

Tema 19 - Maximum Matching num gráfico não direcionado

Miguel Filipe Rodrigues Almeida de Matos Fazenda

Universidade de Aveiro

5 de dezembro de 2021

Resumo

Este projeto teve como objeto de análise a aplicação de dois algoritmos distintos num gráfico $G(V, E)$ com V vértices e E arestas com o intuito de encontrar um ou mais "maximum matching(s)", consoante o algoritmo. Os algoritmos alvo de estudo são:

- brute-forcing
- greedy heuristics

Keywords

brute-force, greedy heuristics, maximum, matching

Foram também utilizadas estas bibliotecas para auxiliar na criação/recolha dos dados pretendidos:

- time, para obtermos o tempo de execução de cada algoritmo
- random, para ajudar a criar vértices nos grafos de maneira aleatória
- math, para ajudar a manter a distância entre vértices
- itertools, para ajudar a realizar todas as combinações de arestas necessárias

1 Introdução

Este projeto foi realizado no âmbito da disciplina de "Algoritmos Avançados", disciplina do Mestrado em Engenharia Informática na Universidade de Aveiro.

O projeto consiste na aplicação de dois algoritmos, "brute-forcing" e "greedy heuristics", de maneira a encontrar um ou mais, consoante o algoritmo, "maximum matching(s)" dentro de um grafo não ordenado $G(V, E)$, onde V são os vértices e E as arestas.

Um "maximum matching" é um subconjunto dentro do conjunto de todas as arestas possíveis, onde as arestas não partilham vértices entre si.

2 Ferramentas e Bibliotecas

Para a realização deste projeto foram utilizadas as seguintes ferramentas:

- PyCharm, IDE para criar e correr o código
- Excel, para armazenar os resultados obtidos

3 Resultados

Os resultados encontram-se todos no ficheiro .xlsx submetido em conjunto com o código.

4 Análise dos dados

Cada resultado obtido corresponde à média de vários resultados obtidos para o mesmo "input", dado haver alguma variância.

A recolha de resultados foi feita de maneira a ver o impacto de cada variável isoladamente, isto é, onde apenas uma variável varia e as restantes mantêm-se constantes. As variáveis que foram alvo de estudo foram:

- vértices
- arestas

As zonas onde foram analisadas o impacto das variáveis foram:

- nº total de arestas possíveis
- nº de combinações de arestas pretendidas
- tempo de execução para brute-forcing

- tempo de execução para greedy heuristics
- n^o total de soluções para brute-forcing

4.1 N^o total de arestas possíveis

A quantidade de arestas possíveis pode ser calculada através da fórmula das combinações:

$$C_2^V = \frac{V!}{2!(V-2)!} \quad (1)$$

para V vértices. A presença do n^o 2 é porque uma aresta é composta por 2 vértices.

4.2 N^o de combinações de arestas pretendidas

Para as combinações de arestas pretendidas a serem analisadas, a fórmula para as calcular é a mesma utilizada para calcular o n^o total de arestas possíveis. No entanto o cálculo é feito com base no n^o total de arestas possíveis:

$$C_E^N = \frac{N!}{E!(N-E)!} \quad (2)$$

para N total de arestas existentes e E quantidade de arestas pretendidas.

4.3 Tempo de execução para brute-forcing

Tendo por base os resultados apresentados na coluna I , coluna que compara sempre o tempo "atual" com o tempo "anterior", podemos observar que a relação entre o n^o de arestas pelo tempo apresenta um crescimento logarítmico. Algo inesperado, tendo por base a complexidade do algoritmo. O crescimento inicialmente esperado seria de complexidade cúbica.

4.4 Tempo de execução para greedy heuristics

Tendo por base os resultados apresentados na coluna L , coluna que compara sempre o tempo "atual" com o tempo "anterior", a mesma situação se observa como no algoritmo brute-forcing.

4.5 N^o total de soluções

Em relação à quantidade de soluções, para o algoritmo brute-forcing, podemos confirmar que também apresenta um crescimento logarítmico em relação à quantidade de combinações de arestas possíveis.

Este crescimento logarítmico possui uma ordem cada vez menor à medida que a quantidade de vértices é aumentada.

Conclusão

De maneira a concluir, os resultados observados são bastante diferentes dos inicialmente assumidos. Onde esperaríamos um crescimento cúbico para o "brute-forcing" e quadrático a cúbico para o "greedy heuristics" na relação entre o n^o de arestas pelo tempo, o resultado obtido foi de crescimento logarítmico.

Referências