

=====

Tarefa AG: Problema de empacotamento (Mochila)

=====

### 1. Explique o método da Penalização em linguagem natural.

A penalização consiste em, caso o valor do peso da mochila seja maior que a capacidade máxima, diminuir o valor do FITNESS pelo quadrado da diferença entre o peso máximo e o peso da mochila. Se esse resultado for menor que zero, ele passa a ser 1, para evitar FITNESS negativa.

```
private void calcularFitnessPenalizacao() {  
    // Penalizar individuos que violam a capacidade da MOCHILA  
    if (this.peso > CAPACIDADE_KG_MOCHILA)  
    {  
        this.valor = this.valor - (this.peso-CAPACIDADE_KG_MOCHILA)*(this.peso-CAPACIDADE_KG_MOCHILA);  
        if (this.valor < 0)  
        {  
            this.valor = 1;  
        }  
    }  
}
```

### 2. Explique o método da Reparação em linguagem natural.

Caso a mochila tenha um peso maior que a capacidade máxima, um item é removido da mochila. O processo se repete até o peso dos itens ser menor que a capacidade máxima da mochila. Esses itens são escolhidos de forma aleatória, e caso ele não esteja na mochila, é tentado novamente, até conseguir.

```
private void calcularFitnessReparacao() {  
    while (this.peso > CAPACIDADE_KG_MOCHILA)  
    {  
        int n;  
        Random rand = new Random();  
        do  
        {  
            n = rand.nextInt(OTD_ITENS_DISPON);  
        }while (retirarItem(n));  
    }  
}
```

## 4. Gráficos

### a. As curvas variam em função do tamanho da população? Explique.

Sim, no geral quanto maior a população mais rápida é a convergência.

**b. As curvas variam em função do modo de cálculo de fitness: penalização x reparação? Explique.**

Sim, a reparação apresenta uma convergência mais rápida. Porém, o de penalidade apresenta melhores resultados para populações menores.

**5. Taxa de sucesso = número de vezes que a solução de maior valor foi encontrada em 200 execuções.**

**a. Quais foram as taxas de sucesso obtidas?**

114 Reparação.

20 Penalização.

**b. Quantas vezes o cálculo de fitness é executado para a configuração em questão por execução? Escreva a fórmula.**

A função responsável por calcular o FITNESS (calcularFitnessMochila) é chamada dentro de laço "for", com as condições (int i = TAM\_POP; i < 2 \* TAM\_POP; i++). Dessa forma, o cálculo é executado um número de vezes igual ao tamanho da população, só que para cada geração. Então, temos que:

Quantidade de cálculo de FITNESS = MAX\_EXECUCOES \* MAX\_GERACOES \* TAM\_POP  
Isso resulta em 1280000, considerando 200 execuções, 200 gerações e tamanho de população 32.

**c. Qual método implementado é mais custoso temporalmente: o de reparação ou de penalização?**

O método de reparação é mais custoso temporalmente, pois não só envolve a função de remover itens da mochila, como também essa remoção é feita através de tentativas, repetindo caso não consiga. Já a função de penalização apenas altera o valor do FITNESS, fazendo uma rápida conta.

**6. Sobre as melhores soluções encontradas:**

**a. Qual foi o valor máximo encontrado para os itens de uma mochila (logicamente, sem violar a capacidade em Kg da mochila)?**

206.

**b. Quantas mochilas diferentes com valores máximos?**

3.

**c. Liste todas as mochilas que obteve que apresentaram valor máximo. Para cada uma delas coloque os itens, valor total e peso total.**

11,113,206,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,1  
11,114,206,0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,1  
12,114,206,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,1