Tarefa AG: Problema de empacotamento (Mochila)

Objetivos de Aprendizagem

- Compreender algoritmos genéticos na sua forma canônica
- Compreender a importância dos parâmetros de configuração em AGs
- Compreender como realizar avaliação de AGs

Enunciado

Um viajante deve levar consigo apenas uma mochila. Essa mochila possui uma capacidade limitada e deve ser carregada apenas com objetos que serão úteis durante a viagem. Cada objeto é único e possui um peso e um determinado valor. Quais objetos devem ser levados pelo viajante de forma a **maximizar o valor dos itens colocados na mochila sem ultrapassar sua capacidade máxima de peso**?

Nesta tarefa deverá ser tratado o problema da mochila binária: cada item pode ser escolhido no máximo 1 vez e há apenas uma mochila. Este problema pode ser formulado algebricamente como:

$$\begin{aligned} &\text{maximizar } f(x) = \sum_{i=1}^{n} x_i v_i \\ &\text{sujeito a } \sum_{i=1}^{n} x_i w_i \leq C, x_i \in \{0,1\} \end{aligned} \quad (1)$$

onde v_i é o valor i-ésimo objeto, w_i o seu peso, x_i indica se o objeto aparece ou não na mochila e C define a capacidade da mochila (em termos de peso).

Obj$_i$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Peso(kg)	3	8	12	2	8	4	4	5	1	1	8	6	4	3
Valor	1	3	1	8	9	3	2	8	5	1	1	6	3	2
Obj$_i$	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**
Peso(kg)	3	5	7	3	5	7	4	3	7	2	3	5	4	3
Valor	5	2	3	8	9	3	2	4	5	4	3	1	3	2
Obj$_i$	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**
Peso(Kg)	7	19	20	21	11	24	13	17	18	6	15	25	12	19
Valor	14	32	20	19	15	37	18	13	19	10	15	40	17	39

`Capacidade da mochila C=113 Kg`

Objetivo da tarefa

Analisar e comparar o comportamento de duas implementações de um AG canônico para o problema em questão:

1. implementação com uma função de **reparação** de mochilas infactíveis e
2. implementação com uma função de **penalização** de mochilas infactíveis.

Mochilas infactíveis são aquelas que violam a restrição de capacidade, isto é, a soma dos pesos dos itens é superior à capacidade da mochila.

Método

Baixe os códigos fonte em Java para a tarefa em questão composto por 3 arquivos: *Mochila.java, AGCOperador.java* e *AGMochila.java***.** Este último contém o Main e os parâmetros de execução do AG. Ao executá-lo, será produzida uma saída textual contendo:

a) o número sequencial da geração (id) e o fitness do melhor indivíduo por geração. Por exemplo, abaixo, na primeira iteração o valor de fitness do MELHOR INDIVÍDUO foi de 143 e assim por diante.

```
run:
1,143
2,143
3,162
4,162
5,162
6,162
...
MAX_GERACOES, fitness
```

b) Ao final da execução, o programa mostra a melhor mochila encontrada no formato CSV – valores separados por vírgula. O formato sé o seguinte:
<*qtd itens, peso, valor, lista sequencial de presença ou ausência de cada um dos itens na mochila***>**:
11,113,206,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,1

c) Ao final da execução, o programa mostra a melhor mochila encontrada com os respectivos itens:

```
Mochila  peso valor
-------------------
item[ 4]    2     8
item[ 9]    1     5
item[10]    1     1
item[15]    3     5
item[18]    3     8
item[24]    2     4
item[30]   19    32
item[34]   24    37
item[35]   13    18
item[40]   25    40
item[42]   19    39
---------------------
Mochila com  11 ITENS
Mochila com 112 KG
Mochila com 197 VALOR
---------------------
```

Responda/faça:

1. Na classe Mochila.java, implemente o método **calcularFitnessPenalizacao()** de forma que mochilas infactíveis (cujo peso é maior do que o máximo permitido) sofrem uma penalização no fitness.
    a. Coloque o código do método de penalização no documento a ser entregue.
    b. Explique o método em linguagem natural.

2. Na classe Mochila.java, implemente o método **calcularFitnessReparacao()** de forma que mochilas infactíveis (cujo peso é maior do que o máximo permitido) sejam reparadas, ou seja, tenham seu peso ajustado para atender à restrição de capacidade.

   a. Adicione o código do método ao documento a ser entregue.
   b. Explique o método em linguagem natural.

3. Configure o AG para rodar com probabilidade de crossover igual a 75%, mutação igual a 4% e máximo de gerações = 200. Para cada uma das implementações (penalização e reparação), varie o tamanho da população utilizando os valores 8, 16 e 32. Para cada tamanho da população, execute 200 vezes o AG. Guarde os valores de evolução do fitness, conforme o item *a* do método, para o melhor indivíduo obtido para cada tamanho de população.

4. **Plote um gráfico** *valor do fitness x geração* da execução na qual obteve o melhor fitness. Isto deve ser feito para cada configuração do item 3, portanto, o gráfico deve ter 6 curvas. Responda:
   a. As curvas variam em função do tamanho da população? Explique.
   b. As curvas variam em função do modo de cálculo de fitness: penalização x reparação? Explique.
5. Compare **a taxa de sucesso** das implementações (penalização x reparação) para a configuração que utiliza tamanho da população=32. Definimos neste problema taxa de sucesso como o número de vezes que a solução de maior valor foi encontrada em 200 execuções. Responda:
   a. Quais foram as taxas de sucesso obtidas?
   b. Quantas vezes o cálculo de fitness é executado para a configuração em questão por execução? Escreva a fórmula.
   c. Qual método implementado é mais custoso temporalmente: o de reparação ou de penalização?
6. Sobre a(s) **melhor(es) solução(ões)** obtida(s):
   a. Qual foi o valor máximo encontrado para os itens de uma mochila (logicamente, sem violar a capacidade em Kg da mochila)?
   b. Quantas mochilas diferentes com valores máximos?
   c. Liste todas as mochilas que obteve que apresentaram valor máximo. Para cada uma delas coloque os itens, valor total e peso total.