

# Tarefa AG: Problema de empacotamento (Mochila)

## 1 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

## 2 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 da mochila sem ultrapassar sua capacidade máxima de peso?**

Existem diferentes tipos de problemas da mochila em função da distribuição de itens e mochilas (ex. itens repetidos ou não, várias mochilas com capacidades diferentes). Nesta tarefa deverá ser tratado o problema da mochila 0/1: cada item pode ser escolhido no máximo uma 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 <sub>i</sub>	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 <sub>i</sub>	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 <sub>i</sub>	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  $C = 120$  Kg**

## 3 Objetivo da tarefa

Analisar e comparar o comportamento da implementação de um AG canônico para o problema em questão utilizando duas técnicas distintas:

1. implementação de uma função de reparação de indivíduos infactíveis e
2. implementação de uma função de penalização aos indivíduos infactíveis.

### 3.1 Método

Faça os passos abaixo para a implementação com reparação de indivíduos infactíveis e para a implementação com penalização. Elas serão comparadas.

- a) Implemente os passos básicos do AG canônico conforme visto em sala (as etapas de reparação/penalização ocorrem após a geração dos filhos e antes da seleção dos sobreviventes para a próxima geração).
- b) Escolha uma configuração para o AG constituída de probabilidade mutação (Pm), de crossover (Pc), e tamanho da população (Np). Esta escolha de configuração <Pm,Pc,Np> deve ser feita após alguns testes preliminares (neste processo avalie a influência dos parâmetros nos resultados). Condição de parada (ex. máximo de gerações ou máximo de avaliações de fitness) deve ser definida como 500 gerações ou 500\*Np avaliações.
- c) Para cada configuração, execute o algoritmo com diferentes sementes do gerador aleatório (no mínimo 10 execuções para cada configuração)
- d) Para cada execução, guarde  
<Ex\_i, best\_ind\_Ex\_i, fitness\_best\_ind\_Ex\_i>

Ex\_i: i-ésima Execução do AG (com uma semente do gerador aleatório diferente)

best\_ind\_Ex\_i: melhor indivíduo da Execução i

fitness\_best\_ind\_Ex\_i: fitness do : melhor indivíduo da Execução i

A melhor solução na população final da i-ésima execução e seu respectivo fitness

Ao final calcule a média de fitness dentre as execuções. Essa é a métrica da qualidade do algoritmo.

- e) Ao final de cada execução, o programa deve salvar também a melhor mochila encontrada com as seguintes informações <quantidade de itens, peso total, valor total, lista de presença ou ausência de cada item na mochila>:  
*11,113,206*,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0,1

PARA ENTREGAR (trabalho individual)

1. Explique como procedeu para penalizar o fitness de mochilas infactíveis (mochilas cujo peso é maior do que o máximo permitido).
  - a. Adicione o **pseudo-código** do método implementado no PDF a ser entregue.
  - b. Explique o método em linguagem natural.
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. Adicione o **pseudo-código** do método ao PDF a ser entregue.
  - b. Explique o método em linguagem natural.
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.

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)?
  - b. Quantas mochilas com valor máximo foram obtidas?
  - c. Liste todas as mochilas que obteve que apresentaram valor máximo. Para cada uma delas coloque os itens, valor total e peso total.
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?
  - 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?

**Bonus Extra:** Implemente o AG usando codificação inteira (permutação) e os operadores específicos (crossover PMX e mutação por swap). Compare com os resultados com o AG implementado com reparação para as mesmas condições de parada.