

3.1

a) Escolha uma configuração para o AG constituída de probabilidade mutação, de crossover, condição de parada (ex. máximo de gerações por execução) e tamanho da população inicial e subsequentes. Planeje também quantas execuções (1.000) fará do algoritmo e execute todas elas com a mesma configuração. Esta escolha de configuração deve ser feita após alguns testes preliminares.

R: MAX_EXECUCOES: 1000

TAM_POP: 32

MAX_GERACOES: 200

PROB_CROSSOVER: 75%

PROB_MUTACAO: 4%

d) Ao final de todas as execuções, salve as informações (itens b e c) de uma das execuções que produziu o melhor resultado.

R: b) 1,158

2,164

3,164

4,166

5,166

6,166

7,170

8,177

9,186

10,186

11,187

12,191

13,196

14,196

15,196

16,196

17,196

18,199

19,199

...

29,199

30,199

31,200

32,202

33,202

...

173,202

174,202

175,206

176,206

...

199,206

200,206

c) 11,113,206,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,1

Mochila com 11 ITENS

Mochila com 113 KG

Mochila com 206 VALOR

- 1) Explique como procedeu para penalizar o fitness de mochilas infactíveis (mochilas cujo peso é maior do que o máximo permitido).

a.

```
private void calcularFitnessPenalizacao() {
    if(this.peso > CAPACIDADE_KG_MOCHILA) {
        int dif = this.peso - CAPACIDADE_KG_MOCHILA;
        float media = mediaValorKG();
        this.valor -= (int) dif*media;
    }
}
```

- b. Se o peso da mochila for maior que o limite, diminui do valor o excesso do peso multiplicado pela média de valor por KG.

- 2) Explique como procedeu para reparar mochilas infactíveis (cujo peso é maior do que o máximo permitido), ou seja, para que tivessem seu peso ajustado para atender à restrição de capacidade.

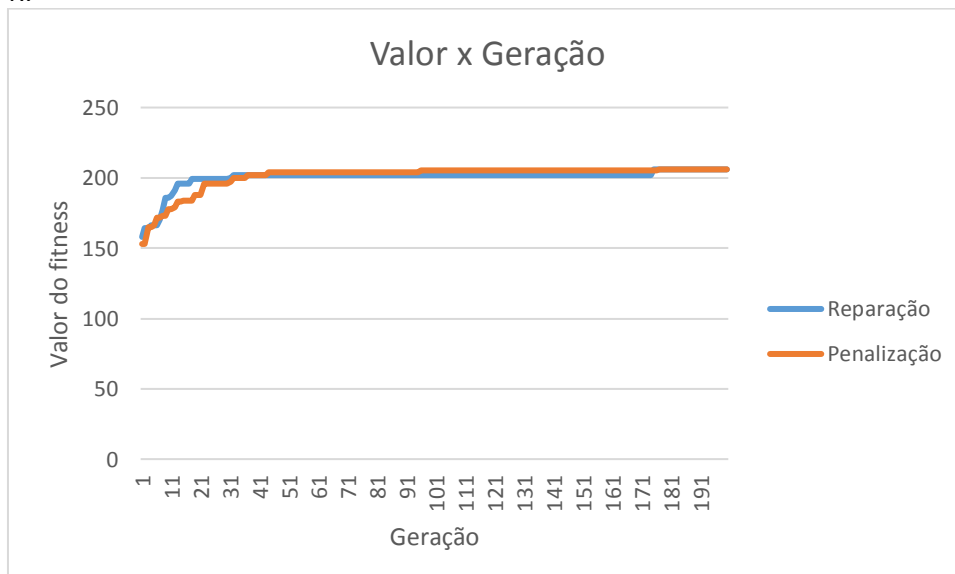
a.

```
private void calcularFitnessReparacao() {
    if(this.peso > CAPACIDADE_KG_MOCHILA) {
        int item;
        while(this.peso >= CAPACIDADE_KG_MOCHILA){
            item = s.nextInt(QTD_ITENS_DISPON);
            if(colocado.get(item) == true){
                retirarItem(item);
            }
        }
    }
}
```

- b. Se o peso da mochila for maior que o limite, enquanto o limite ainda tiver sido excedido, retira um item da mochila aleatoriamente.

- 3) Plote um gráfico valor do fitness x geração da execução na qual obteve o melhor fitness, portanto, o gráfico deve ter 2 curvas (uma para cada implementação). Responda: as curvas variam em função do modo de cálculo de fitness: penalização x reparação? Explique.

R:



Não variam, ambas seguem a mesma curva em ambos os modos de calculo.

- 4) Sobre as melhores soluções obtidas, responda para cada uma das implementações:
- a. Qual foi o valor máximo para os itens de uma mochila que você encontrou (sem violar a capacidade em Kg da mochila)?
R: 206
 - b. Quantas mochilas com valor máximo foram obtidas?
R: Penalização: 289, Reparação: 264
- 5) Compare a taxa de sucesso das implementações penalização e reparação. Para este problema, taxa de sucesso é o número de vezes que a solução de maior valor (possivelmente a ótima) foi encontrada nas execuções realizadas (recorda-se que o total de execuções foi definido no item a do método). Responda:
- a. Quais foram as taxas de sucesso obtidas?
R: Penalização: 28,9%; Reparação: 26,4%
 - b. Quantas vezes o cálculo de fitness é executado para a configuração em questão por execução? Escreva a fórmula.
R: 6400. $F(x) = \text{TAM_POP} * \text{MAX_GERACOES}$
 - c. Qual método implementado é mais custoso temporalmente: o de reparação ou de penalização?
R: O de reparação.