

## 1 Critérios de Correção

A distribuição de pontos do trabalho será feita do seguinte modo:

1. Implementação (código): até 6 pontos, dependendo da qualidade do código e dos resultados;
  - 1.1. Implementação da heurística Lagrangiana (3.0 pt)
  - 1.2. Implementação da meta-heurística (3.0 pt)
2. Relatório: até 4 pontos, dependendo da qualidade do documento;
3. Comparativo dos grupos: bônus de até 1 ponto.
4. Bônus:
  - 4.1. Heurística(s) Lagrangiana(s) alternativa(s), junto de sua descrição e análise comparativa com as demais variações. (+1 ponto)
  - 4.2. Meta-heurística(s) alternativa(s), junto de sua descrição e análise comparativa com as demais variações. (+1 ponto)

## 2 Avaliação

Em cada subseção, as notas de um grupo são apresentadas. A coluna **Item** se refere aos itens dos critérios de correção descritos na Seção 1 e a coluna **Nota** se refere à nota obtida no respectivo item. Detalhes sobre o comparativo dos grupos podem ser vistos na Seção 3.

Alguns trabalhos geraram soluções inviáveis, por algum dos seguintes motivos:

- I(1) Grafo gerado não tem  $n - 1$  arestas.
- I(2) Grafo não é conexo.
- I(3) Árvore não é geradora.
- I(4) Violação das restrições de grau.

Quando isso ocorre, o tipo de inviabilidade é indicado nas observações.

### 2.1 Grupo 01

Item	Nota	Item	Nota
1.1.	2.5	2	3.5
1.2.	1.5	3	0.4
4.1.	-	4.2.	-
<b>Total</b>	7.9		

**Observações:**

- Rigorosamente, a meta-heurística não é um GRASP pois não possui busca local. Possui somente a fase construtiva aleatorizada.
- Pelo que está descrito no relatório, a escolha do tamanho da lista restrita de arestas (10%) e do número de iterações ( $5|E|$ ) parece arbitrária. Seria interessante pelo menos mencionar outros valores testados e porque esses foram escolhidos, mostrando, por exemplo, qual foi a diferença de qualidade de solução em algumas instâncias.
- Não foram apresentados os resultados para todas as instâncias.
- Algumas soluções geradas são inviáveis: I(1).

## 2.2 Grupo 02

Item	Nota	Item	Nota
1.1.	3.0	2	4.0
1.2.	3.0	3	0.8
4.1.	-	4.2.	1.0
<b>Total</b>	11.8		

Observações:

## 2.3 Grupo 03

Item	Nota	Item	Nota
1.1.	3.0	2	4.0
1.2.	3.0	3	1.0
4.1.	-	4.2.	1.0
<b>Total</b>	12.0 - 0.05 = 11.95		

Observações:

- Arquivo de solução é gerado com os vértices com rótulos iniciados em 0. -0.05

## 2.4 Grupo 04

Item	Nota	Item	Nota
1.1.	2.0	2	3.0
1.2.	2.0	3	0.6
4.1.	-	4.2.	-
<b>Total</b>	7.6		

Observações:

- No relatório, não ficou claro como a correção de violação é feita. Qual o critério para substituição das arestas? Guloso ou a primeira aresta que garante viabilidade? Uma heurística para obtenção de uma solução inicial também é utilizada, mas não é descrita no relatório.

- No relatório, a descrição dos valores de parâmetros escolhidos está vaga. Seria interessante pelo menos mencionar outros valores testados e porque esses foram escolhidos, mostrando, por exemplo, qual foi a diferença de qualidade de solução em algumas instâncias.
- Algumas soluções geradas pela são inviáveis: I(4).

## 2.5 Grupo 05

Item	Nota	Item	Nota
1.1.	1.0	2	3.0
1.2.	1.5	3	0.2
4.1.	-	4.2.	-
<b>Total</b>	5.7 - 0.05 = 5.65		

### Observações:

- No relatório, não ficou claro qual o método utilizado para obtenção de soluções primais na relaxação lagrangiana. O único texto que menciona tal estratégia é “deslocamento de aresta”.
- `segmentation fault` quanto o solver escolhido é a relaxação lagrangiana. (variável `solvern` fica com valor 59, não existindo essa posição no vetor de solvers).
- Saída do programa está inconsistente com o que foi especificado no enunciado. -0.05
- Arestas repetidas no arquivo de solução.
- Algumas soluções geradas são inviáveis: I(1).

## 3 Comparativo dos grupos

Nessa seção, descrevemos como foi conduzido o experimento para realização da análise comparativa dos grupos. As instâncias utilizadas foram as mesmas distribuídas juntas do enunciado do trabalho. O seguinte ambiente computacional foi utilizado:

- Processador: Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz
- 32GB de memória RAM
- Sistema operacional Ubuntu 16.04.6 LTS
- gcc versão 5.4.0 20160609

Os parâmetros estabelecidos foram de 180 segundos (3 minutos) como limite de tempo, com 3 segundos de tolerância. Caso o programa excedo o limite de tempo, o processo é terminado. Como algumas implementações estavam erradas, os seguintes critérios foram adotados para computar o gap de otimalidade:

- Se a solução encontrada para a respectiva instâncias é inviável, o valor do limitante primal passa a ser  $\infty$ .
- Em alguns casos, o valor reportado do limitante primal difere do obtido ao somar o peso das arestas do arquivo de solução. Então, considera-se a soma dos pesos como limitante primal.

- Seja  $i$  uma instância com melhor limitante primal (entre todos os grupos)  $p_i^*$ , considerando apenas soluções viáveis. Se, para  $i$ , o limitante dual de um grupo  $g$ ,  $d_i^g$ , for maior que  $p_i^*$ , então define-se  $d_i^g = 0$ .

A Tabela 1 apresenta os valores de *rank* médio e  $f$  (cf. enunciado) para todos os grupos. Note que se um subconjunto de grupos tem o mesmo valor de *gap*, então todos esses receberão o mesmo *rank*.

Tabela 1: Valores de *rank* médio e  $f$  para todos os grupos.

<b>grupo</b>	<b>rank médio</b>	<b>f</b>
01	3.325	0.4
02	1.950	0.8
03	1.275	1.0
04	2.900	0.6
05	3.825	0.2