

Maximum Weight Cut Problem

Hugo Veríssimo - 124348 - hugoverissimo@ua.pt

Abstract – ... abstrato em ingles

Resumo – Este relatório apresenta a implementação e comparação de dois métodos para resolver o problema *Maximum Weight Cut*: uma pesquisa exaustiva e uma heurística gulosa. O problema *Maximum Weight Cut* com ESTE É O ANTIPO FAZER NOVO

I. INTRODUÇÃO

ja se analisou no outro relatorio a descrição do problema *Maximum Weight Cut*, [1] e ns q, super fixe

este relatoria visa explorar algoritmos com um certo grau de estocacidade/aleatorieda com vista em otimizar a complexidade e as solucoes.

para alem disso os resultados são comparados aos obtidos anteriormente

serao entao implexmmentados 3 algoritmos, nomeadamente: ... e ...

II. METODOLOGIA DA ANÁLISE

vamos usar o python por ter o modulo random e outros

ns q vamos usar os ficheiro tal e tal

e para testar os algortimos serão testados os graficos do Gset e criados por nós com o ficheiro tal

Graphs for the Computational Experiments: mine and elearnig ou links and gset

III. ALGORITMO DE 1

- falar de como sao construidos: componente aleatoria e determinisica ?

- Ensuring that no such solutions are tested more than once., como fiz isto

- quando é q o algortimo para?

Algoritmo 1 Pesquisa Exaustiva

Entrada: matriz de adjacência G

Saída: subconjuntos S e T , peso do corte $weight$

```

1: input_set  $\leftarrow \{0, 1, \dots, \text{len}(G) - 1\}$ 
2: subsets  $\leftarrow$  EMPTY LIST
3: n  $\leftarrow$  LENGTH OF input_set
    $\triangleright$  Generate all subsets
4: for r from 0 to n do
5:   for each S in combinations(input_set, r) do
6:     Add S to subsets
7:   end for
8: end for
9: best  $\leftarrow$  input_set
10: weight  $\leftarrow$  0
    $\triangleright$  Evaluate each subset
11: for each S in subsets do
12:   new_weight  $\leftarrow$  0
13:   for each i in S do
14:     for each j in input_set - S do
15:       new_weight  $\leftarrow$  new_weight +  $G[i, j]$ 
16:     end for
17:   end for
18:   if new_weight > weight then
19:     best  $\leftarrow$  S
20:     weight  $\leftarrow$  new_weight
21:   end if
22: end for
23: S  $\leftarrow$  best
24: T  $\leftarrow$  input_set - best
25: return S, T, weight

```

- complexidade

IV. ALGORITMO DE 2

...

- complexidade

- falar de como sao construidos: componente aleatoria e determinisica ?

- Ensuring that no such solutions are tested more than once., como fiz isto

- quando é q o algortimo para?

V. ALGORITMO DE 3

...

- complexidade

- falar de como sao construidos: componente aleatoria e determinisica ?

- Ensuring that no such solutions are tested more than once., como fiz isto

- quando é q o algortimo para?

VI. ANÁLISE DOS RESULTADOS

Compare the results of the experimental and the formal analysis.

todos os grafos devem ser corridos pelo menos 5 vezes, e a media dos resultados deve ser calculada e mediana do tempo , por causa dos tempos e da aleatoriedade dos resultados

Graphs for the Computational Experiments: mine and elearnig and gset
asdasds

A. (1) the number of basic operations carried out

dsadasds

B. 2 the execution time

- Determine the largest graph that you can process on your computer, without taking too much time.

- Estimate the execution time that would be required by much larger problem instances.

dsadasd

C. solution

asdad

C.1 (3) the number of solutions / configurations tested

sadsad

C.2 precision

asdasd

BIBLIOGRAFIA

- [1] J. Buhler e S. Wagon, “Basic algorithms in number theory”, *Algorithmic Number Theory*, vol. 44, 2008, <https://pub.math.leidenuniv.nl/~stevenhagenp/ANTproc/02buhler.pdf>. Accessed: 2024-11-02.