U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Checkpoint #1

IART – Trabalho 1

João Praça – up201704748

Liliana Almeida – up201706908

Sílvia Rocha – up201704684

Especificação do Trabalho

Neste problema de otimização é pretendido que, dada uma lista de fotos e das *tags* associadas a cada foto, estas sejam organizadas num *slideshow* que obtenha a maior pontuação possível. A pontuação de um *slideshow* é dada como a soma dos fatores de interesse calculados para cada par de slides.

O fator de interesse corresponde ao mínimo entre o número de *tags* em comum entre os dois slides, o número de *tags* presente no primeiro slide mas não no segundo e o número de *tags* presente no segundo slide mas não no primeiro.

Referências

- https://github.com/kobr4/g_hashcodes
- https://github.com/AhmedSoli/Hashcode-2019
- https://github.com/LJMUAstroecology/hashcode_2019
- https://github.com/inaitana/hashcode-2019-qualification
- https://medium.com/@danieleratti/how-we-placed-1st-in-italy-and-22nd-in-the-world-google-hashcode-2019-e59e52232b4e

Formulação do problema

Representação da solução

A solução encontrada irá corresponder a um vetor de vetores, do tipo [[slide1], [slide2],...,[slideN]], otimizado de modo a que a pontuação total seja máxima.

Restrições rígidas

- Cada imagem poder ser usada 0 ou 1 vezes;
- Cada slide tem de conter uma imagem horizontal ou duas verticais;
- O slideshow tem de ter pelo menos 1 slide;
- Uma imagem não pode ser inserida num slide já cheio.

Função de vizinhança

Como função de vizinhança planeámos seguir dois métodos como base: *Hill Climbing* e *Simulated Annealing*. Para além destes, decidimos experimentar ainda uma terceira abordagem em que o estado inicial é identificado calculando os pares ótimos de slides e escolhendo o melhor destes. A partir daí, é identificada a melhor solução vertical e a melhor solução horizontal quando combinadas com o último slide já adicionado e é selecionada a melhor de entre estas.

Para encontrar a melhor solução vertical, é primeiro identificada a melhor imagem vertical em conjunto com o slide anterior e depois emparelhada a imagem vertical que, combinada com a primeira, permite a maior pontuação quando comparada ao slide anterior.

Caso a melhor solução encontrada seja de 0 pontos, ou não haja mais imagens para utilizar, o algoritmo termina.

Função de avaliação/heurística

A pontuação de dois slides adjacentes é calculada como o mínimo entre número de *tags* em comum, número de *tags* presente no primeiro mas não no segundo e o número de *tags* presente no segundo mas não no primeiro.

Trabalho de Implementação

Linguagem de Programação

Para o desenvolvimento do problema de otimização descrito, a linguagem de programação a utilizar será C++.

Ambiente de Desenvolvimento

O ambiente escolhido para desenvolvimento deste projeto foi o Visual Studio Code em Windows 10.

Estruturas de dados

- A informação de cada imagem será guardada em *structs* (com o ID, a orientação, o número de *tags* e o vetor de *tags*);
- Um slide corresponde a um vetor de imagens (de tamanho 1 ou 2);
- Um *slideshow* corresponde a um vetor de slides.

Estruturas dos ficheiros de texto

Cada conjunto de imagens encontra-se num ficheiro de texto que contém exclusivamente caracteres ASCII. Na primeira linha do ficheiro é dada a indicação do número de imagens desse conjunto. Cada linha contém a descrição da foto correspondente. Essa descrição contém: um 'V' ou 'H' que indica se a imagem é vertical ou horizontal, o número de *tags* associadas à imagem seguido dessas mesmas *tags*.

Trabalho desenvolvido

Até à data, foi já desenvolvida uma primeira versão do projeto em que resolve o problema apresentado através do uso de *Hill Climbing*. No entanto, é de notar que nesta primeira versão, as imagens verticais ainda não estão a ser agrupadas num slide só, todas estão a ser avaliadas da mesma forma.