

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 indivíduos (mochilas) infactíveis e
2. implementação com uma função de **penalização** de indivíduos (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

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

- 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). *Você pode utilizar o esqueleto de implementação fornecido pelo professor ou fazer um a partir do zero.*
- Escolha uma configuração para o AG constituída de:
 - probabilidade mutação (P_m),
 - probabilidade de crossover (P_c), e
 - tamanho da população (N_p).

Esta escolha de configuração $\langle P_m, P_c, N_p \rangle$ 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 $500 \cdot N_p$ avaliações.
- Para cada configuração, execute o algoritmo com diferentes sementes do gerador aleatório (no mínimo 10 execuções para cada configuração)
- Para cada execução, guarde $\langle ex_i, best_ind_ex_i, fitness_best_ind_ex_i \rangle$, tal que:
 - ex_i : número sequencial da execução
 - $best_ind_ex_i$: melhor indivíduo da execução i
 - $fitness_best_ind_ex_i$: fitnesses do melhor indivíduo da execução i
- Ao final da i -ésima execução, calcule a média dos $fitness_best_ind_ex_i$ sendo esta a métrica de qualidade do algoritmo para a configuração escolhida.

Para quem for usar o código fornecido pelo professor

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:

- 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
```

- 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>:
11,113,206,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0
,0,0,1,0,0,0,1,0,1,0,1
```

- 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		

PARA ENTREGAR:

- 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.
 - Coloque o código do método de penalização no documento a ser entregue.
 - Explique o método em linguagem natural ou com um pseudo-código.
- 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.
 - Adicione o código do método ao documento a ser entregue.
 - Explique o método em linguagem natural ou com um pseudo-código.
- Escolha três configurações diferentes para o AG e execute **no mínimo 10** vezes para cada uma delas. Calcule a métrica de qualidade do algoritmo para a configuração. Faça isto para cada uma das implementações (penalização e reparação) utilizando as mesmas configurações. **Compare os valores da métrica de qualidade obtidos** na forma de uma tabela e, também, respondendo
 - Os valores variam em função de quais parâmetros? Explique.
 - Os valores variam em função do modo de cálculo de fitness: penalização x reparação? Explique.

		REPARAÇÃO			PENALIZAÇÃO	
EXECUÇÃO	CONFIG 1	CONFIG 2	CONFIG 3	CONFIG 1	CONFIG 2	CONFIG 3
1						
2						
...						
10						
MÉDIA						

- Plote um gráfico *valor do fitness x geração* da execução na qual obteve o melhor fitness. Isto deve ser feito para a solução por REPARAÇÃO e por PENALIZAÇÃO. Portanto, o gráfico conterá duas curvas.
- Compare a **taxa de sucesso** das implementações (penalização x reparação) para a melhor configuração de REPARAÇÃO e a melhor configuração de PENALIZAÇÃO. Definimos neste problema taxa de sucesso como o número de vezes que a solução de maior valor foi encontrada nas *i* 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.