Trabalho Prático 2
Algoritmos I

Vitor Augusto Reis Azevedo
2023002294

Universidade Federal de Minas Gerais
Belo Horizonte - MG - Brasil

vitoraugreis@dcc.ufmg.br

---

1. Introdução

A Metalmax é uma grande siderúrgica especializada na produção de aço, que depende de uma rede elétrica interna complexa para alimentar seus processos industriais, como fornos, laminadores e sistemas auxiliares. Recentemente, a empresa tem enfrentado dificuldades no fornecimento de energia, o que tem impactado sua produção e elevado os custos operacionais. A rede elétrica da Metalmax é composta por geradores, consumidores e conexões, cada uma com suas especificidades e limitações.

O objetivo é realizar uma análise detalhada dessa rede elétrica para diagnosticar possíveis falhas e gargalos, com foco em quatro aspectos principais:

1. **Capacidade Máxima da Rede**: Determinar qual a capacidade máxima que a rede pode suportar, considerando todos os geradores e as conexões, assegurando que todos os consumidores sejam atendidos.
2. **Energia Não-Atendida**: Calcular a quantidade de energia que está faltando para os consumidores, ou seja, o quanto da demanda total não está sendo atendida pela rede.
3. **Energia Perdida**: Identificar a quantidade de energia que se perde ao longo das conexões, considerando as limitações de capacidade dessas vias de transmissão.
4. **Conexão Crítica**: Identificar quais conexões estão operando em sua capacidade máxima, o que pode indicar pontos de falha potenciais na rede.

A análise desses pontos fornecerá insights essenciais para otimizar o funcionamento da rede elétrica e minimizar impactos na produção de aço da empresa.

2. Modelagem

O problema descrito foi modelado como um grafo direcionado ponderado. Cada vértice do grafo representa um gerador ou um consumidor, enquanto cada aresta representa uma conexão, que pode ser estabelecida entre um gerador e um consumidor, ou entre consumidores. As arestas possuem sentido único.

Este grafo é utilizado para representar o fluxo na rede. Além disso, a estrutura contém um grafo residual, que permite simular o envio de fluxo de maneira dinâmica, ajustando o fluxo ao longo das arestas conforme as capacidades são preenchidas ou esgotadas, facilitando a busca pelo fluxo máximo da rede, que é usado para determinar a energia máxima que pode ser transmitida.

Para determinar essa energia, utilizamos o algoritmo de Ford-Fulkerson, que retorna o fluxo máximo da rede (energia máxima disponível). A partir da execução desse algoritmo, também podemos identificar as conexões críticas, que terão, ao final de sua execução, capacidade igual a 0, no grafo residual, indicando que estão operando em sua potência máxima. Para definir a energia não atendida, calculamos a demanda total da rede e verificamos se o fluxo máximo é suficiente para suprir essa demanda. Por fim, a energia perdida na rede é calculada subtraindo a energia que chega efetivamente aos consumidores da energia gerada pelos geradores.

3. Solução

Inicialmente, como os identificadores dos consumidores ou geradores não possuem uma ordem fixa, foi atribuído a cada novo vértice um identificador no grafo para facilitar as operações na matriz de adjacência. Cada vértice é armazenado em um dicionário, onde a chave é o identificador especificado na entrada do programa e o valor é um par composto pelo identificador designado no grafo (usado para acessar a posição na matriz de adjacência) e pela demanda do vértice. Caso a demanda do vértice especificado na entrada seja igual a zero, o vértice é automaticamente considerado um gerador e armazenado em um vetor de geradores.

A matriz de adjacência do grafo possui tamanho $n \, x \, n$ (n = número de vértices), onde estão representadas todas as conexões especificadas na entrada. O grafo residual é uma cópia do grafo original, com a adição de dois novos vértices: o super gerador e o super sumidouro, ou seja, de tamanho $(n + 2) \, x \, (n + 2)$.

O super gerador é um vértice fictício conectado a todos os geradores reais da rede (super gerador → geradores). Em vez de considerar cada gerador individualmente, o super gerador centraliza o fornecimento de energia, estabelecendo conexões com os geradores reais com capacidade de fluxo infinita. Isso permite tratar todos os geradores como uma única fonte de fluxo, facilitando a análise do fluxo total da rede.

De forma semelhante, o super sumidouro é um vértice fictício conectado a todos os consumidores da rede (consumidores → super sumidouro). Ele coleta o fluxo proveniente dos consumidores e permite que o algoritmo trate todos eles como um único ponto de consumo, simplificando a análise da demanda e a determinação do fluxo necessário para atender a todos os consumidores simultaneamente.

A inclusão desses vértices fictícios no grafo tem como objetivo transformar o problema de fluxo envolvendo múltiplos geradores e múltiplos consumidores em um problema de fluxo com uma única fonte e um único sumidouro. Essa abordagem reduz a complexidade do problema e torna o algoritmo de fluxo máximo mais direto e eficiente.

A seguir, serão discutidos os passos para resolver cada uma das questões apresentadas no problema.

3.1. Capacidade Máxima da Rede

Para calcular a capacidade máxima da rede, foi utilizado o algoritmo de Ford-Fulkerson. Este algoritmo é uma técnica para encontrar o fluxo máximo em uma rede de fluxo. O algoritmo começa com o fluxo inicial igual a zero em todas as arestas da rede. Em seguida, ele tenta encontrar um caminho aumentante da fonte para o sumidouro, ou seja, um caminho no qual ainda é possível enviar mais fluxo, ou seja, a capacidade residual de todas as arestas nesse caminho é positiva. A busca por um caminho aumentante é feita utilizando busca em largura (BFS). Quando um caminho é encontrado, o algoritmo calcula a capacidade residual mínima ao longo desse caminho, que indica quanto fluxo adicional pode ser enviado por esse caminho. O fluxo é então aumentado ao longo desse caminho, ajustando as capacidades residuais das arestas: a capacidade das arestas ao longo do caminho é reduzida, e se houver fluxos reversos, suas capacidades residuais são aumentadas, permitindo que o fluxo seja ajustado em iterações subsequentes.

Após aumentar o fluxo, o algoritmo repete o processo de busca por novos caminhos aumentantes e o aumento do fluxo até não ser possível encontrar mais nenhum caminho válido para ampliar o fluxo. Como mencionado anteriormente, foi utilizado o grafo residual no algoritmo, garantindo que a fonte e o sumidouro sejam únicos. Ao longo da execução do algoritmo, a capacidade das arestas do grafo residual é atualizada, subtraindo a capacidade atual pelo fluxo que está sendo transmitido por elas, o que será muito útil futuramente. Ao fim do algoritmo, obtemos o fluxo máximo da rede, que corresponde à capacidade máxima que a rede pode comportar.

3.2. Energia Não Atendida

A energia não atendida representa a quantidade de energia necessária pelos consumidores que não foi suprida pelos geradores. Em outras palavras, é a parte da demanda total dos consumidores que não foi satisfeita pelo fluxo de energia no grafo. Como o fluxo de energia já foi calculado anteriormente, basta determinar a demanda total de todos os consumidores e subtraí-la da energia máxima disponível na rede.
Para calcular a energia não atendida, percorre-se o dicionário de vértices do grafo e soma-se todas as demandas. Com a demanda total obtida, subtrai-se a energia máxima. Se o resultado for positivo, ele representa a energia não atendida. Caso seja zero ou negativo, isso significa que toda a demanda foi atendida, e não há déficit de energia.

3.3. Energia Perdida

A energia perdida refere-se à quantidade de energia gerada pelos geradores que não chega efetivamente aos consumidores devido às limitações nas conexões da rede. Em outras palavras, é a diferença entre a energia gerada e a energia efetivamente utilizada ou transmitida aos consumidores.

Para calcular a energia gerada pelos geradores, soma-se as capacidades das conexões de cada gerador. Isso representa a quantidade de energia produzida pela rede a partir dos geradores. A energia efetivamente utilizada na rede, calculada anteriormente, corresponde ao fluxo máximo ou à energia máxima da rede. Para obter a energia perdida, subtrai-se a energia efetivamente utilizada da energia total gerada.

3.4. Conexões Críticas

Uma conexão crítica é uma aresta que está operando em sua capacidade máxima, ou seja, está sendo utilizada no limite do que pode transmitir. Para detectar se uma aresta está sendo utilizada ao máximo, basta verificar se a sua capacidade residual, após a execução do algoritmo de fluxo máximo, é igual a zero no grafo residual. Isso ocorre porque, no grafo residual, uma capacidade igual a zero significa que não há mais capacidade disponível para enviar fluxo adicional através dessa aresta.

Para encontrar todas as conexões críticas, percorremos a matriz de adjacência do grafo residual e verificamos se a capacidade de cada aresta é igual a zero. Caso isso seja verdadeiro, acessamos a mesma posição no grafo original para obter a capacidade original dessa conexão. Em seguida, armazenamos as informações relevantes sobre essa conexão em um vetor de tuplas, que contém todos os dados necessários, como os vértices envolvidos e a capacidade original da aresta. Esse processo nos permite identificar as conexões críticas da rede, ou seja, as arestas que estão sendo utilizadas no máximo de sua capacidade e que são fundamentais para o fluxo máximo da rede.

Por fim, como a impressão das conexões críticas deve ser de maneira decrescente, de acordo com a capacidade original, usamos uma fila de prioridade, onde vamos retirando o maior valor da fila de cada vez e, ao fim, a impressão ocorre em ordem decrescente.

4. Análise de Complexidade

Considere n como o número de vértices do grafo e m como o número de arestas do grafo.

4.1. Capacidade Máxima da Rede

Para calcular a capacidade máxima da rede, foi utilizado o algoritmo de Ford-Fulkerson. O loop principal deste algoritmo é limitado pelo número de vezes que um fluxo pode ser enviado através do grafo. A cada iteração do loop, a BFS encontra um caminho aumentante mais curto do vértice de origem ao destino. Considerando que o comprimento máximo de qualquer caminho aumentante é $n - 1$, pois não pode conter ciclos, cada aresta pode ser saturada no máximo $n - 1$ vezes. Portanto, o número total de iterações do algoritmo é limitado por $O(n \cdot m)$. No pior caso, a BFS visita todos os vértices e todas as arestas do grafo. Como a BFS é executada em cada iteração do loop, a complexidade final do algoritmo é:

$$O((n \cdot m) \cdot (n + m)) = O(n \cdot m^2)$$

Além disto, temos a complexidade espacial $O(n)$, pois na execução da BFS é criada e utilizada uma fila de tamanho n toda iteração.

4.2. Energia Não Atendida

Para calcular a energia não atendida, percorremos o dicionário de vértices uma vez para calcular a demanda total e, após isso, fazemos a subtração pela energia máxima gerada. Ou seja, a complexidade para calcular a energia não atendida é $O(n)$.

4.3. Energia Perdida

Para calcular a energia perdida, percorremos todas as arestas ligadas a todos os geradores da rede. Portanto, a complexidade depende do número de arestas associadas a cada gerador, que, no pior caso, é proporcional ao número total de arestas no grafo. Ou seja, a complexidade para calcular a energia perdida é $O(m)$.

4.4. Conexões Críticas

Para definir quais conexões são críticas, percorremos toda a matriz do grafo residual para detectar quais arestas possuem capacidade residual igual a zero. Ou seja, para obter tal informação, temos uma complexidade $O(n^2)$.

Também temos a etapa de impressão em ordem decrescente, onde utilizamos uma fila de prioridade. Esta estrutura, no pior caso, é preenchida por um número de elementos proporcional ao número de arestas do grafo. A cada remoção do maior número, que está no topo, ocorre uma reordenação interna com complexidade $O(log\, m)$. Ou seja, a complexidade do ordenamento é $O(m\, log\, m)$.

Assim, a complexidade final é $O(n^2 + m\, log\, m)$.

5. Considerações Finais

O trabalho uniu a teoria aprendida em sala de aula com a prática, sendo uma grande oportunidade de aprendizado para mim. Particularmente, a maior dificuldade foi no desenvolvimento do algoritmo de Ford-Fulkerson, pois, inicialmente, não consegui implementá-lo utilizando uma lista de adjacência. No entanto, de forma geral, o trabalho foi muito proveitoso para aplicar os conhecimentos adquiridos na disciplina.

6. Referências

Jussara Marques de Almeida. **Slides disponibilizados na disciplina de Algoritmos I**. Departamento de Ciência da Computação (UFMG), 2024, Belo Horizonte.

KLEINBERG, Jon. **Algorithm Design**. 3ª ed. revista e ampliada. 2011, Editora Cengage.

Site **GeeksForGeeks - Ford-Fulkerson Algorithm for Maximum Flow Problem**. Disponível em https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/