Ricardo Coelho da Silva LEIC-A 72843
Ana Costeira Araújo LEIC-A 73214

RELATÓRIO DO 2º PROJECTO

ANÁLISE E SÍNTESE DE ALGORITMOS

1. Introdução

O problema proposto para o projecto consiste na resolução de problemas de segurança de multidões, onde se procura uma forma de barrar as ruas que ligam os pontos críticos – pontos onde há possibilidade de haver motins - , fazendo com que a segurança da cidade feche o número mínimo de ruas que os ligam. Para impedir um motim basta isolar um dos grupos perigosos dos restantes.

Algoritmicamente falando, o problema proposto pode ser modelado através de um algoritmo de Fluxos Máximos, onde as ruas barradas equivale a saber quais são os cortes mínimos que podem ser aplicados ao grafo, assumindo uma *source* e diferentes *targets (*Aplicam-se diferentes targets porque podem existir vários pontos críticos).

Assim, existem dois algoritmos principais a serem considerados, sendo o primeiro uma versão do segundo. São eles o **Edmonds-Karp** e o **Ford-Fulkerson**.

Uma vez que, numa primeira instância, o problema não pode ser modelado com nenhum dos algoritmos, foram necessárias algumas alterações ao problema, que iremos desenvolver na explicação da solução.

2. Descrição da Solução

Para resolver o problema proposto, utilizámos o algoritmo de Fluxos Máximos de **Edmonds-Karp**. Este diferente do algoritmo **Ford-Fulkerson** essencialmente por utilizar uma *BFS*, garantindo que percorre os caminhos mais curtos primeiro. Desta forma, este algoritmo torna-se mais eficiente do que o **Ford-Fulkerson**, razão pela qual foi escolhido.

Tal como referido, várias alterações tinham de ser feitas ao problema em questão para que pudesse ser utilizado o algoritmo:

A primeira seria tornar o grafo dirigido. Atendendo ao facto de que o grafo não era dirigido, isso significa que uma ligação entre A e B poderia ser atravessada em ambos os sentidos. Dessa forma, tornar o grafo dirigido implicou apenas que existia um arco de A para B e outro de B para A.

A segunda alteração foi tornar o arco pesado, para efeitos de contabilizar as capacidades. Concluímos então que os arcos teriam de ser unitários, escolhendo para esse efeito que todos os arcos teriam fluxo de 1.

Assim, com isso em mente, optámos por representar o nosso grafo com uma lista de adjacências de pares, onde, para cada par, o primeiro elemento seria o vértice ao qual estaria ligado, e o segundo elemento seria a capacidade de fluxo. Optámos por não utilizar matriz de adjacências por ser demasiado lento.

Analisando vários grafos, foi também possível encontrar uma solução ainda mais eficiente do que a até então estudada para fazer os cortes, que funcionava para todos os casos, e que consistia em contar o número de *BFS* completadas com sucesso. São esses o número dos caminhos de aumento através dos quais é possível chegar da *source* até ao *target*.

Uma vez que todos os pesos eram unitários, muitos passos do algoritmo *standart* foram removidos, uma vez que o fluxo variava apenas entre 0 e 1, sendo este facto bastante útil para efeitos de comparações e casos de paragem.

Utilizámos ainda algumas estruturas auxiliares, como um vector para marcar os vértices visitados, e um vector com os *pais* dos vértices à medida que iam ser encontrados na *BFS* para ser possível fazer *backtracking*. Foi utilizada também uma queue para ir guardando os elementos encontrados na *BFS.*

3. Análise teórica

A complexidade do algoritmo escolhido é $O(VE^2)$, uma vez que, cada caminho de aumento pode ser encontrado em tempo O(E), ao mesmo tempo que pelo menos um dos E arcos fica saturado (atinge o fluxo máximo possivel) e que a distância é no máximo V. Isto é, o número de aumentos será O (VE), dando origem então ao $O(VE^2)$

Dado que neste projecto específico existem vários problemas e, cada um dos problemas tem vários *targets*, para cada teste, o algoritmo Edmonds-Karp é corrido P * C vezes, onde P = Número de problemas e C = número de pontos críticos. De modo que existe uma influência notória no tempo de execução quando o número de problemas e pontos críticos aumenta, para o mesmo número de V e E.

4. Avaliação experimental dos resultados.

Teste	V	E	Problemas	Tempo(s)
t1	4	4	2	0,002
t2	48	43	4	0,002
t3	131	130	16	0,01
t4	130	129	94	0,06
t5	155	154	127	0,09
t6	4525	589	1	0,01

A tabela assima relaciona o número de vértices, arcos e problemas com o tempo de execução, comprovando o que foi exposto na análise teórica.

5. Referências

http://en.wikipedia.org/wiki/Edmonds%E2%80%93Karp_algorithm
http://www.cs.fsu.edu/~asriniva/courses/alg06/Lec28.pdf
Introduction to Algorithms , Third Edition, Cormen, Leiserson, Rivest, Stein - ***The Edmonds-Karp Algorithm****[p. 728 - 730]*