

TITULO DO TRABALHO

Hugo Veríssimo - 124348 - hugoverissimo@ua.pt

Abstract – abstrato em ingles

Resumo – abstrato em pt resumo

I. INTRODUÇÃO

introoooooooo

II. METODOLOGIA DA ANÁLISE ?

metodologia ?

III. CONTAGEM 1

O primeiro algoritmo a ser implementado foi um algoritmo de Corte Aleatório, que consiste em gerar várias soluções aleatórias, comparar as mesmas e escolher a melhor entre elas [1].

Este algoritmo pode ser então traduzido para o seguinte pseudocódigo:

Algoritmo 1 Corte Aleatório

Entrada:

- lista de arestas e respetivos pesos (*edges*)
- número de vértices (*n_nodes*)
- número de soluções a gerar (*solutions*)

Saída: subconjuntos *S* e *T*, peso do corte (*weight*)

```

1: best_solution ← None
2: weight ← 0
3: seen_solutions ← empty set
4: for i ← 1 to solutions do
5:   partition ← random partition of the nodes
6:   if length(seen_solutions) = 2n_nodes then
7:     break
8:   end if
9:   partition_hash ← hash the partition
10:  if partition_hash ∈ seen_solutions then
11:    continue
12:  end if
13:  Add partition_hash to seen_solutions
14:  new_cut_weight ← compute the cut weight
15:  if new_cut_weight > weight then
16:    weight ← new_cut_weight
17:    best_solution ← copy of partition
18:  end if
19: end for
20: S ← set of nodes assigned to 0 in best_solution
21: T ← set of nodes assigned to 1 in best_solution
   return S, T, weight

```

Quanto ao número de soluções testadas, a partir da Fig.

TABELA I: Complexidade dos algoritmos pelo número de operações básicas.

| Algoritmo | Complexidade |
|---------------------|-------------------|
| Corte Aleatório | $O(m)$ |
| Simulated Annealing | $O(m)$ |
| Guloso Aleatório | $O(m^2 \times n)$ |

ffyf

IV. CONCLUSÃO

conclusaooooo

BIBLIOGRAFIA

- [1] Anupam Gupta, “15-854: Approximations algorithms”, 2014, <https://www.cs.cmu.edu/afs/cs/academic/class/15854-f05/www/scribe/lec02.pdf>. Accessed: 2024-11-28.