
15)	5.478087	0.000000
16)	0.000000	-7.961284

NO. ITERATIONS= 5

RANGES IN WHICH THE BASIS IS UNCHANGED:

VARIABLE	CURRENT COEF	OBJ COEFFICIENT RANGES	
		ALLOWABLE INCREASE	ALLOWABLE DECREASE
FO	1.000000	INFINITY	1.000000
X1	3.250000	3.000000	INFINITY
X2	4.750000	1.500000	INFINITY
X3	6.250000	2.700000	1.500000
X4	6.300000	1.409480	INFINITY
X5	10.000000	INFINITY	2.290520
X6	13.200000	INFINITY	5.490520
X7	8.950000	4.237403	2.607501
X8	14.500000	INFINITY	5.550000
X9	20.500000	INFINITY	11.550000

ROW	CURRENT RHS	RIGHTHAND SIDE RANGES	
		ALLOWABLE INCREASE	ALLOWABLE DECREASE
2	4.966900	INFINITY	3.477897
3	3.176400	0.449140	1.267340
4	1.971800	INFINITY	0.408225
5	4.359700	INFINITY	0.653095
6	6.095000	INFINITY	2.769249
7	11.147200	INFINITY	6.407225
8	3.043300	INFINITY	3.521913
9	4.033100	INFINITY	5.522103
10	5.823700	INFINITY	9.000100
11	7.028200	INFINITY	8.591775
12	4.643300	INFINITY	8.349905
13	2.905000	INFINITY	6.230751
14	-2.147100	INFINITY	2.592875
15	5.956700	INFINITY	5.478087
16	15.000000	1.714929	2.568586

Quadro 1: Relatório de análise de sensibilidade fornecido pelo LINDO

5.2.2 Caso IB – Métodos de Pontos Interiores

Os dados adotados são idênticos aos utilizados no caso IA. Todos os problemas rodados no algoritmo de Pontos Interiores não obtiveram resposta nem aproximada com relação aos outros métodos. Em todos os casos, o algoritmo retornou que a região que limitava as restrições era inactivível. O Quadro 2 mostra o resultado parcial obtido pelo algoritmo de Pontos Interiores do pacote LIPSOL³ desenvolvido por Zhang (1995). Como pode ser verificado, a mensagem do relatório *<* Not converged *> Primal program infeasible* informa que o problema é não convergente para os dados de entrada, o que comprova o fato.

```
runlipsol
Enter problem name: ipm
Searching for ipm ...
Já existe uma subpasta ou um arquivo c:\matlab6p5\lipsol\tmp.
    1 arquivo(s) copiado(s).
Loading tmp\default.mat ...
Preprocessing ...
(m=7, n=9)
Dense columns (nnz/m > 1): 0

<<<<< This is MIIP algorithm >>>>>
min-degree ordering ... Done. CPU seconds: 0.125
calling symfct.mex* ... Done. CPU seconds: 0.031

Residuals:   Primal      Dual      U-bounds   Gap      TR_error
-----
Iter   0:  3.75e+002  5.33e-015  5.91e+002  3.41e+004  1.89e+003
Iter   1:  2.45e+001  7.58e-002  3.15e+001  2.24e+003  1.24e+002
Iter   2:  2.18e+001  1.90e-001  2.72e+001  1.97e+003  1.09e+002
Iter   3:  2.13e+001  1.27e-001  2.63e+001  3.59e+004  2.00e+003
Iter   4:  2.11e+001  1.22e-001  2.61e+001  2.47e+005  1.37e+004
Iter   5:  2.09e+001  1.37e+002  2.37e+001  1.04e+018  5.77e+016
```

³ LIPSOL – Linear Interior Points Solver

```

PROBLEMA IPM TESTE
<* Not converged *> Primal program infeasible??
[m n] = [7 9], nnz(A) = 63, nnz(L) = 28
Nonzero lower bounds exist.
Number of Upper bounds: 9
CPU seconds:  0.34 ... loading
              0.08 ... preprocessing
              0.86 ... solving
              0.00 ... postprocessing
              1.28 ... total

```

Quadro 2 : Relatório do LIPSOL

Acredita-se que sendo o problema de pequeno porte (poucas variáveis de controle) e limites das restrições pequenos, a região factível (que contém a solução) fica inviável para o uso do algoritmo, uma vez que a fronteira desta região é alcançada com considerada rapidez, impedindo a projeção das direções interiores tomadas pelo algoritmo. Situações semelhantes, isto é, incapacidade do algoritmo de pontos interiores de alcançar a solução de pequenos problemas são reportadas por (LIMA, 2004) e no *help* do LIPSOL.

5.2.3 Caso IC – MatLab: Uso da *Toolbox Optimization – LinProg*

Os dados adotados são idênticos aos utilizados no caso IA. O Quadro 3 mostra o relatório do MatLab contendo os resultados do problema. No relatório, o vetor x_1 representa o custo final da função objetivo, a variável x_2 informa que o problema teve execução normal, isto é, a solução é viável, x_3 fornece informações sobre o número de iterações e o algoritmo utilizado, x_4 , fornece os multiplicadores de Lagrange. Por exemplo, o multiplicador de Lagrange $x_4(2) = 3.7752$ indica o custo marginal, associado à restrição 2.

```
Optimization terminated successfully.
```

```
x =
```

```
3.0000  
3.0000  
2.3720  
3.4000  
0.0000  
0.0000  
3.2280  
0.0000  
0.0000
```

```
x1 =      89.1356
```

```
x2 =         1
```

```
x3 =
```

```
iterations: 8  
cgiterations: 0  
algorithm: 'lipsol'
```

```
x4 =
```

```
ineqlin: [14x1 double]  
eqlin: -7.9613  
upper: [9x1 double]  
lower: [9x1 double]
```

```
x4.ineqlin =
```

```
0.0000  
3.7752  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000  
0.0000
```

Quadro 3: Relatório fornecido pelo MatLab

Os resultados foram obtidos por uma chamada da rotina LINPROG do MatLab, a qual foi desenvolvida baseando-se na referência

Dantzig, G.B., A. Orden, and P. Wolfe, "Generalized Simplex Method for Minimizing a Linear from Under Linear Inequality Constraints," Pacific Journal Math. Vol. 5, pp. 183-195, citada no *help* do MatLab versão 6.5.0 *release 13*

Nessa documentação, é relatado que o MatLab resolve também problemas de larga escala utilizando o algoritmo LIPSOL, desenvolvido conforme as referências:

Mehrotra, S., "On the Implementation of a Primal-Dual Interior Point Method," SIAM Journal on Optimization, Vol. 2, pp. 575-601, 1992.

Zhang, Y., "Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment," Technical Report TR96-01, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD, July 1995.

5.2.4 Análise da complexidade dos algoritmos – Casos IA, IB e IC

Como verificado no capítulo 3, analisar um algoritmo ou analisar a complexidade de um algoritmo significa prever os recursos de que o algoritmo necessitará. Ocasionalmente, recursos como memória, largura de banda de comunicação ou hardware de computador são as principais preocupações, mas com frequência é o tempo de computação que se deseja medir. No problema resolvido nos Casos IA, IB e IC, a complexidade do algoritmo será analisada pelo número de iterações gastas para obter a solução ótima da função objetivo.

A Tabela 6 mostra o número de iterações gastas em cada caso.

Tabela 6
Número de iterações dos algoritmos

Caso	Algoritmo usado	Número de iterações
IA	Método Simplex	5
IB	Pontos Interiores	-
IC	MatLab – LinProg	8

Como pode ser analisado na Tabela 6, o algoritmo Simplex se mostrou mais eficiente.

5.3 Despacho Econômico – Caso II

O caso II consiste em resolver o problema do despacho econômico usando a função objetivo 4.17, descrita no capítulo anterior, página 75, cujo alvo é minimizar o custo de geração de potência ativa. A rotina LINPROG do MatLab será usada para resolver esse caso.

A Figura 10, mostra o diagrama unifilar do sistema de 6 barras, sendo 2 barras de geração, 3 de carga e uma barra do tipo 3, isto é, uma de referência e 7 linhas, usado para ilustrar o Caso II.

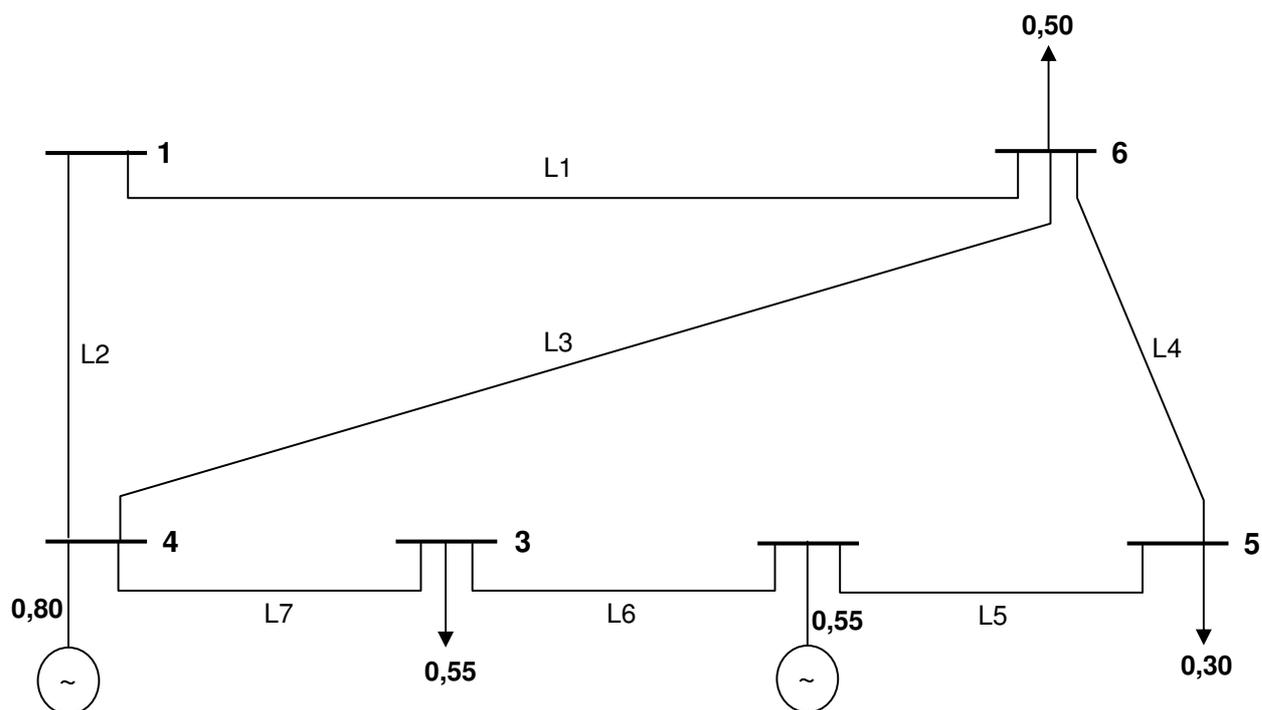


Figura 10 – Sistema de 6 barras e 7 linhas.

Os dados para modelagem, resolução e análise do problema, encontram-se nas Tabelas 7 e 8. Na Tabela 7 são apresentados os dados das linhas.

Tabela 7

Dados das linhas /Transformadores para o sistema de 6 barras e 7 linhas

Legenda das Linhas	Barra Inicial	Barra Final	X (pu)	Limites nos Fluxos (pu)
L1	1	2	0,518	0,80
L2	1	3	0,370	0,80
L3	2	3	0,407	0,30
L4	1	4	0,300	0,18
L5	2	5	0,640	0,80
L6	3	5	1,050	0,90
L7	4	5	0,133	0,80

Na Tabela 8, apresenta-se os dados de barra.

Tabela 8
Dados de barra para o sistema de 6 barras e 7 linhas

Barra	Barra Tipo		(+)P _G (pu) (-)P _D (pu)	P _{G min} (pu)	P _{G max} (pu)	Custo unitário (\$ / pu)
1	3	Ref.	-	-	-	-
2	1	PV	0,55	0	1	0,4798
3	2	PQ	-0,55	-	-	-
4	1	PV	0,80	0	1	0,6535
5	2	PQ	-0,30	-	-	-
6	2	PQ	-0,50	-	-	-

Os custos unitários de geração são mostrados na Figura 11 e dados pelas curvas:

$$Cg_2 = \begin{cases} 0,4Pg_2 + 0,25 & \text{se } 0 \leq Pg_2 \leq 0,5 \\ 0,9Pg_2 & \text{se } 0,5 \leq Pg_2 \leq 1,0 \end{cases}$$

$$Cg_4 = \begin{cases} 0,1Pg_4 + 0,35 & \text{se } 0 \leq Pg_4 \leq 0,5 \\ 0,8Pg_4 & \text{se } 0,5 \leq Pg_4 \leq 1,0 \end{cases} \quad (5.46)$$

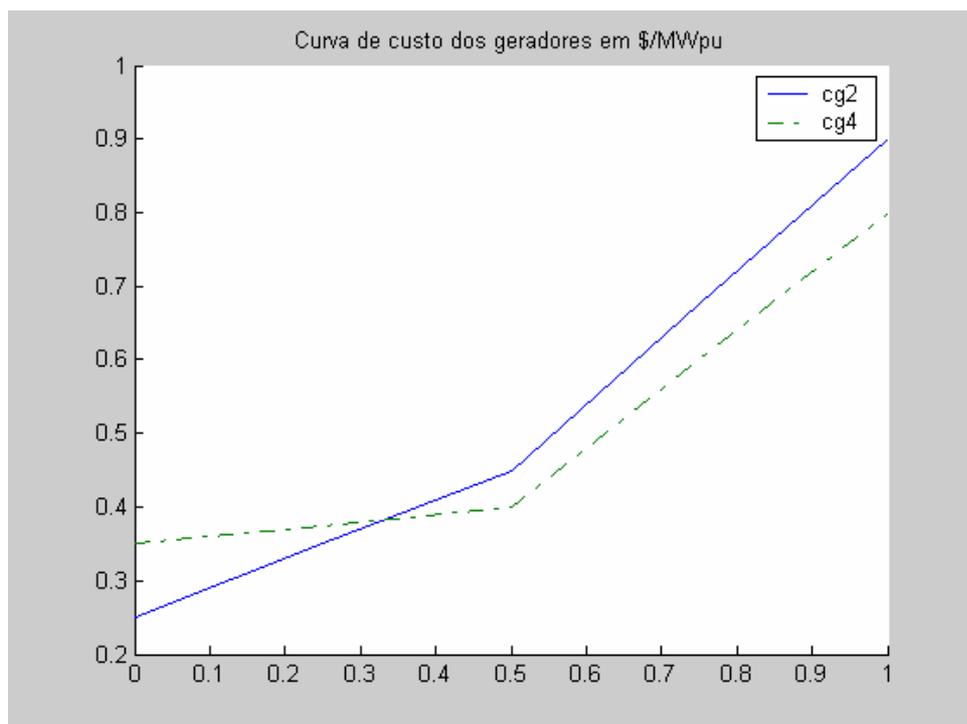


Figura 11 – Custo unitário de geração

O problema pode ser formulado como:

$$\begin{aligned} \min \sum_{i \in I_G} CP_i \cdot P_{Gi} \\ \text{s.a.} \\ \sum_{i \in I_G} P_{Gi} &= \sum_{i \in I_D} P_{Di} \\ P_{Gi}^{\min} &\leq P_{Gi} \leq P_{Gi}^{\max} \end{aligned} \quad (5.47)$$

Reescrevendo o problema formulado, tem-se:

$$\begin{aligned} \min C_T &= 0,4P_{G2}^A + 0,9P_{G2}^B + 0,1P_{G4}^A + 0,8P_{G4}^B \\ \text{s.a.} \end{aligned}$$

$$P_{G2}^A + P_{G2}^B + P_{G4}^A + P_{G4}^B = 1,35$$

$$\begin{bmatrix} 0,0 \\ 0,0 \\ 0,0 \\ 0,0 \end{bmatrix} \leq \begin{bmatrix} P_{G2}^A \\ P_{G2}^B \\ P_{G4}^A \\ P_{G4}^B \end{bmatrix} \leq \begin{bmatrix} 0,5 \\ 0,5 \\ 0,5 \\ 0,5 \end{bmatrix}$$

A Figura 12 mostra o despacho, antes e após a otimização.

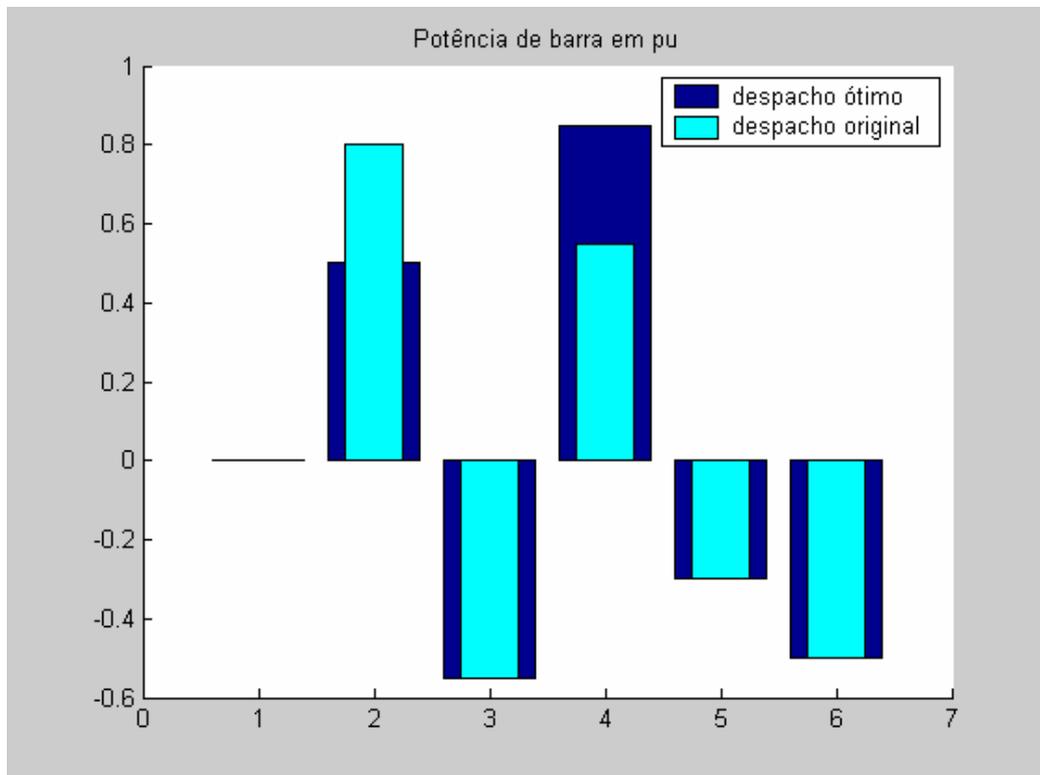


Figura 12 – Potência de Barra

Os fluxos de potência antes e após a otimização são mostrados na Figura 13.

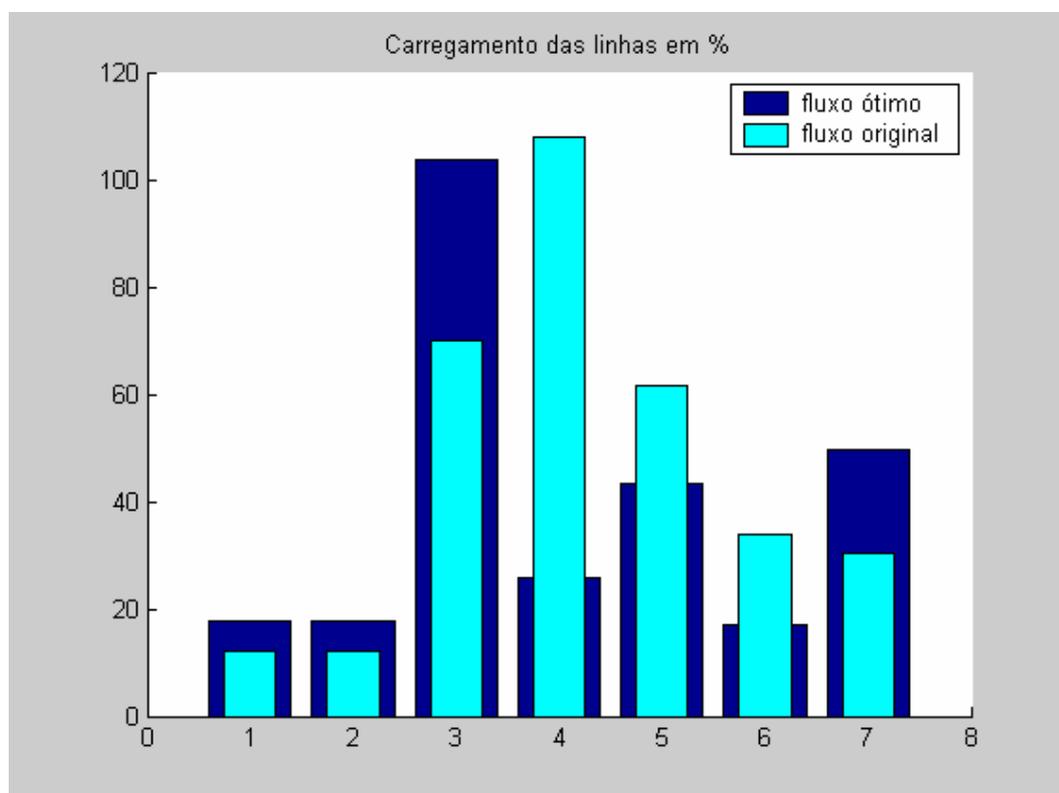


Figura 13 – Fluxos nas linhas

5.4 Gerenciamento de Congestionamento – Caso III

Para esse caso, são empregados os dados do caso II, considerando as restrições de desigualdades nos limites de fluxos nas linhas usando, também, a função objetivo 4.17, cuja finalidade é minimizar o custo de geração de potência ativa. A rotina LINPROG será usada para resolver esse caso.

O problema pode ser formulado como:

$$\min \sum_{i \in I_G} CP_i \cdot P_{Gi}$$

s.a.

$$\sum_{i \in I_G} P_{Gi} = \sum_{i \in I_D} P_{Di}$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}$$

(5.48)

Reescrevendo o problema formulado, tem-se:

$$\min C_T = 0,4P_{G2}^A + 0,9P_{G2}^B + 0,1P_{G4}^A + 0,8P_{G4}^B$$

s.a.

$$P_{G2}^A + P_{G2}^B + P_{G4}^A + P_{G4}^B = 1,35$$

$$\begin{bmatrix} -0,80 \\ -0,80 \\ -0,30 \\ -0,18 \\ -0,80 \\ -0,90 \\ -0,80 \end{bmatrix} \leq \begin{bmatrix} -0,4557 & -0,3183 & -0,3009 & -0,5394 & -0,5787 \\ -0,5443 & -0,6817 & -0,6991 & -0,4606 & -0,4213 \\ -0,0852 & 0,2146 & 0,2525 & -0,2679 & -0,3535 \\ 0,5409 & 0,1038 & 0,0484 & 0,8073 & -0,0678 \\ 0,5409 & 0,1038 & 0,0484 & -0,1927 & -0,0678 \\ 0,4591 & -0,1038 & -0,0484 & 0,1927 & 0,0678 \\ 0,4591 & 0,8962 & -0,0484 & 0,1927 & 0,0678 \end{bmatrix} \begin{bmatrix} P_{G2}^A + P_{G2}^B \\ P_{D3} \\ P_{G4}^A + P_{G4}^B \\ P_{D5} \\ P_{D6} \end{bmatrix} \leq \begin{bmatrix} 0,80 \\ 0,80 \\ 0,30 \\ 0,18 \\ 0,80 \\ 0,90 \\ 0,80 \end{bmatrix}$$

$$\begin{bmatrix} 0,0 \\ 0,0 \\ 0,0 \\ 0,0 \end{bmatrix} \leq \begin{bmatrix} P_{G2}^A \\ P_{G2}^B \\ P_{G4}^A \\ P_{G4}^B \end{bmatrix} \leq \begin{bmatrix} 0,5 \\ 0,5 \\ 0,5 \\ 0,5 \end{bmatrix}$$

A Figura 14 mostra o despacho antes e após a otimização.

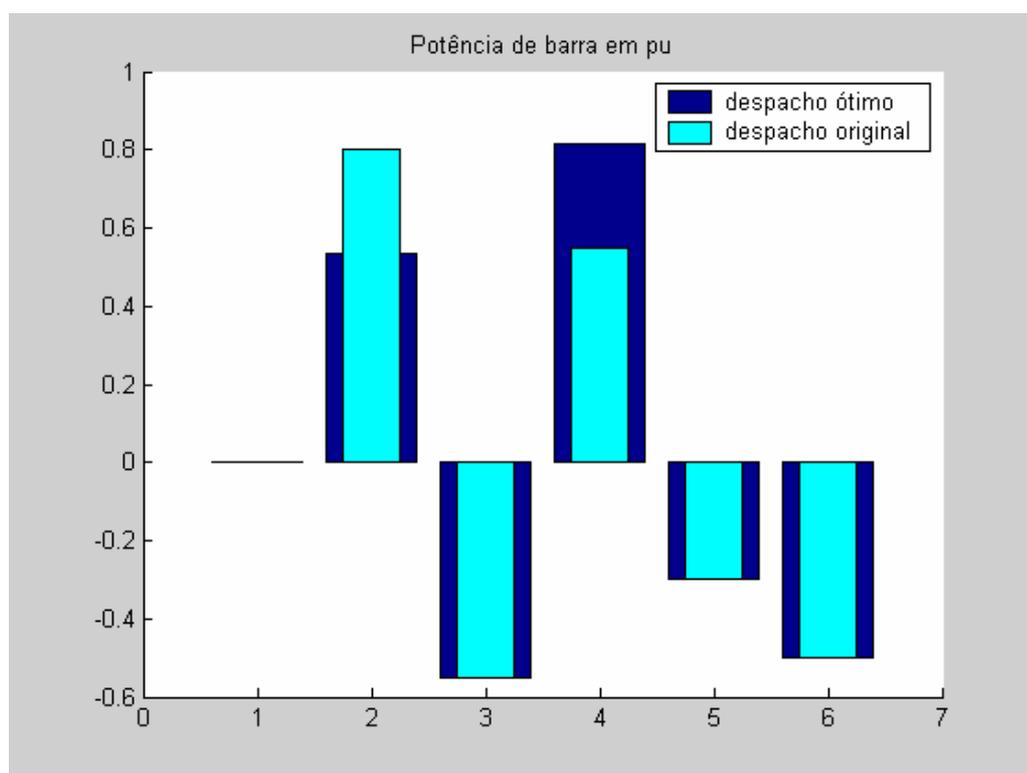


Figura 14 – Potência de Barra

Os fluxos de potência antes e após a otimização são mostrados na Figura 15.

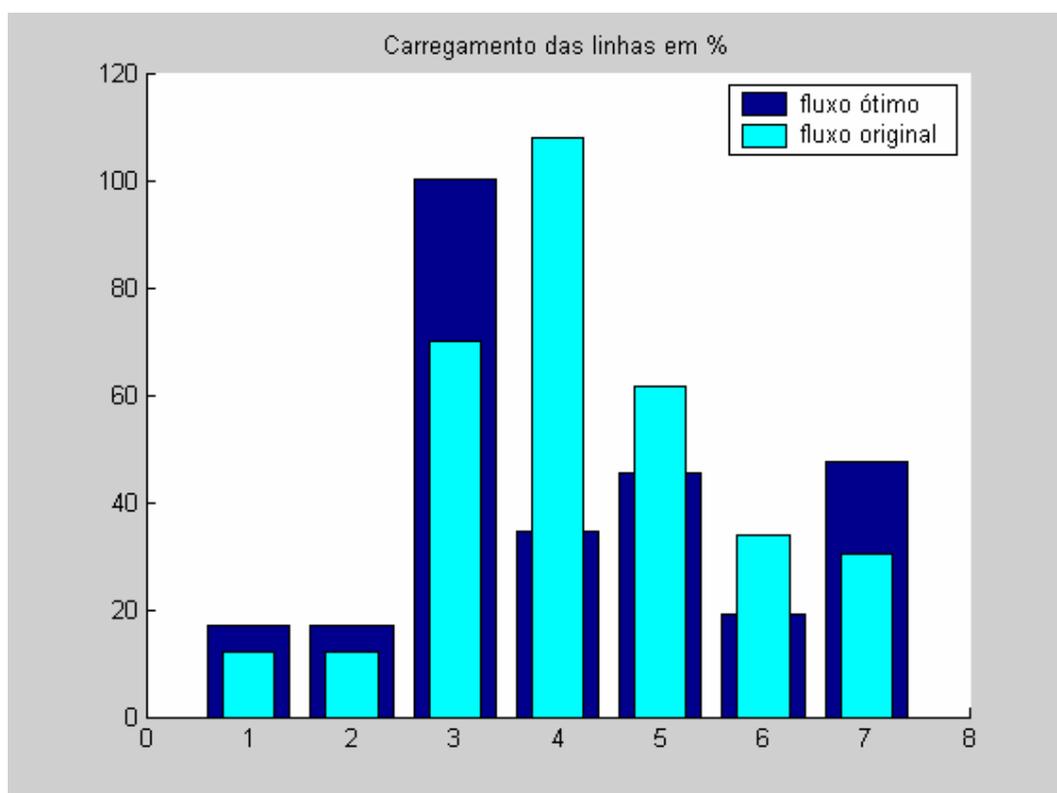


Figura 15 – Fluxos nas linhas

5.5 Mínima Injeção de Potência Ativa – Caso IV

Para esse caso, igualmente serão usados os dados do caso II. A função objetivo para este caso está descrita em 4.19, cuja finalidade é minimizar a injeção de potência ativa ao sistema. De maneira semelhante, será usada a rotina LINPROG do MatLab.

O problema pode, então ser formulado como segue:

$$\begin{aligned}
 & \min \sum_{i \in I_G} P_{Gi} \\
 & s.a. \\
 & \sum_{i \in I_G} P_{Gi} = \sum_{i \in I_D} P_{Di} \\
 & P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}
 \end{aligned} \tag{5.49}$$

Reescrevendo o problema formulado, tem-se:

$$\min C_T = P_{G2} + P_{G4}$$

s.a.

$$P_{G2} + P_{G4} = 1,35$$

$$\begin{bmatrix} -0,80 \\ -0,80 \\ -0,30 \\ -0,18 \\ -0,80 \\ -0,90 \\ -0,80 \end{bmatrix} \leq \begin{bmatrix} -0,4557 & -0,3183 & -0,3009 & -0,5394 & -0,5787 \\ -0,5443 & -0,6817 & -0,6991 & -0,4606 & -0,4213 \\ -0,0852 & 0,2146 & 0,2525 & -0,2679 & -0,3535 \\ 0,5409 & 0,1038 & 0,0484 & 0,8073 & -0,0678 \\ 0,5409 & 0,1038 & 0,0484 & -0,1927 & -0,0678 \\ 0,4591 & -0,1038 & -0,0484 & 0,1927 & 0,0678 \\ 0,4591 & 0,8962 & -0,0484 & 0,1927 & 0,0678 \end{bmatrix} \begin{bmatrix} P_{G2} \\ P_{D3} \\ P_{G4} \\ P_{D5} \\ P_{D6} \end{bmatrix} \leq \begin{bmatrix} 0,80 \\ 0,80 \\ 0,30 \\ 0,18 \\ 0,80 \\ 0,90 \\ 0,80 \end{bmatrix}$$

$$\begin{bmatrix} 0,0 \\ 0,0 \\ 0,0 \\ 0,0 \end{bmatrix} \leq \begin{bmatrix} P_{G2}^A \\ P_{G2}^B \\ P_{G4}^A \\ P_{G4}^B \end{bmatrix} \leq \begin{bmatrix} 0,5 \\ 0,5 \\ 0,5 \\ 0,5 \end{bmatrix}$$

A Figura 16 mostra o despacho antes e após a otimização.

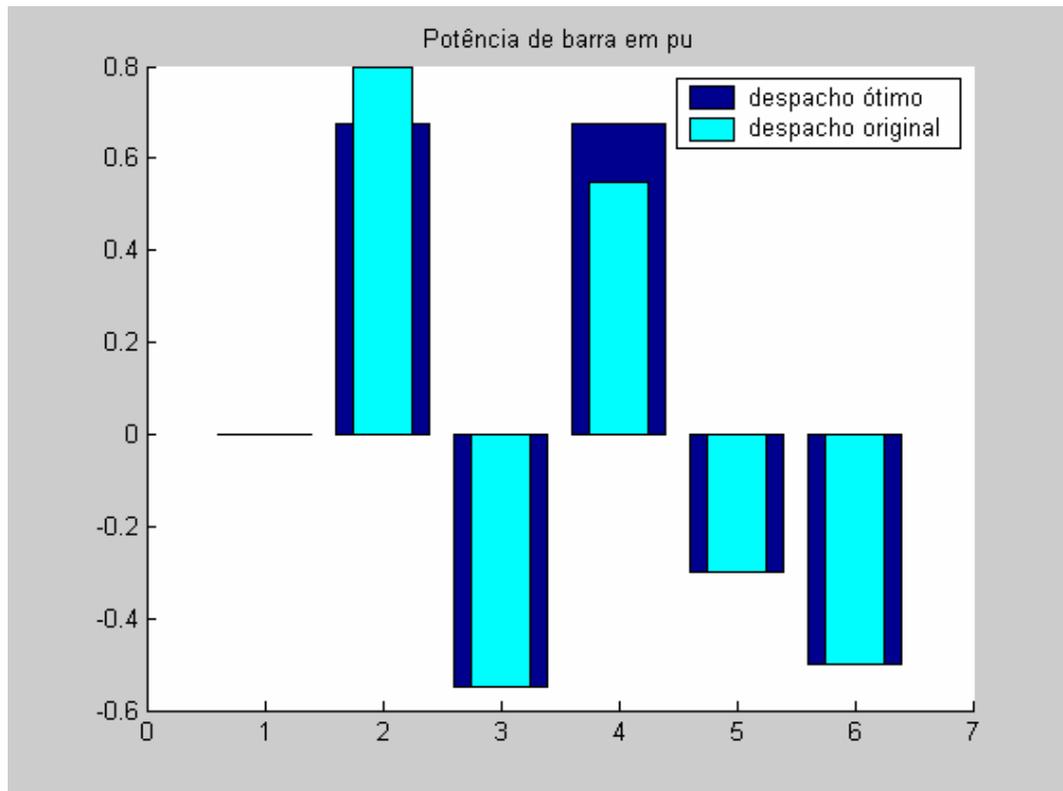


Figura 16 – Potência de Barra

A Figura 17 exibe os fluxos de potência antes e após a otimização.

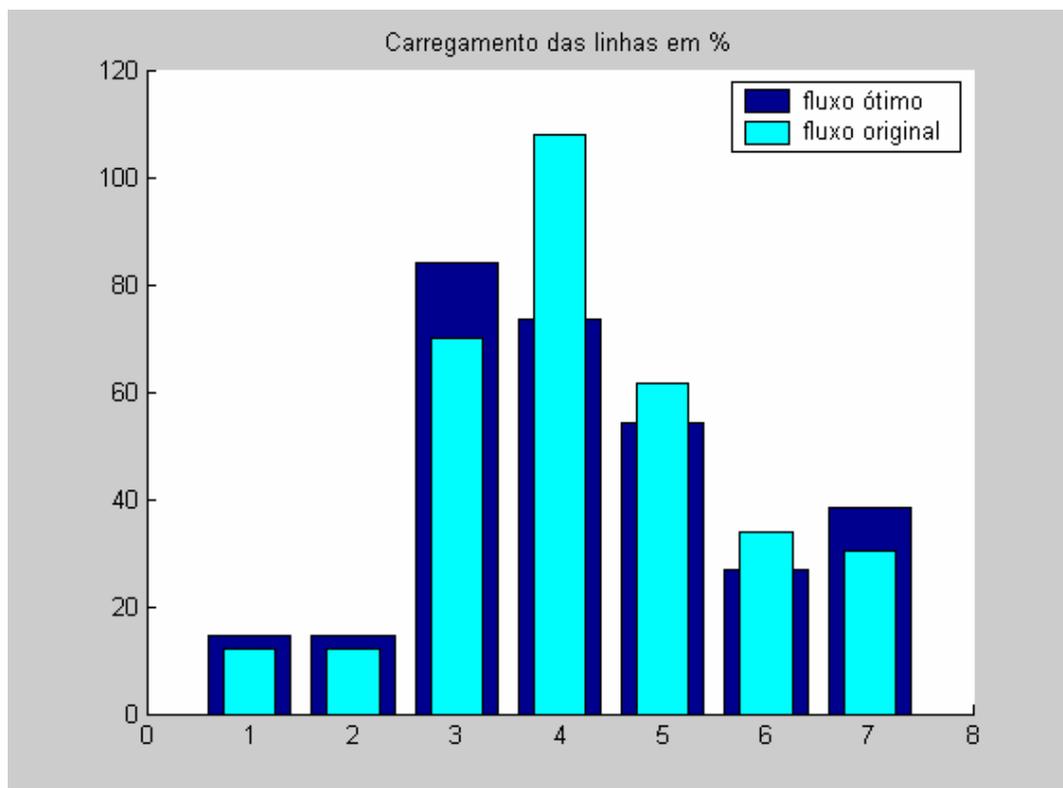


Figura 17 – Fluxos nas linhas

5.6 Gerenciamento de Congestionamento – Caso V

Esse caso envolve gerenciamento de congestionamento com o mínimo corte de carga. A função objetivo está descrita na equação 4.21 do capítulo 4, página 79. Os dados de rede são os mesmos do Caso II. Os limites das variáveis de controle são descritos na Tabela 9. Os limites de fluxos de potencia foram reduzidos em 50% com a finalidade criar situações de congestionamentos múltiplos. Para a resolução do problema será mantida a rotina LINPROG do *MatLab toolbox optimization*.

Tabela 9
Limites nas Variáveis de Controle

Variavel de Controle	Limite Inferior (pu)	Limite Superir (pu)
P_{G2}	0,40	0,80
P_{D3}	-0,55	-0,35
P_{G4}	0,10	0,55
P_{D5}	-0,30	-0,25
P_{D6}	-0,50	-0,40

O problema pode ser formulado na forma incremental:

$$\min f = [0,001 \ 0,5 \ 0,001 \ 0,5 \ 0,5] \cdot [\Delta P_{G2} \ -\Delta P_{D3} \ \Delta P_{G4} \ -\Delta P_{D5} \ -\Delta P_{D6}]^T \quad (5.50)$$

s.a.

$$[1 \ -1 \ 1 \ -1 \ -1] \cdot [0,8 + \Delta P_{G2} \ (0,55 + \Delta P_{D3}) \ 0,55 + \Delta P_{G4} \ (0,3 + \Delta P_{D5}) \ (0,5 + \Delta P_{D6})]^T = 0$$

$$\begin{bmatrix} -0,80 \\ -0,80 \\ -0,30 \\ -0,18 \\ -0,80 \\ -0,90 \\ -0,80 \end{bmatrix} \leq \begin{bmatrix} -0,4557 & -0,3183 & -0,3009 & -0,5394 & -0,5787 \\ -0,5443 & -0,6817 & -0,6991 & -0,4606 & -0,4213 \\ -0,0852 & 0,2146 & 0,2525 & -0,2679 & -0,3535 \\ 0,5409 & 0,1038 & 0,0484 & 0,8073 & -0,0678 \\ 0,5409 & 0,1038 & 0,0484 & -0,1927 & -0,0678 \\ 0,4591 & -0,1038 & -0,0484 & 0,1927 & 0,0678 \\ 0,4591 & 0,8962 & -0,0484 & 0,1927 & 0,0678 \end{bmatrix} \begin{bmatrix} P_{G_2} \\ P_{D_3} \\ P_{G_4} \\ P_{D_5} \\ P_{D_6} \end{bmatrix} \leq \begin{bmatrix} 0,80 \\ 0,80 \\ 0,30 \\ 0,18 \\ 0,80 \\ 0,90 \\ 0,80 \end{bmatrix}$$

$$\begin{bmatrix} 0,40 \\ -0,55 \\ 0,10 \\ -0,30 \\ -0,50 \end{bmatrix} \leq \begin{bmatrix} 0,80 + \Delta P_{G_2} \\ -(0,55 + \Delta P_{D_3}) \\ 0,55 + \Delta P_{G_4} \\ -(0,30 + \Delta P_{D_5}) \\ -(0,50 + \Delta P_{D_6}) \end{bmatrix} \leq \begin{bmatrix} 0,80 \\ -0,35 \\ 0,55 \\ -0,25 \\ -0,40 \end{bmatrix}$$

A Figura 18 mostra as variáveis de controle antes e depois da resolução do problema de corte de carga.

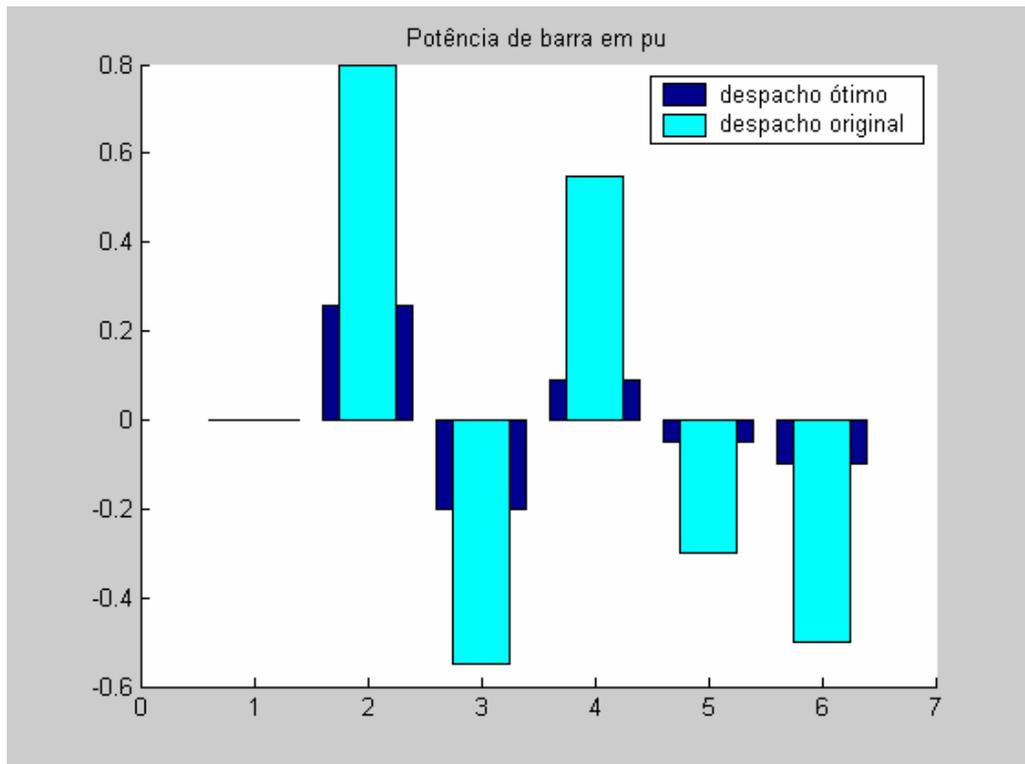


Figura 18 – Potência de barra

Os fluxos iniciais e finais estão representados na Figura 19.

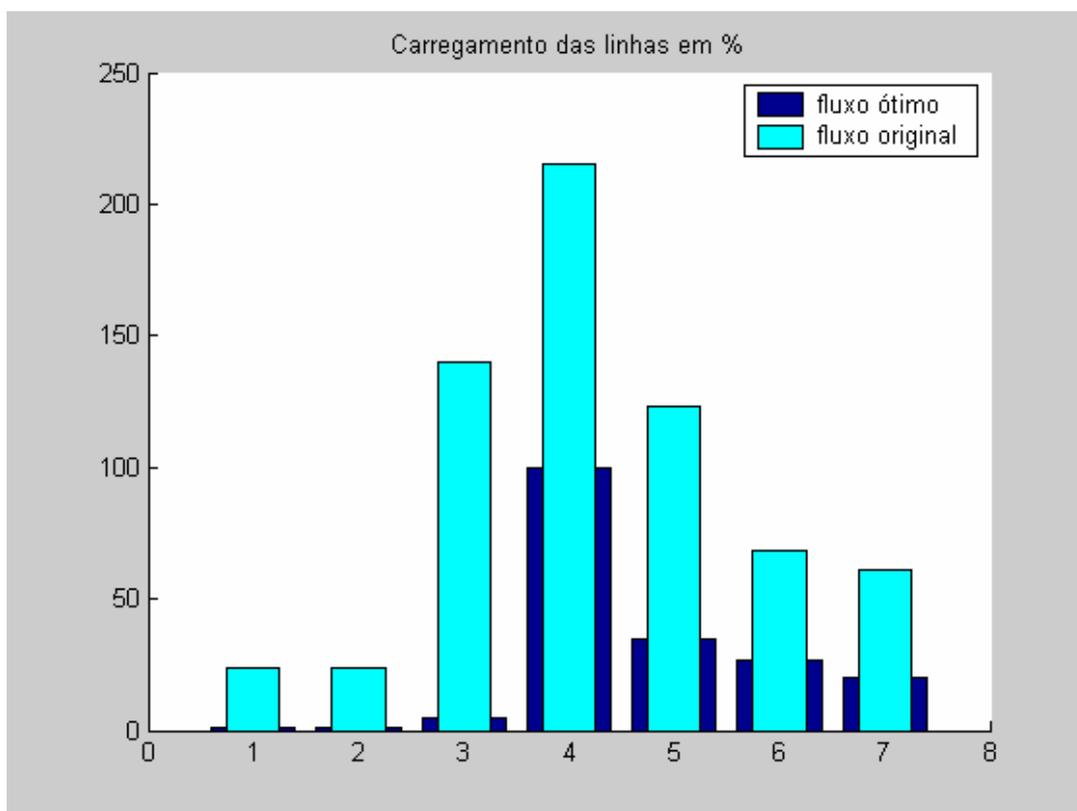


Figura 19 – Fluxos nas linhas

5.7 Gerenciamento de Congestionamento – Caso VI

Os dados para esse caso são fornecidos a partir dos elementos de uma rede padrão do sistema IEEE. Trata-se do sistema IEEE14bus, uma rede com 14 barras sendo 4 barras de geração, 9 barras de carga e 1 barra tomada como referência e 20 linhas interligando as barras. O objetivo do caso é minimizar, como em casos anteriores, o custo de geração de potência ativa. A função objetivo está representada em 4.17. A Figura 20, a seguir, ilustra o sistema do IEEE. Na Tabela 10 são apresentados os dados da rede.

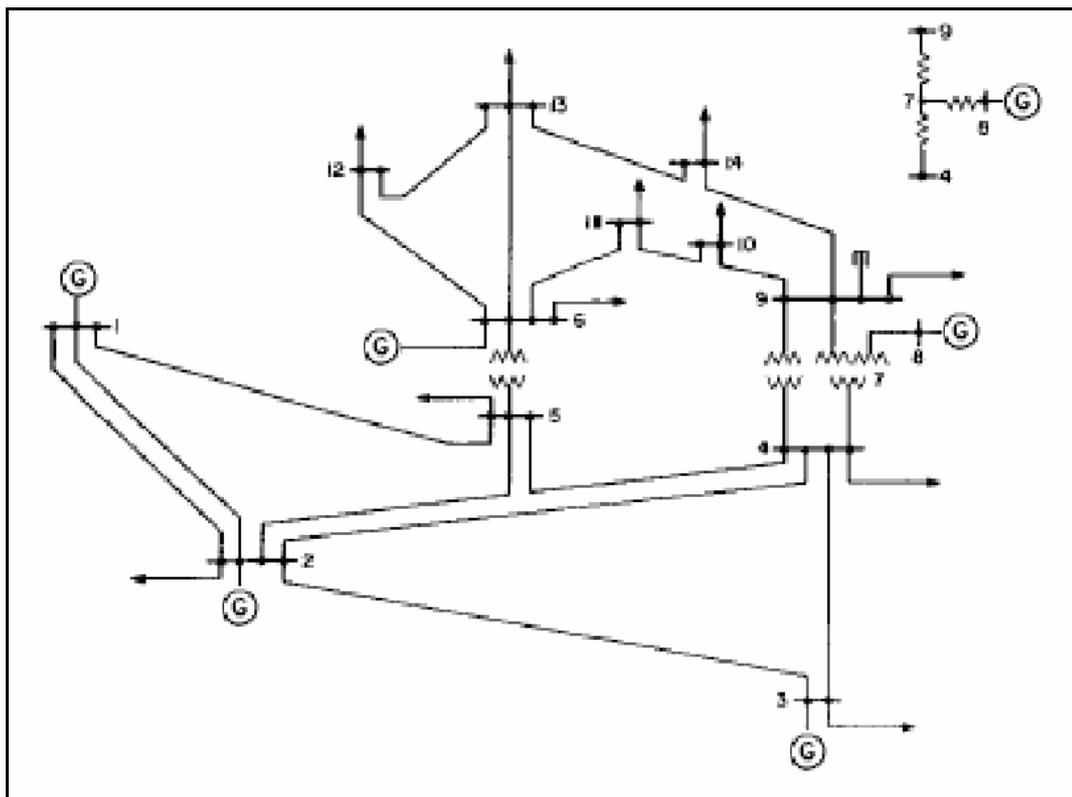


Figura 20 – Sistema IEEE14bus

Tabela 10
Dados da rede IEEE14bus

Legenda das Linhas	Barra Inicial	Barra Final	X (pu)	Limites de Fluxos de Potencia (pu)	Fluxos Ótimos (%)
L1	1	2	0,05920	1,30	6,6320
L2	1	5	0,22300	1,30	6,6320
L3	2	3	0,19800	1,30	5,0391
L4	2	4	0,17630	1,30	14,8743
L5	2	5	0,17390	1,30	10,7622
L6	3	4	0,17100	1,30	9,5006
L7	4	5	0,04210	1,30	17,8336
L8	4	7	0,20910	0,65	2,6593
L9	4	9	0,55620	0,65	8,2194
L10	5	6	0,25200	0,80	10,2139
L11	6	11	0,19890	0,32	50,8368
L12	6	12	0,25580	0,32	28,1553

L13	6	13	0,13030	0,32	69,2231
L14	7	8	0,17610	0,22	100,0000
L15	7	9	0,11000	0,65	36,5054
L16	9	10	0,08450	0,32	11,7743
L17	9	14	0,27040	0,32	10,4342
L18	10	11	0,19210	0,16	79,7986
L19	12	13	0,19990	0,16	18,1856
L20	13	14	0,34800	0,16	72,2567

Na Tabela 11 são apresentados os dados da barra. Considerando que as barras 2, 3, 6 e 8 são barras do tipo PV (geração), a barra 1 é a barra de referência (tipo 3) e as outras são consideradas barra de carga (PQ), os outros dados são aí apresentados.

Tabela 11
Dados de Barra do sistema IEEE14bus

Barra	Barra Tipo		(+) P_G (pu)	$P_{G \min}$ (pu)	$P_{G \max}$ (pu)	Custo unitário (\$ / pu)
			(-) P_G (pu)			
1	3	<i>Ref.</i>	2,556	0	0	0
2	1	PV	-0,183	0	1,0	0,03
3	1	PV	-0,942	0	1,0	0,02
4	2	PQ	-0,478	-	-	-
5	2	PQ	-0,076	-	-	-
6	1	PV	-0,112	0	0,8	0,03
7	2	PQ	-0,000	-	-	-
8	1	PV	-0,000	0	0,8	0,02
9	2	PQ	-0,295	-	-	-
10	2	PQ	-0,090	-	-	-
11	2	PQ	-0,035	-	-	-
12	2	PQ	-0,061	-	-	-
13	2	PQ	-0,135	-	-	-
14	2	PQ	-0,149	-	-	-

O problema pode então ser formulado:

$$\begin{aligned} \min C &= 0,03P_{G2} + 0,02P_{G3} + 0,03P_{G6} + 0,02P_{G8} \\ \text{s.a} & \\ P_{G2} + P_{G3} + P_{G6} + P_{G8} &= 2,556 \end{aligned} \tag{5.53}$$

$$\begin{bmatrix} -1,30 \\ -1,30 \\ -1,30 \\ -1,30 \\ -1,30 \\ -1,30 \\ -1,30 \\ -0,65 \\ -0,65 \\ -0,80 \\ -0,32 \\ -0,32 \\ -0,32 \\ -0,22 \\ -0,65 \\ -0,32 \\ -0,32 \\ -0,16 \\ -0,16 \\ -0,16 \end{bmatrix} \leq [C] \begin{bmatrix} P_{G2} - 0,183 \\ P_{G3} - 0,942 \\ P_{D4} - 0,478 \\ P_{D5} - 0,076 \\ P_{G6} - 0,112 \\ P_{D7} - 0,000 \\ P_{G8} - 0,000 \\ P_{D9} - 0,295 \\ P_{D10} - 0,090 \\ P_{D11} - 0,035 \\ P_{D12} - 0,061 \\ P_{D13} - 0,135 \\ P_{D4} - 0,149 \end{bmatrix} \leq \begin{bmatrix} 1,30 \\ 1,30 \\ 1,30 \\ 1,30 \\ 1,30 \\ 1,30 \\ 1,30 \\ 0,65 \\ 0,65 \\ 0,80 \\ 0,32 \\ 0,32 \\ 0,32 \\ 0,22 \\ 0,65 \\ 0,32 \\ 0,32 \\ 0,16 \\ 0,16 \\ 0,16 \end{bmatrix}$$

A matriz de conexão $[C]$ é dada no apêndice B. A Figura 21 mostra as variáveis de controle, antes e depois da resolução do problema de minimização do custo de geração de potência ativa para um sistema de 14 barras.

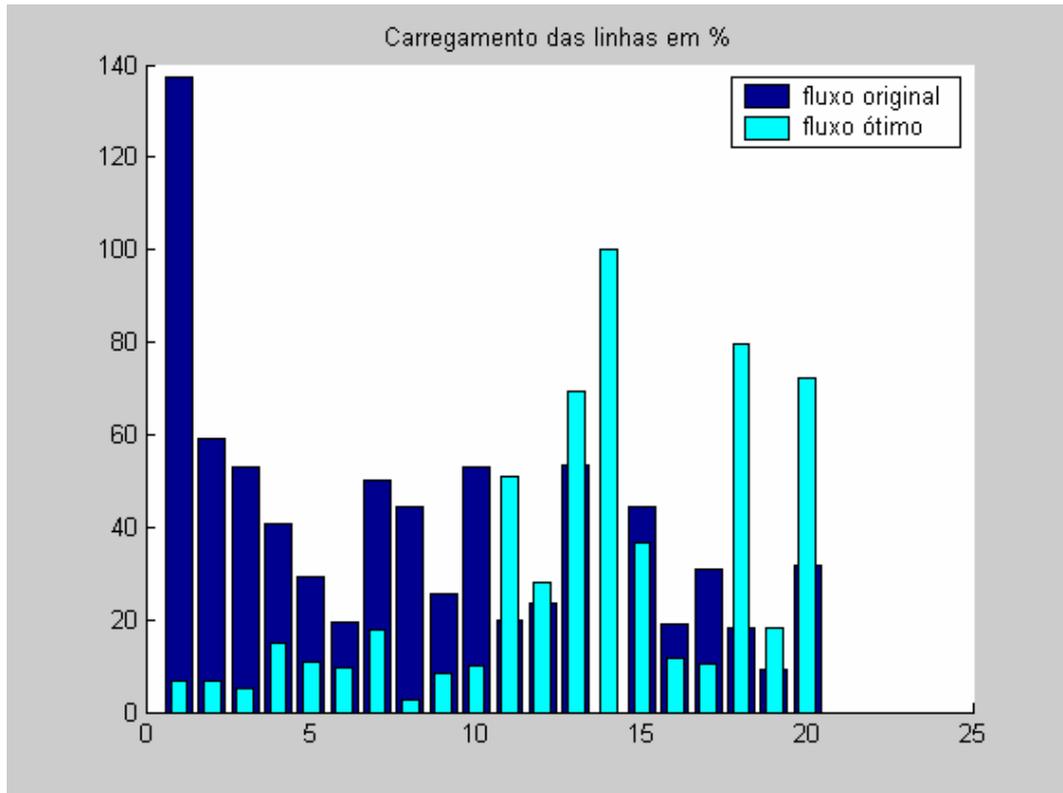


Figura 21 – Fluxos nas Linhas

A Figura 22 exibe as potências nas barras antes e depois da otimização.

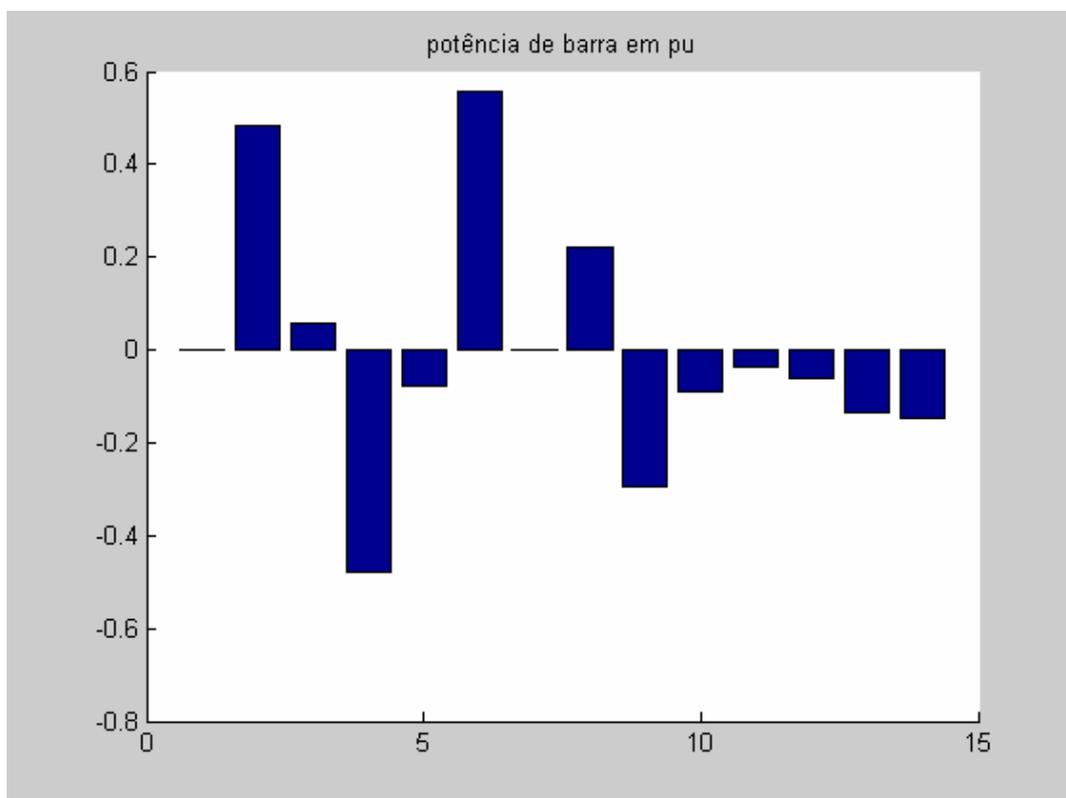


Figura 22 – Potência de barra

5.8 Sumário

Neste capítulo mostrou-se a implementação de diferentes formulações da Programação Linear aplicada ao fluxo de potência ótimo linearizado. Nos casos IA, IB e IC usou-se duas versões de algoritmo Simplex e uma versão dos Pontos Interiores Primal-Dual. Esse último demonstrou-se ineficiente para a solução do problema. Os casos II, III, IV, V e VI envolveram formulações aplicáveis à resolução de diferentes problemas de otimização em Engenharia Elétrica.

Capítulo 6 Conclusão

A tendência mundial no setor energético é torná-lo competitivo e atraente, com a finalidade de levar ao consumidor energia de boa qualidade, retirando do Estado a responsabilidade de investimento. Além desses fatores, há também o fator ambiental, que leva o homem a buscar novas fontes de energia elétrica. Apesar disso, existe um crescimento contínuo da demanda de energia elétrica, o que torna necessário ampliar os sistemas elétricos e otimizar o seu uso.

A construção de novas linhas de transmissão é limitada pelos impactos ambientais provocados e também pelos direitos de passagem. Uma boa alternativa é aproveitar a capacidade máxima das linhas, otimizando seu fluxo, minimizando as perdas e os custos de geração de potência.

Este trabalho investigou aplicações de programação matemática, mais precisamente, da programação linear ao problema do fluxo de potência ótimo linearizado.

Foram apresentados e discutidos diferentes métodos de resolução de problemas em programação linear. O método dos pontos interiores foi reportado como indicado para problemas de grande porte. Para redes pequenas ou médias, o algoritmo Simplex mostrou-se adequado.

O tema complexidade de algoritmos foi examinado e reportado.

O problema do fluxo de potência ótimo, em suas diversas formulações, foi examinado e as funções objetivas e restrições de problemas típicos descritas.

Formulações dos problemas do Despacho Ótimo, do Corte de Carga e do Congestionamento de Rede, em sua versão linear, foram apresentados e exemplos numéricos demonstrativos apresentados.

Direções para Trabalhos Futuros:

- Avaliar as formulações das funções objetivo e restrições apresentadas nesta dissertação, endereçando a parte não-linear do problema do fluxo de potência ótimo;
- Estudar a estrutura das matrizes esparsas dos algoritmos para resolução de grandes redes;
- Investigar formulações matemáticas adequadas à resolução de problemas envolvendo essas grandes redes;
- Investigar a adequação de métodos eficientes para resolução de grandes redes.

Referências Bibliográficas

ARAUJO, Leandro Ramos de. **Uma Contribuição ao Fluxo de Potência Ótimo Aplicado a Sistemas de Potência Trifásicos usando o Método dos Pontos Interiores**. Tese (Doutorado). COPPE – UFRJ, Rio de Janeiro, 2005.

AZEVEDO, A.T.de, at. al. **Métodos de Pontos Interiores Aplicados a Problemas de Multifluxo com Restrições Adicionais**. Tendências em Matemática Aplicada e Computacional, 3, No. 1 p. 41-50, 2002a.

AZEVEDO, A.T. de. **Aplicação dos métodos de pontos interiores em problemas de manufaturas e energia elétrica**. Dissertação (Mestrado). Unicamp. São Paulo, 2002b.

BAPTISTA, E. C., at al. **Um método primal-dual aplicado na resolução do problema de fluxo de potência ótimo**. Pesquisa Operacional, v.24, n.2, p.215-226, 2004.

BARBOSA, Marco Antonio de Castro. **Ferramenta para Automatização da Análise da Complexidade de Algoritmos**. Semana Acadêmica, 2000. <http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana2000/MarcoBarbosa/>. Acesso em 15 ago 2006.

BAZARAA, M.S.; Jarvis, J.J. & Serali, H.D. **Linear Programming and Network Flows**. 2nd Ed., Wiley, New York. 1992.

BORGES, C.L.T., at al. **Melhoria do Desempenho de Algoritmos de Análise de Sistemas de Potência em Processamento Vetorial**. Departamento de Eletrotécnica – UFRJ, Rio de Janeiro, 1996.

BUENO, Elivelton Ferreira. MENEZES, Marco Antonio Figueiredo. **Implementação de Algoritmos das Famílias Simplex, Elipsóides e Pontos Interiores para Programação Linear**. Departamento de Computação da Universidade Católica de Goiás. 2002.

CORMEN, Thomas H. et al. **Algoritmos: Teoria e prática**. Tradução da Segunda edição Americana. Rio de Janeiro. Editora Campus, 2002.

ELGERD, O. I: **Introdução à Teoria de Sistemas de Energia Elétrica**, Ed. McGraw-Hill. São Paulo, 1978.

FALCAO, Djalma M.: **Análise de Redes Elétricas**. COPPE – UFRJ, Rio de Janeiro, 2003.

GOLDBARG, M.C. LUNA, Henrique Pacca L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: Campus, 2000.

GOLDBARG, M.C. LUNA, Henrique Pacca L. **Otimização combinatória e programação linear: modelos e algoritmos**. 2^a. Edição. Rio de Janeiro: Elsevier, 2005.

GONDZIO, J.: **Re-Optimization with the primal-dual interior point method**. Julio, 2002.

GONZAGA, C.C. **Algoritmos de Pontos Interiores Para a Programação Linear**, Rio de Janeiro, IMPA, séries Colóquio Brasileiro de Matemática, n. 17, 1989,.

GREVE, Fabiola Gonçalves Pereira. **Notas de Aula**. Universidade Federal da Bahia. 2004. <http://twiki.im.ufba.br/bin/view/MAT053/MaterialDidatico>. Acesso em 18 jul. 2004.

IEZZI, Gelson. HAZZAN Samuel. **Fundamentos de Matemática Elementar 4**. 6ª. Edição. São Paulo: Ed. Atual, 1993.

LEITÃO, Hélio. <http://www.dei.isep.ipp.pt/~hleitao/EI/AnaliseAlgoritmos.pdf>. Acesso em maio 2006.

LIMA, A.M. de. **Comparação entre diferentes abordagens do problema do fluxo de potência ótimo utilizando o método de pontos interiores**. Dissertação (Mestrado). UCMC-USP, São Paulo, 2004.

LUENBERGER, D.G.. **Linear and Nonlinear Programming**. Second Edition, 1937.

MONTICELLI, Alcir. **Fluxo de carga em redes de energia elétrica**. São Paulo: Edgard Blucher, c1983. 164p.

OLIVEIRA, A.R.L., FILHO, S.S.. **Métodos de pontos interiores para problemas de fluxo de potência ótimo DC**. Revista Controle & Automação. Vol. 14 nº 3.2003.

PUCCINI, A.L. **Introdução à pesquisa operacional**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora AS, 1984.

QUINTANA, V.H. **Interior-Point Methods and Their Applications to Power Systems: A Classification of Publications and Software Codes**. IEEE Transactions on Power Systems. Vol. 1 N. 1 Fevereiro, 2000.

RIBEIRO, Pablo Motta: **Remuneração dos Serviços Ancilares de Suporte de Potência Reativa e Reserva de Potência Quando Providos por Geradores**. Dissertação (Mestrado) PPGEE – PUC Rio, 2005.

RODRIGUES, Vanessa Cristina. **Controle de Rotas de Fluxo de Potência**. Dissertação (Mestrado). PPGEE – PUC Minas, 2002.

SILVA, E. M. da, et al. **Pesquisa Operacional para os cursos de: Economia, Administração e Ciências Contábeis**. 3 ed. São Paulo: Atlas, 1998.

SILVA, I.N. da, et al. **Resolvendo problemas de fluxo de potencia ótimo através de uma rede de Hopfield modificada**. Revista Controle & Automação/Vol.15 no.4 Outubro, Novembro e Dezembro, 2004.

SOUSA, Ricardo Silveira, SILVA, Carla Taviane Lucke da and ARENALES, Marcos Nereu. **Métodos do tipo dual simplex para problemas de otimização linear canalizados**. Pesquisa Operacional., setembro/Dezembro. 2005, vol.25, no.3, p.349-382.

SOUSA, R.S. **Estudos em Otimização Linear**. Dissertação (Mestrado), ICMC-USP. São Paulo, 2000.

TERRA, L.D.B. **A global Methodology for Reactive Power Management and Voltage Control in Power System**. Tese (Doutorado). Imperial College – London 1989.

THOMAZ, A.O., A.R.L. **Método de Pontos Interiores Primal-Dual Aplicado ao Problema de Fluxo de Potência Ótimo Utilizando Coordenadas Cartesianas**. USP, São Paulo, 2000.

TOSCANI, Laira V. & VELOSSO, Paulo S. **Complexidade de Algoritmos: análise, projetos e métodos**. Porto Alegre: Instituto de informática da UFRGS: Sagra-Luzzatto,

2001. ISBN 8524106492. <http://www.inf.ufrgs.br/~laira/>. Acesso em set. 2006.

WEBER, J. D. **Implementation of a Newton-Based Optimal Power Flow into a Power System Simulation Environment**. Dissertação (Mestrado), B.S., University of Wisconsin – Platteville, 1995.

WRIGHT, M.H. **The interior-point revolution in optimization: History, recent developments, and lasting consequences**. Bulletin of the American Mathematical Society. Vol. 42, nº 1, p. 39-56. 2004.

Zhang, Y. **Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment**. Technical Report TR96-01, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD, July 1995.

ZIVIANI, Nívio. **Projeto de Algoritmos com Implementação em Pascal e C++**. 5ª edição. São Paulo: Pioneira, 2000. (www.dcc.ufmg.br/algoritmos/)

Apêndice

Matriz de Conexão para o Caso VI

	-0.8379	-0.7464	-0.6674	-0.6105	-0.6298	-0.6572	-0.6572	-0.6518	-0.6479	-0.6390	-0.6316	-0.6329	-0.6436
	-0.1621	-0.2536	-0.3326	-0.3895	-0.3702	-0.3428	-0.3428	-0.3482	-0.3521	-0.3610	-0.3684	-0.3671	-0.3564
	0.0274	-0.5319	-0.1513	-0.1031	-0.1195	-0.1427	-0.1427	-0.1381	-0.1348	-0.1273	-0.1210	-0.1221	-0.1311
	0.0573	-0.1434	-0.3168	-0.2157	-0.2501	-0.2986	-0.2986	-0.2891	-0.2822	-0.2664	-0.2532	-0.2556	-0.2744
	0.0774	-0.0711	-0.1993	-0.2917	-0.2603	-0.2159	-0.2159	-0.2246	-0.2309	-0.2453	-0.2575	-0.2553	-0.2380
	0.0274	0.4681	-0.1513	-0.1031	-0.1195	-0.1427	-0.1427	-0.1381	-0.1348	-0.1273	-0.1210	-0.1221	-0.1311
	0.0801	0.3072	0.5033	-0.3016	-0.0279	0.3590	0.3590	0.2831	0.2278	0.1022	-0.0033	0.0159	0.1662
	0.0029	0.0111	0.0182	-0.0109	-0.2171	-0.6342	-0.6342	-0.4514	-0.4097	-0.3151	-0.2356	-0.2501	-0.3633
	0.0017	0.0064	0.0104	-0.0062	-0.1245	-0.1661	-0.1661	-0.2590	-0.2351	-0.1808	-0.1352	-0.1435	-0.2085
	-0.0045	-0.0174	-0.0286	0.0171	-0.6584	-0.1997	-0.1997	-0.2897	-0.3552	-0.5042	-0.6292	-0.6065	-0.4282
C =	-0.0027	-0.0105	-0.0172	0.0103	0.2057	-0.1202	-0.1202	-0.1744	-0.2846	-0.5350	0.1757	0.1522	-0.0316
	-0.0004	-0.0015	-0.0025	0.0015	0.0302	-0.0177	-0.0177	-0.0256	-0.0157	0.0069	-0.5201	-0.1687	-0.0882
	-0.0014	-0.0054	-0.0088	0.0053	0.1057	-0.0618	-0.0618	-0.0896	-0.0549	0.0240	-0.2848	-0.5900	-0.3084
	0	0	0	0	0	0.0000	-1.0000	0	0	0	0	0	0
	0.0029	0.0111	0.0182	-0.0109	-0.2171	0.3658	0.3658	-0.4514	-0.4097	-0.3151	-0.2356	-0.2501	-0.3633
	0.0027	0.0105	0.0172	-0.0103	-0.2057	0.1202	0.1202	0.1744	-0.7154	-0.4650	-0.1757	-0.1522	0.0316
	0.0018	0.0069	0.0114	-0.0068	-0.1359	0.0794	0.0794	0.1152	0.0706	-0.0309	-0.1951	-0.2413	-0.6034
	0.0027	0.0105	0.0172	-0.0103	-0.2057	0.1202	0.1202	0.1744	0.2846	-0.4650	-0.1757	-0.1522	0.0316
	-0.0004	-0.0015	-0.0025	0.0015	0.0302	-0.0177	-0.0177	-0.0256	-0.0157	0.0069	0.4799	-0.1687	-0.0882
	-0.0018	-0.0069	-0.0114	0.0068	0.1359	-0.0794	-0.0794	-0.1152	-0.0706	0.0309	0.1951	0.2413	-0.3966