Algoritmia e Desempenho em Redes de Computadores

3º Mini Projeto - *Connectivity in graphs*

Bernardo Gomes, 75573 Tomás Falcato, 75876

2 de Dezembro de 2015

1 Descrição do problema

Neste mini-projeto pretende-se avaliar a conectividade de uma rede representada por um grafo, ou seja, determinar qual o número mínimo de nós que ao retirar não permitem que o grafo esteja ligado.

Numa primeira fase, dado um par de nós do grafo, calcula-se o número mínimo de nós que é necessário retirar para que não haja nenhum caminho a ligar o par especificado.

Posteriormente, fez-se uma análise mais detalhada da rede, repetindo o processo anterior para todos os pares de nós de forma a armazenar a informação do número de nós que foi necessário retirar para cada par. Com a informação anterior, torna-se possível calcular a probabilidade cumulativa do número de nós a separar um par de nós.

Por fim, é verificado qual o número mínimo de nós que previne o grafo de ser conexo e é mostrado ao utilizador quais são os identificadores dos mesmos, por forma a que a informação do programa possa ser facilmente comprovada.

2 Abordagem ao problema

Sendo do nosso conhecimento que o número mínimo de nós que separa a fonte do destino é igual ao número máximo de caminhos independentes entre a origem e o destino, é de notar que o problema em causa é bastante semelhante à determinação de quais as arestas de um grafo que previnem que este seja conexo. No caso anterior, o problema seria reduzido a um problema de fluxos. Ora recorrendo à técnica de *vertex splitting* será possível resolver o problema de forma semelhante.

Ao dividir cada nó em dois, define-se "nó -" como o nó que irá receber todas as ligações, que o nó respetivo do grafo inicial receberia, entre os restantes e ele mesmo e que apenas tem ligação ao nó com o mesmo identificador que ele ("nó +"). Define-se "nó +" como o nó com o mesmo identificador que o nó que lhe deu origem, mantendo este as ligações entre este nó e todos os outros, que o nó respetivo do grafo inicial teria, recebendo apenas a ligação do respetivo "nó -".

Colocando as arestas que ligam as extremidades "-" e "+" de cada nó com capacidade "1" e todas as outras arestas, que já pertenciam ao grafo inicial, com capacidade infinita, aplicando o algoritmo de *Ford-Fulkerson*, é-nos possível resolver o problema enunciado.

3 Função *ford_fulkerson*

Apesar da referência ao algoritmo com o mesmo nome da função, o nosso código é uma adaptação do algoritmo em questão.

Tendo todas as arestas do grafo dado no ficheiro capacidade infinita e todas as arestas "internas" de cada nó capacidade "1", não foi necessário preocuparmo-nos com as atualizações dos fluxos. Apenas é necessário atualizar a rede residual no que respeita à inversão do sentido da aresta que liga os nós $-$ e $+$ de cada nó pertencente a dado caminho descoberto, garantindo assim que todos os caminhos descobertos em cada iteração são independentes.

O pseudo-código da função será o seguinte:

Result: number of nodes that separate the graph and its identifiers
see if there's a path from source to destination;
while *there's a path from source to destination* **do**
    invert edge link between - and + node in each original node from path;
    see if there's a path in new residual network;
end
see in discovered vector which nodes were only half discovered and count them;
if *counted nodes are less than previous counts* **then**
    pass to a string which nodes prevented the graph from being connex;
end

Algorithm 1: Função *ford_fulkerson*

Para a realização desta função, realizámos sucessivas BFSs por forma a verificar se existia um caminho independente da origem para o destino. Esta função, irá retornar um vetor *discovered* que indica quais os nós que foram descobertos e um vector *parent* que indica através de que nó o nó em causa foi descoberto.

Através desta informação, a função *path* irá reconstruir o caminho da BFS começando pelo destino. Se conseguir chegar à origem significa que o algoritmo conseguiu identificar um caminho independente dos descobertos anteriormente pelo que é necessário alterar novamente a rede residual.

Quando a função indica que não existe caminho entre os nós especificados, significa que se chegou a uma situação semelhante à ilustrada na figura 1. Nesse caso, nos nós da fronteira, onde se verifica qual o corte mínimo, a posição do vetor *discovered* irá conter informação que apenas os nós − foram descobertos ("*half discovered*").

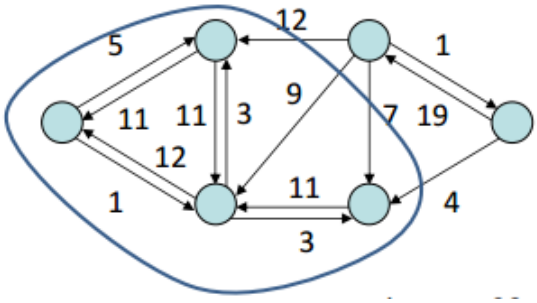

Figura 1