

Trabalho Prático 2

Ítalo Dell'Areti

Universidade Federal de Minas Gerais (UFMG)

Belo Horizonte – MG – Brasil

1. Introdução

O problema proposto consiste em desenvolver um sistema de diagnóstico para a rede elétrica da Metalmax, uma grande siderúrgica especializada na produção de aço. A rede é composta por geradores, consumidores e conexões entre eles, onde os geradores produzem energia, enquanto os consumidores possuem demandas específicas. O sistema deve resolver quatro problemas principais:

1. Calcular a capacidade máxima que a rede pode operar.
2. Determinar quanto de energia não está sendo atendida devido às limitações da rede.
3. Calcular quanto de energia é perdida durante a transmissão.
4. Identificar as conexões críticas que operam em capacidade máxima

Ao final, devemos computar estes diagnósticos e retornar os resultados para a empresa. Portanto, com base nas características e propriedades grafos e seus componentes, algoritmos de fluxo máximo e suas propriedades, implementamos uma solução que respeita todas as nuances do problema. Tendo em vista disso, este documento possui como objetivo explicitar o funcionamento desta implementação.

2. Modelagem

O problema foi modelado utilizando teoria dos grafos com fluxo, onde:

- Vértices (V): Representam os geradores e consumidores.
- Arestas direcionadas (E): Representam as conexões com suas capacidades
- Grafo $G(V,E)$: Representa a rede elétrica completa
- Matriz de Adjacência: Estrutura escolhida para representar o grafo, permitindo fácil manipulação de fluxos

Principais algoritmos utilizados:

- Busca em Largura (BFS): Para encontrar caminhos aumentantes
- Ford-Fulkerson: Para cálculo de fluxo máximo

3. Solução.

3.1 Cálculo da Energia Total

A solução envolve adaptar o problema para um cenário de fluxo máximo:

1. Adaptação do grafo:
 - Adiciona-se um nó `super_fonte` conectado a todos os geradores com capacidade infinita
 - Adiciona-se um nó `super_sumidouro` conectado a todos os consumidores com capacidade igual à demanda
2. Processo detalhado:
 - Backup da matriz de capacidades original
 - Redimensionamento da rede para incluir super nós
 - Configuração das conexões com a `super_fonte` e `super_sumidouro`
 - Execução do Ford-Fulkerson
 - O fluxo máximo encontrado representa a energia total possível
3. Importância:
 - Garante o máximo de energia que pode fluir dos geradores aos consumidores
 - Respeita todas as restrições de capacidade das conexões
 - Considera múltiplos caminhos possíveis

3.2 Energia Não-atendida

O cálculo identifica quanto da demanda dos consumidores não pode ser atendida:

1. Cálculo da demanda total:
 - Soma das demandas de todos os consumidores
 - Representa a energia total necessária
2. Processo de cálculo:
 - Configura rede com super nós
 - Calcula fluxo máximo possível
 - Subtrai fluxo máximo da demanda total
3. Significado:
 - Valor positivo indica deficiência no atendimento
 - Zero indica que todas as demandas são atendidas

3.3 Energia Perdida

Calcula a diferença entre energia disponível e efetivamente utilizada:

1. Energia total gerada:
 - Soma das capacidades de todas as conexões que saem dos geradores
 - Representa o potencial máximo de geração

2. Processo detalhado:
 - Calcula energia inicial disponível dos geradores
 - Configura rede com super nós
 - Calcula fluxo máximo efetivo
 - Subtrai fluxo máximo da energia inicial
3. Significado:
 - Identifica perdas na transmissão
 - Ajuda a encontrar ineficiências na rede
 - Indica potencial de otimização

3.4 Conexões Críticas

Identifica conexões que operam no limite:

1. Processo de identificação:
 - Executa Ford-Fulkerson para estabelecer fluxos
 - Compara fluxo com capacidade em cada conexão
 - Seleciona conexões onde fluxo = capacidade
2. Ordenação:
 - Ordena conexões por capacidade em ordem decrescente e em caso de empate, ordena pelo identificador dos vértices, em ordem crescente
 - Implementa ordenação sem usar bibliotecas externas
3. Importância:
 - Identifica pontos críticos da rede
 - Ajuda no planejamento de melhorias para a empresa

4. Análise de Complexidade:

4.1 Estruturas de Dados

- Matriz de capacidade: $O(|V|^2)$
- Matriz de fluxo: $O(|V|^2)$
- Vetor de demandas: $O(|V|)$
- Vetor de identificação de geradores: $O(|V|)$

4.2 Construtores e Funções de Configuração

RedeEletrica(int vertices)

- Tempo: $O(|V|^2)$ para resize das matrizes
- Espaço: $O(|V|^2)$

adicionar_conexao

- Tempo: $O(1)$
- Espaço: $O(1)$

definir_tipo_no

- Tempo: $O(1)$
- Espaço: $O(1)$

4.3 Funções de Fluxo

busca_em_largura

- Tempo: $O(|V| + |E|)$
- Espaço: $O(|V|)$ para vetor visitado e fila

ford_fulkerson

- Tempo: $O(|V|E^2)$
- Espaço: $O(|V|^2)$ para grafo residual

valor_minimo

- Tempo: $O(1)$
- Espaço: $O(1)$

4.4 Funções Principais

calcular_energia_total

- Tempo: $O(|V|^2)$ para cópia + $O(|V|E^2)$ para Ford-Fulkerson
- Espaço: $O(|V|^2)$
- Dominante: $O(|V|E^2)$

calcular_energia_nao_atendida

- Tempo: $O(|V|^2)$ para setup + $O(|V|)$ para soma + $O(|V|E^2)$ para Ford-Fulkerson
- Espaço: $O(|V|^2)$
- Dominante: $O(|V|E^2)$

calcular_energia_perdida

- Tempo: $O(|V|^2)$ para cópias e somas + $O(|V|E^2)$ para Ford-Fulkerson
- Espaço: $O(|V|^2)$
- Dominante: $O(|V|E^2)$

encontrar_conexoes_criticas

- Tempo: $O(|V|E^2)$ para Ford-Fulkerson + $O(|E|^2)$ para ordenação
- Espaço: $O(|V|^2) + O(|E|)$ para vetor de conexões críticas
- Dominante: $O(|V|E^2)$

4.5 Complexidade Total do Sistema

- Tempo: $O(|V|E^2)$ dominado pelo Ford-Fulkerson
- Espaço: $O(|V|^2)$ dominado pelas matrizes de capacidade e fluxo

5. Considerações Finais

O desenvolvimento deste trabalho proporcionou uma experiência prática valiosa na aplicação de algoritmos de fluxo máximo em redes, especialmente na resolução de problemas relacionados à distribuição de energia elétrica.

A parte mais direta de implementar foi o cálculo da energia total (2.1), por ser uma aplicação direta do algoritmo de Ford-Fulkerson combinado com o conceito de super fonte e super sumidouro. O cálculo da energia não-atendida (2.2) também foi relativamente simples, envolvendo a diferença entre a demanda total e o fluxo máximo que a rede consegue entregar.

A principal dificuldade encontrada foi na implementação do cálculo de energia perdida (2.3), que exigiu uma compreensão mais profunda sobre como calcular a diferença entre a energia gerada inicialmente e o fluxo máximo alcançado na rede. Esta parte demandou várias iterações até encontrar a implementação correta que satisfizesse todos os casos de teste.

A implementação das conexões críticas (2.4) também apresentou desafios interessantes, principalmente na parte de identificação das conexões e realizar uma ordenação mantendo uma complexidade aceitável.

Além disso, utilizei a vpl disponibilizada no moodle e passou em todos os casos de teste, o que demonstra que a lógica implementada no programa está correta. Por segurança, também verifiquei se existia algum vazamento de memória através da aplicação Valgrind e compilei o programa para demonstrar seu funcionamento.

Vídeo com a implementação e teste do Valgrind:

<https://drive.google.com/file/d/1PDd1vNfJdK9kL43P3KGgS9luYK2iPz0L/view?usp=sharing>

6. Referências

[*Geeks for geeks : Breadth First Search or BFS for a Graph*](#)

[*Geeks for Geeks: Ford Fulkerson Algorithm for Maximum Flow Problem*](#)

CORMEN, T. H. et al. Introduction to Algorithms. 3rd ed. MIT Press, 2009.

KLEINBERG, J.; TARDOS, E. Algorithm Design. Pearson, 2005.