

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 <i>i</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>i</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>i</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

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 inactiváveis e
2. implementação de uma função de penalização aos indivíduos inactiváveis.

3.1 Método

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

- 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.
- b) Para cada execução, guarde o identificador da geração (número sequencial) e o fitness do melhor indivíduo. No exemplo abaixo, tem-se que na primeira iteração o valor de fitness do MELHOR INDIVÍDUO foi de 143 e assim até o MAX_GERAÇÕES.

Exemplo

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

- c) Ao final de cada execução, o programa deve salvar 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>*:

11,113,206,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,1

Esta informação impressa produz a seguinte saída:

| 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 | | |
| ----- | | |

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

4 PARA ENTREGAR

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