INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

Relatório 2º Projeto (2017/2018)

Grupo: 142 (Alameda) | Francisco Sena 86420

02 de abril de 2018

## Introdução

O problema que foi proposto consiste em segmentar as imagens que são obtidas pelos veículos de transporte de mercadorias da empresa do Sr. Caracol, ou seja, classificar os pixéis que constituem a imagem como sendo de primeiro plano ou de cenário. Para tal, recebe-se no standard input linha com o número de linhas de pixéis da imagem M (M>=2) e o número de colunas da mesma N (N>=1). De seguida, recebem-se os pesos de plano e de cenário de cada pixel e a relação de vizinhança entre todos os pixéis vizinhos.

Com estes dados, o programa irá determinar a segmentação que minimiza o peso total da imagem e também irá classificar todos os pixéis em "Cenário" ou "Plano".

## Descrição da Solução

O algoritmo utilizado na resolução do problema foi o algoritmo de *Edmonds-Karp.* Era também possível implementar o algoritmo *Relabel To Front*, mas dada a diferença de dificuldades de implementação optou-se pelo primeiro referido.

Utilizou-se uma lista de adjacências como estrutura de dados para representar o grafo dirigido, em que cada vértice V representa um pixel da imagem e as arestas E representam as relações de vizinhança entre os pixéis. As relações de vizinhança do input entre dois pixéis quaisquer são tratados como duas ligações em sentidos opostos (dirigidas) entre os pixéis.

Criou-se um vértice "source" que irá ligar a todos os pixéis com capacidade igual ao valor do peso de plano dos pixéis. Criou-se também um vértice "target" em que todos os pixéis têm uma ligação para o mesmo com capacidade igual ao valor do peso de cenário do pixel. A ideia da criação destes dois vértices permite usar um algoritmo de fluxos para que, após a sua execução, o corte mínimo irá dar uma partição dos pixéis entre "Cenário" e "Plano", onde o corte é descoberto usando uma BFS. Portanto, o corte será constituído pelo conjunto das primeiras arestas que estão saturadas, ou seja, cujos vértices de chegada dessas arestas são inatingíveis na rede residual. O valor do corte mínimo (que será igual ao valor do fluxo máximo que o *Edmonds-Karp* nos dá) irá dar o peso mínimo da segmentação da imagem. Os vértices que pertencem ao subconjunto "S" da partição resultante do corte serão de cenário e os que pertencem a "T", de primeiro plano. Quando a soma do fluxo que se faz passar de um pixel para outros até ao target tudo somado não é superior ao peso de plano do pixel, esse pixel será de cenário, caso contrário, a ligação "source" → pixel ficaria saturada e então seria de primeiro plano.

Ao receber os valores dos campos de "Cenário" de cada pixel (após se ter guardado os valores de "Plano"), faz-se passar fluxo por todos os caminhos que sejam source → pixel → target onde esse fluxo é o mínimo dos campos "C" e "P" desse pixel.

São inicializadas as seguintes estruturas de dados com respetiva descrição antes da execução do algoritmo:

- Vetor de inteiros onde vão estar marcados se os pixeis são "C" ou "P";
- Uma queue FIFO para a BFS;
- Um vetor de inteiros dedicado apenas às capacidades das ligações da source – em todos os pixéis o número de ligações é O(6), ou seja, O(1), com exceção da source. Dado que a source liga a todos os pixéis, o seu número de ligações é O(M*N), o que seria mau pois cada vez que fosse necessário aceder a uma capacidade seria necessário percorrer a lista de adjacências da source. Com este vetor, o acesso em causa é O(1).
- Uma lista de predecessores de cada caminho de aumento encontrado pela BFS;
- Vetor "currentPathCapacity" –para um pixel $p_n$, currentPathCapacity[$p_n$] contém a capacidade do caminho de aumento descoberto pela BFS até esse pixel, ou seja, a capacidade mínima de todas as ligações visitadas até esse pixel.

Depois da execução do algoritmo *Edmonds-Karp*, basta imprimir o valor do fluxo máximo determinado pelo algoritmo e percorrer o vetor "SP_values" onde os seus valores ditam se um certo pixel é de cenário ou plano.

## Análise Teórica

Para analisar a complexidade do algoritmo implementado, procede-se à análise das várias secções que o constituem.

A inicialização e construção do grafo com base no input é em tempo constante sendo que a inserção de uma aresta na lista de adjacências é efetuada em tempo constante.

Existem O(VE) caminhos de aumento até se determinar um fluxo máximo, e, para encontrar cada um deles a BFS é O(V+E), na verdade O(E) pois o grafo em causa trata-se de uma rede em que E>=V-1, então, vem que a complexidade do *Edmonds-Karp* é $O(VE^2)$. Como cada pixel tem no máximo 6 ligações, tem-se que E<=6V (é contabilizada as ligações da source para os pixéis no próprio número de ligações dos pixéis para facilitar as contas), ou seja, "E" é O(V), logo é possível exprimir a complexidade do algoritmo apenas em função de V - $O(V^3)$.

Na receção do input no que diz respeito à ideia descrita na "Descrição da solução", apesar de reduzir o tempo total de execução real do *Edmonds-Karp* (pois são executadas menos "M*N" BFS e reduz o número de iterações onde se faz passar fluxo pelo caminho encontrado pela BFS) não altera a complexidade assimptótica do algoritmo pois reduz apenas "M*N" BFSs, portanto a quantidade de caminhos de aumento continua a ser O(VE).

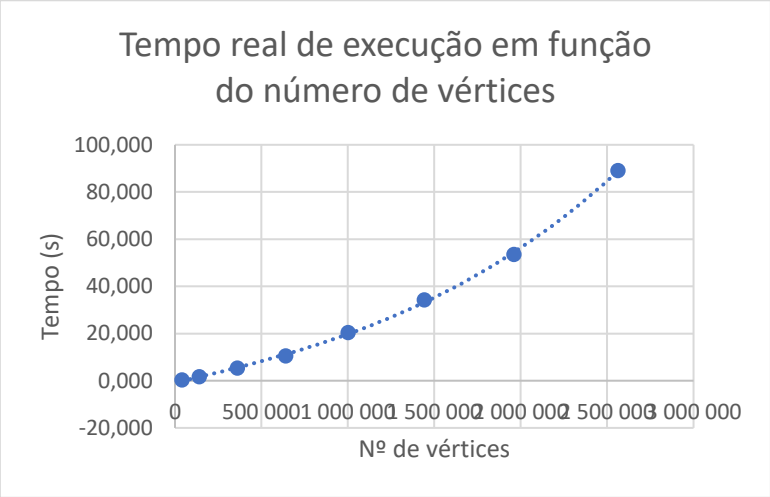
Por fim, imprimir o valor que minimiza a segmentação (fluxo máximo que o algoritmo devolve) e se cada pixel é "C" ou "P", é O(V).

Conclui-se que a complexidade de todo o programa é $O(V^3)$.

## Análise Experimental dos Resultados

Procedendo à análise do programa, realizaram-se vários testes utilizando o gerador disponibilizado. Foi utilizado o comando *time* e foi extraído o "real time" 5 a 8 vezes para cada input, sendo calculada a média aritmética entre eles para a obtenção dos valores apresentados. De seguida, construíram-se gráficos cuja curva é a que melhor se ajusta aos mesmos.

Variou-se apenas o número de pixéis (M*N) pois o número de ligações (E) varia em função do número de pixéis como referido na análise teórica.A curva descrita pelo gráfico da função não é satisfatória em relação ao que se previu na análise teórica. É curva está mais próxima de ser descrita por uma função quadrática do que cúbica. Talvez, com mais pontos com maiores valores "V", se verificasse o crescimento cúbico.




| Pixeis de input (M*N) | Tempo (ms) |
|---|---|
| 40200 | 0,190 |
| 140400 | 1,459 |
| 360350 | 4,899 |
| 620800 | 10,210 |
| 1002000 | 20,342 |
| 1462400 | 33,182 |
| 1961400 | 53,421 |
| 2561600 | 89,942 |

## Comentários e Referências

Na escolha da linguagem a ser utilizada, teve-se em conta o controlo que seria desejável ter sobre a memória utilizada e sobre todas as estruturas necessárias na implementação do projeto. Como tal, foi escolhida a linguagem de programação C.

A implementação da estrutura de dados "Grafo" foi baseada na implementação sugerida nos slides da cadeira de IAED 16/17. A implementação do algoritmo "*Edmonds-Karp"* foi baseada em implementações encontradas com pesquisa na internet.