

ANÁLISE E SÍNTESE DE ALGORITMOS

RELATÓRIO DO 2º PROJETO

O objetivo deste projeto é implementar uma solução que permita determinar a capacidade máxima de produtos de variados produtores que uma rede é capaz de transportar, tendo em conta várias características da rede: o número de fornecedores de produtos, cada um com uma capacidade de produção; os meios de transporte, que têm associado a eles um limite máximo de produtos que conseguem transportar; e as estações de abastecimento, que permitem abastecer os meios de transporte e efetuar o controlo do mesmo, tendo também uma capacidade máxima de produtos que podem processar.

A solução do problema mencionado faz uso de uma rede de fluxo, onde os vértices representam os fornecedores, as estações de abastecimento e de controlo, um hipermercado (que recebe os produtos transportados) e uma fonte, de onde partem todos os produtos (que correspondem ao fluxo), sendo que os arcos representam os meios de transporte e a sua respetiva capacidade. O grupo decidiu implementar a solução em linguagem C, usando uma derivação do algoritmo Push Relabel.

Na solução do problema, executam-se os seguintes passos:

- Leitura do número de fornecedores, estações de abastecimento, o número de ligações que existem na rede, a produção de cada fornecedor e a capacidade de cada estação de abastecimento. Inicialização das alturas e excessos dos vértices e fluxo dos arcos ambos a 0, exceto no caso dos fornecedores, que começam com o excesso igual à sua produção, lida do stdin. Os vértices são representados pela estrutura *vertex_t*, enquanto que os arcos são representados pela estrutura *edge_t*.
- Por cada vértice que representa uma estação de abastecimento, criamos um outro vértice e dois arcos entre estes dois. Estes arcos representam a capacidade máxima da estação. É de notar que, por cada arco criado, criamos um outro, de modo a poder enviar fluxo para trás.
- Sendo que os fornecedores começam com excesso, estes são adicionados a uma fila de espera (queue), de modo a executar a função de descarga em cada um deles. Os vértices com excesso irão sendo adicionados à queue, à medida que o algoritmo é executado, de modo a que a operação de descarga

lhes seja aplicada. O algoritmo acaba quando não existirem mais vértices na queue.

- Cada vez que é executada a operação de descarga sobre um vértice u , vai-se buscar a lista de arcos de vértices adjacentes a u . Enquanto existir excesso em u , o algoritmo vai tentar enviar o excesso de fluxo para um vértice adjacente a u , através dos arcos provenientes de u . Isso só será possível caso o fluxo seja menor do que a capacidade do respetivo arco, ou seja, se o arco não estiver saturado. Este procedimento é executado, ao executar-se a operação Push.
- A operação Push é apenas executada caso o que foi anteriormente mencionado seja cumprido e caso a altura do vértice com excesso seja superior, em pelo menos uma unidade à altura do vértice adjacente. O fluxo a enviar de um vértice u para outro v , será o mínimo entre o excesso de u e a capacidade residual do arco, que corresponde à diferença entre a capacidade do arco e o fluxo no mesmo. Uma vez que o este valor mínimo de fluxo é enviado, subtrai-se o excesso ao vértice u e adiciona-se o excesso ao outro vértice. Este, como passa a ter excesso é adicionado à queue (caso ainda não tivesse na queue).
- Caso o vértice u continue com excesso, o algoritmo irá continuar a tentar encontrar um arco não saturado, de modo a enviar fluxo para outro vértice através dele. Se a lista de arcos que ligam a vértices adjacentes a u chegar ao fim e u continuar com excesso, o algoritmo irá aumentar a altura de u e tentar novamente. Só irá enviar fluxo para outro vértice adjacente a u , caso a altura de u seja superior à altura desse mesmo vértice.
- Para aumentar a altura de um vértice u , executa-se a operação de Relabel que a irá aumentar em pelo menos uma unidade. A nova altura será igual à soma de 1 com a altura mínima do conjunto dos vértices adjacentes a u , caso o arco de u para o vértice com a altura mínima não esteja já saturado.
- Assim que a queue fique vazia, o algoritmo acaba e executa-se um algoritmo de DFS sobre o grafo, a partir do vértice que corresponde ao hipermercado, de forma a detetar quais os vértices que são alcançáveis a partir do hipermercado e assim, o corte mínimo. Os arcos saturados entre um vértice alcançável e um não alcançável fazem parte do corte mínimo, correspondendo, portanto, a ligações a serem melhoradas. Caso estes arcos estejam entre uma estação de abastecimento e a sua estação duplicada (na rede de fluxo), não são apresentados como ligações a serem melhoradas,

mas sim como estações a serem melhoradas, já que esses arcos correspondem à capacidade da estação.

- Por fim, depois de obtermos as estações e ligações a serem melhoradas, o programa imprime a capacidade máxima da rede (que corresponde ao excesso do vértice que representa o hipermercado) e as estações e ligações a serem melhoradas, para o stdout.

Deste modo, o grupo utilizou uma variação do algoritmo Push Relabel (usando uma queue) e um algoritmo de DFS, como implementação de uma solução que permita determinar a capacidade máxima de produtos de variados produtores que uma rede é capaz de transportar.

As complexidades de cada um destes algoritmos são, respetivamente, $O(V^3)$, ou seja, cúbica, e $O(V + E)$, ou seja, linear, sendo V o número de vértices e E o número de arestas de um grafo. Logo, a complexidade teórica da solução é $O(V^3)$.

De forma a comprovar a complexidade teórica executou-se o programa com inputs de diferentes dimensões. O gráfico com essas execuções encontra-se apresentado abaixo.



Ao observar o gráfico, é possível concluir que o tempo de execução do algoritmo aumenta, aproximadamente, de forma cúbica, em relação ao número de vértices da rede. Logo, a complexidade teórica $O(V^3)$ é comprovada.