

Prob.1 – Clique of size k

Sobral, Pedro – 98491 – sobral@ua.pt

Resumo – Este relatório procura mostrar como o algoritmo “Clique of size k” funciona, quer como uma pesquisa de brute-force, quer através de uma pesquisa com uma heurística. Todos os algoritmos foram escritos em Python (3.10), e serão feitas análises sobre a complexidade computacional dos mesmos.

Abstract – This report seeks to show how the Clique of size k algorithm works, with a brute-force search and with a heuristic search. All the algorithms were written in Python (3.10), and will be made analyzes about the computational complexity of the algorithms.

I. INTRODUÇÃO

O problema apresentado, é o 1 – Cliques of size k, ou seja, dado um grafo G, um clique de G é um sub-grafo completo de tamanho k, sendo k o número de vértices desse mesmo sub-grafo. [2]

Além deste relatório, foram criados os ficheiros de código em Python, e os ficheiros que guardam os resultados dos testes dos algoritmos dos grafos gerados através da seed correspondente ao número mecanográfico. Como a geração de grafos é um processo rápido, não foi necessário guarda-los num ficheiro como sugerido no enunciado.

O problema irá ser resolvido quer por um algoritmo de pesquisa exaustiva, quer por um algoritmo de pesquisa com heurística, onde se espera que o primeiro consuma mais recursos, essencialmente tempo de execução em comparação com o primeiro, no entanto, espera-se que o primeiro encontre sempre a solução ótima e o segundo não, pois não são analisados os casos todos e são tomadas decisões com base na heurística.

Será ainda analisada uma outra implementação dos algoritmos que retornem a quantidades de cliques diferentes cada grafo analisado tem.

De modo a entender melhor o funcionamento do programa, basta correr o seguinte comando, ou ler o README do projeto:

```
$ python3 generator.py -h
```

II. PROBLEMA

O problema consiste na resolução e na procura de subgrafos de tamanho k (k corresponde ao número de vértices), de um determinado grafo G . O problema passa

por começar a procura num determinado vértice e ver se é possível completar um sub-grafo de tamanho k . [3][4][7]

O problema adaptado ao mundo real, passa pelo seguinte exemplo: O grafo representa uma rede social, e os vértices desse mesmo grafo, as pessoas que utilizam essa rede social, sendo que as arestas são as conexões das pessoas, como por exemplo serem amigos. Um clique neste contexto pode ser caracterizado como os amigos em comum, pois todos se conhecem entre todos. Este tipo de problema tem mais aplicações práticas, nomeadamente em matérias de bioinformática, e de engenharia elétrica.[1]

III. GERAÇÃO DE GRAFOS

Os grafos são gerados, criando os vértices que podem estar compreendidos entre 1 e 20, e os mesmos são identificados por dígitos, sendo posteriormente feito de forma random a atribuição das arestas com os respectivos vértices criados anteriormente. Para o contexto da resolução do problema é criada uma lista de adjacência, que nada mais é que um dicionário com o vértice como chave e com os vértices a que está ligado como valor, como o exemplo seguinte demonstra:

$$\{0: [3, 4, 9], 3: [0, 1, 5, 7, 8], 4: [0, 5], (\dots)$$

Para exemplo de demonstração, a seguinte figura demonstra um grafo gerado:

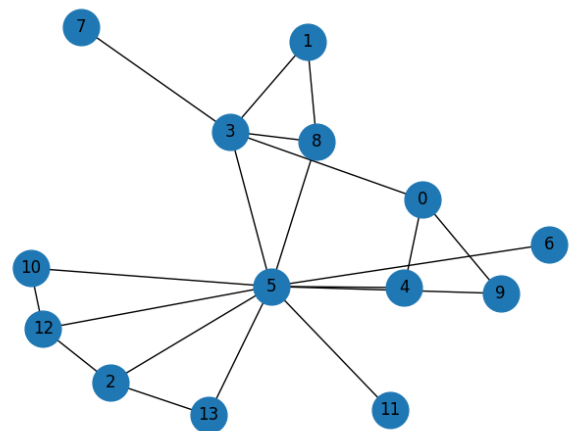


Figura 1 - Exemplo de um grafo gerado

III. ALGORITMO DE PESQUISA EXAUSTIVA

O algoritmo de pesquisa exaustiva desenvolvido para esta solução, é baseado no algoritmo de Bron-Kerbosch com pivot [5], uma forma melhorada da versão original do algoritmo. Esta versão permite ganhar tempo, e fazer o backtracking da recursão mais rápido, pois a recursão não é feita em todos os cliques como na versão base do algoritmo. A escolha de um pivot random, permite fazer menos recursões feitas pelo algoritmo e daí um ganho de tempo na sua execução. De uma forma genérica, o algoritmo funciona da seguinte forma: Cada clique tem de ter incluído o pivot ou um dos seus vértices não vizinhos, sendo portanto assim somente testar se o pivot ou os vértices não vizinhos são as escolhas certas para ser o vértice ser adicionado e continuar a recursão do algoritmo.

IV. ALGORITMO DE PESQUISA COM HEURISTICA

O algoritmo de pesquisa com heurística [6], foi desenvolvido com o seguinte conceito em mente, é necessário escolher um próximo vértice que esteja ligado a todos os que já fazem parte da solução que está a ser construída. Ainda para confirmar que a solução é uma clique, é verificado tal situação. De forma mais detalhada, o algoritmo começa por escolher um vértice dos demais do grafo, e começa por verificar se um outro vértice são vizinhos (conectados por uma aresta), sendo este processo repetido até todos estarem diretamente conectados e sendo então encontrado uma clique nova.

V. RESULTADOS

Foram tidas em conta duas abordagens, uma com grafos diferentes com o número a variar entre 5 e 80, sendo que para cada valor de vértices foram gerados 4 grafos com [12.5, 25, 50, 75] % de arestas em relação ao número de vértices, onde procurou-se por uma clique de tamanho k (sendo k o valor calculado pelas percentagens [12.5 25 50 75] com o número de vértices), e outra com grafos de 5 a 200 vértices, sendo que para cada valor de vértices foram gerados 4 grafos com [12.5, 25, 50, 75] % de arestas em relação ao número de vértices, onde foram encontrados quantos cliques de tamanho diferente cada grafo tinha.

Para comparação dos algoritmos desenvolvidos, serão avaliados os seguintes parâmetros para ambas as abordagens, tais como:

Abordagem “Há cliques de tamanho k ?”

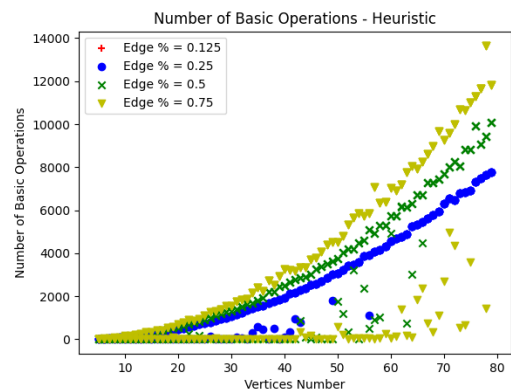
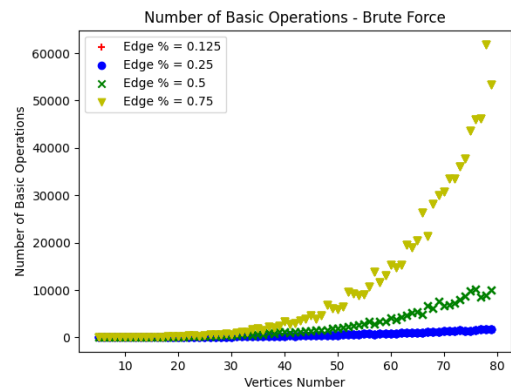
- Numero de Operações Básicas
- Tempos de execução
- Memória usada para a resolução do algoritmo.
- Analise estatística de resultados

Abordagem “Quantas cliques de tamanho diferente há?”

- Numero de Operações Básicas
- Tempos de execução
- Tamanho do Clique comparado com o número de vértices e arestas
- Memória usada para a resolução do algoritmo.
- Analise estatística de resultados

Nota: Foram somente consideradas cliques de tamanho igual ou superior a 3, visto que cliques de tamanho 2 são uma solução trivial.

A. Número de Operações Básicas

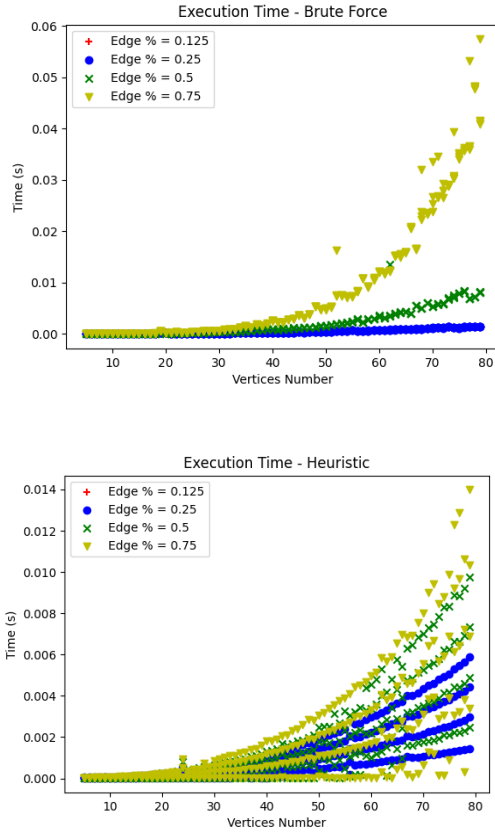


Como é possível observar pelas imagens existe uma grande discrepância entre o número de operações básicas realizadas entre os dois algoritmos, sendo o número bastante menor no algoritmo de pesquisa com heurística, onde o máximo é um valor pouco maior que 14000, e no algoritmo de pesquisa exaustiva o valor chega a mais de 60000. Ou seja, é possível concluir que o algoritmo de pesquisa com heurística faz sensivelmente 4.2 vezes menos operações que o algoritmo de pesquisa exaustiva.

Também é importante realçar que no algoritmo de pesquisa exaustiva o o número de operações básicas tem um crescimento bastante acentuado para grafos acima de 100 nós com 75% das arestas em relação aos nós, o que não se verifica de forma tão acentuada no algoritmo de pesquisa com heurística, onde o número de arestas por si

só não influencia tão diretamente o número de operações básicas.

B. Tempos de Execução



Vendo o eixo das coordenadas, em ambos os gráficos, é possível verificar que a diferença entre os dois algoritmos é notória. Enquanto o algoritmo de pesquisas exaustiva, no pior caso, demora sensivelmente 0.06 segundos, o algoritmo de pesquisa com heurística, no pior caso, demora pouco mais de 0.014 segundos, aproximadamente 5 vezes menos.

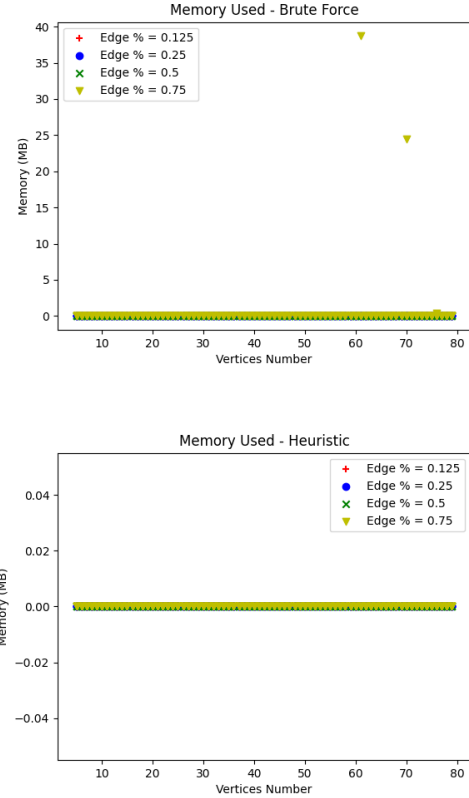
Novamente, o número de arestas influencia bastante o tempo de execução, em ambos os algoritmos, no entanto essa influência é notoriamente maior no algoritmo de pesquisa exaustiva, pois faz com que o algoritmo comece a demorar cada vez mais tempo a encontrar a solução ótima.

Como é possível ver, o algoritmo de pesquisa exaustiva tem um comportamento exponencial, e por aproximação de algumas pesquisas e de testes podemos concluir que o algoritmo tem um tempo de execução no pior caso de $O(3^{n^3})$, que é polinomial. [8]

Já o algoritmo de pesquisa com heurística, como é possível observar na figura, apresenta um comportamento exponencial, no entanto muito menos acentuado, uma vez que o tempo mais demorado de resolução é pouco superior a 14 milésimas de segundo (0.014 segundos), e visto que o seu algoritmo implementa 2 ciclos for, o tempo de

execução no pior caso será representado por $O(n^2)$, que é polinomial.

C. Memória usada para a resolução do algoritmo



Foi ainda feita uma análise ao consumo de memória por parte dos algoritmos, sendo que se nota que o consumo da mesmo nos dois algoritmos é muito residual, apresentado somente no caso do algoritmo da pesquisa exaustiva alguns casos em que houve um consumo efetivo e notável de memória.

D. Análise estatística de Resultados

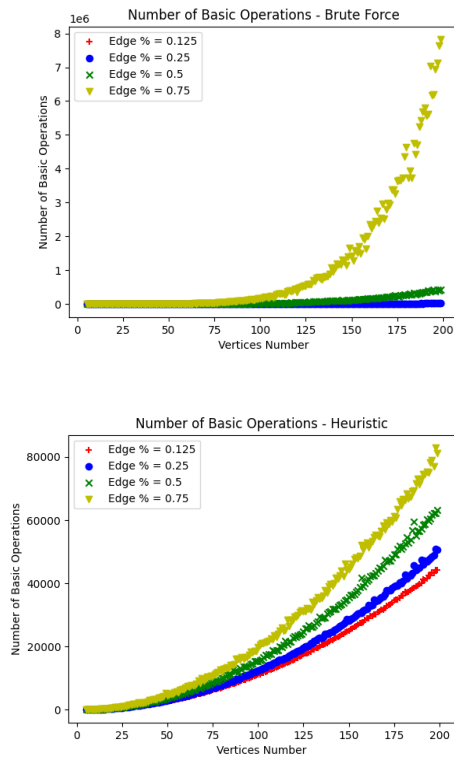
Fazendo uma comparação de resultados obtivos entre os dois algoritmos, podemos observar que dos 1200 grafos gerados e analisados, apenas há 40 resultados diferentes. Ou seja, o algoritmo de pesquisa com heurística não é 100% certo, mas apresenta uma taxa de sucesso de 96.667%, comparando os resultados com o algoritmo de pesquisa exaustiva.

Deste modo, podemos concluir que o algoritmo de pesquisa com heurística é numa vasta maioria das vezes capaz de entregar uma solução ótima com muito menos tempo de execução e número de operações básicas feitas.

Quanto à correlação dos resultados, se esta for menor que zero significa que a correlação é inversamente proporcional, igual a zero que não existe relação, e se for maior que zero, que é diretamente proporcional. Neste caso, como mostra a tabela abaixo, as variáveis apresentam uma correlação diretamente proporcional uma vez que apresentam um valor de 0.902117.

	Pesquisa exaustiva	Pesquisa com heurística
Pesquisa exaustiva	1.000000	0.902117
Pesquisa com heurística	0.902117	1.000000

E. Número de Operações Básicas

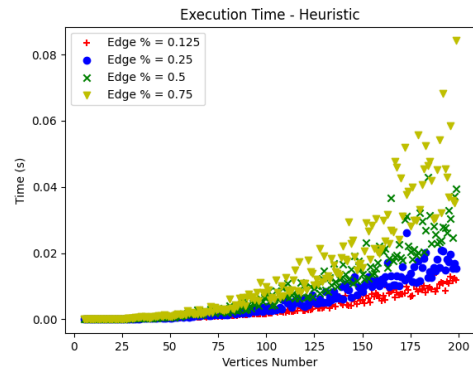
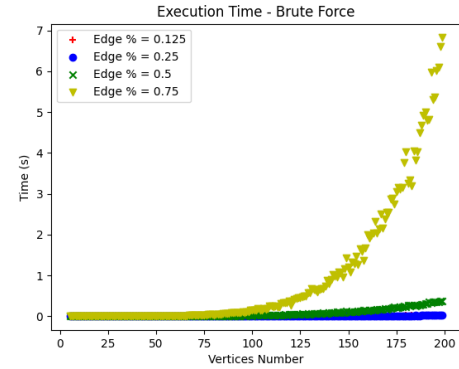


Como é possível observar pelas imagens existe uma grande discrepância entre o número de operações básicas realizadas entre os dois algoritmos, sendo o número imensamente menor no algoritmo de pesquisa com heurística, onde o máximo é um valor pouco maior que 80000, e no algoritmo de pesquisa exaustiva o valor chega aos 8000000. Ou seja, é possível concluir que o algoritmo de pesquisa com heurística faz sensivelmente 100 vezes menos operações que o algoritmo de pesquisa exaustiva.

Também é importante realçar que no algoritmo de pesquisa exaustiva o número de operações básicas tem um crescimento bastante acentuado para grafos acima de 100 nós com 75% das arestas em relação aos nós, o que

não se verifica de forma tão acentuada no algoritmo de pesquisa com heurística, onde o número de arestas por si só não influencia tão diretamente o número de operações básicas.

F. Tempos de Execução



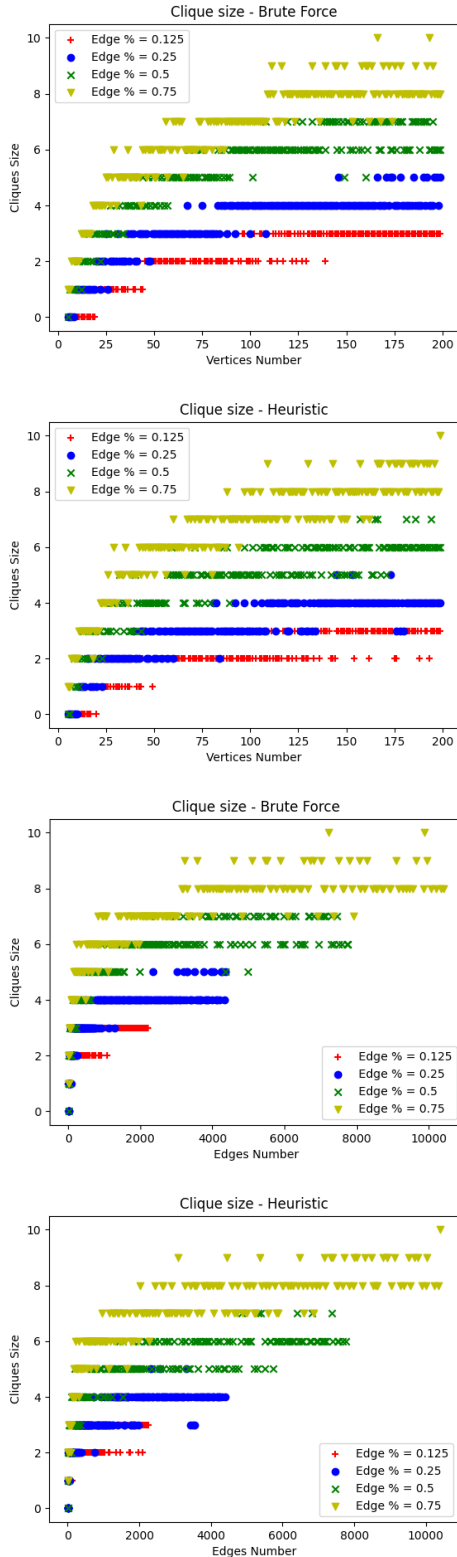
Vendo o eixo das coordenadas, em ambos os gráficos, é possível verificar que a diferença entre os dois algoritmos é notória. Enquanto o algoritmo de pesquisas exaustiva, no pior caso, demora sensivelmente 7 segundos, o algoritmo de pesquisa com heurística, no pior caso, demora pouco mais de 0.08 segundos, aproximadamente 100 vezes menos.

Novamente, o número de arestas influencia bastante o tempo de execução, em ambos os algoritmos, no entanto essa influência é notoriamente maior no algoritmo de pesquisa exaustiva.

Como é possível ver, o algoritmo de pesquisa exaustiva tem um comportamento exponencial, e por aproximação de algumas pesquisas e de testes podemos concluir que o algoritmo tem um tempo de execução no pior caso de $O(3^{n/3})$, que é polinomial. [8]

Já o algoritmo de pesquisa com heurística, como é possível observar na figura, apresenta um comportamento exponencial, no entanto muito menos acentuado, uma vez que o tempo mais demorado de resolução é pouco superior a 8 centésimas de segundo (0.08 segundos), e visto que o seu algoritmo implementa 3 ciclos for, o tempo de execução no pior caso será representado por $O(n^2)$, que é polinomial.

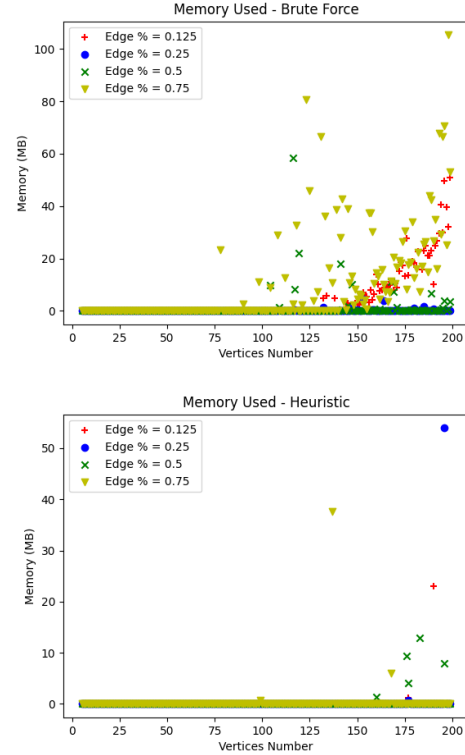
G. Tamanho da Clique comparado com o número de vértices e de arestas



De modo a poder haver uma comparação mais visual dos resultados de ambos os algoritmos, foram criados 4 gráficos anteriores, onde como se pode observar que os gráficos apresentam padrões muito semelhantes, o que

significa que os resultados dos algoritmos, neste caso a comparação do algoritmo de pesquisa com heurística é muito semelhante ao resultado do algoritmo de pesquisa exaustiva.

H. Memória usada para a resolução do algoritmo



Foi ainda feita uma análise ao consumo de memória por parte dos algoritmos, havendo uma maior utilização da memória na implementação com pesquisa exaustiva do que na implementação com pesquisa com heurística, visto que nesta implementação o consumo de memória foi praticamente nulo, com exceção de alguns grafos.

Realçar que observa-se um consumo maior em grafos com percentagem de arestas de 75% em relação aos vértices, especialmente no algoritmo de pesquisa exaustiva.

I. Análise estatística de Resultados

Fazendo uma comparação de resultados obtidos entre os dois algoritmos, relativamente ao tamanho das cliques, podemos observar que dos 780 grafos gerados e analisados, 324 apresentam um número diferente de cliques, no entanto desses 324, somente 18 apresentam um desvio superior a uma unidade. Ou seja, o algoritmo de pesquisa com heurística não tem garantias quanto a obter uma solução ótima, no entanto mesmo nos casos em que a solução não foi a correta, não ficou nada longe da solução certa, só em 18 casos é que houve um desvio superior a uma unidade.

Deste modo, podemos concluir que o algoritmo de pesquisa com heurística é na sua maioria das vezes capaz de entregar uma solução ótima com muito menos tempo de execução e número de operações básicas feitas.

Foi calculado a média, a covariância e a correlação, dos tamanhos das cliques encontradas, por cada algoritmo e obtiveram-se os seguintes resultados em relação à média de tamanhos das cliques:

- Pesquisa exaustiva: 4.410256
- Pesquisa com heurística: 4.192308

Em relação à covariância, que quantifica a independência das variáveis, (quanto mais próximo de zero mais independentes, ou seja menos a ver uma com a outra têm), temos os seguintes resultados:

	Pesquisa exaustiva	Pesquisa com heurística
Pesquisa exaustiva	4.912347	4.602646
Pesquisa com heurística	4.602646	4.730621

Quanto à correlação, se esta for menor que zero significa que a correlação é inversamente proporcional, igual a zero que não existe relação, e se for maior que zero, que é diretamente proporcional. Neste caso, como mostra a tabela abaixo, as variáveis apresentam uma correlação diretamente proporcional uma vez que apresentam um valor de 0.954782.

	Pesquisa exaustiva	Pesquisa com heurística
Pesquisa exaustiva	1.000000	0.954782
Pesquisa com heurística	0.954782	1.000000

VI. CONCLUSÃO

Em tom de conclusão, um pouco como esperado, o algoritmo de pesquisa exaustiva é mais lento e executa muitas mais operações para obter os seus resultados em comparação com o algoritmo de pesquisa com heurística.

No entanto, na abordagem “O grafo G tem uma clique de tamanho k?” o algoritmo de pesquisa com heurística é bastante certo, tendo uma taxa de sucesso de 96.667%, o que revela que a heurística procura sempre no sentido correto. Na abordagem, “Quantas cliques de tamanho diferente há?”, podemos ver que a implementação com pesquisa com heurística não é tão certa como a de

pesquisa exaustiva, no entanto na maioria dos casos obtém os mesmos resultados.

É possível concluir tal como se previa, que o algoritmo de pesquisa exaustiva alcança a solução ótima, porém é muito dispendioso a nível de tempo e de operações básicas, ao passo que o algoritmo de pesquisa com heurística, nem sempre alcança a solução ótima, sendo que quando não alcança é por pouco como mostrado na secção V. Resultados (segunda abordagem), e tem tempos de execução e de número de operações básicas que são necessários efetuar muito inferiores.

VII. REFERÊNCIAS

- [1] - https://en.wikipedia.org/wiki/Clique_problem
- [2] - <https://pt.wikipedia.org/wiki/Clique>
- [3] - <https://iq.opengenus.org/algorithm-to-find-cliques-of-a-given-size-k/>
- [4] - <https://www.geeksforgeeks.org/find-all-cliques-of-size-k-in-an-undirected-graph/>
- [5] - https://en.wikipedia.org/wiki/Bron-Kerbosch_algorithm
- [6] - <https://arxiv.org/pdf/0710.0748.pdf>
- [7] - <https://www.geeksforgeeks.org/cliques-in-graph/>
- [8] - <https://www.geeksforgeeks.org/proof-that-clique-decision-problem-is-np-complete/>