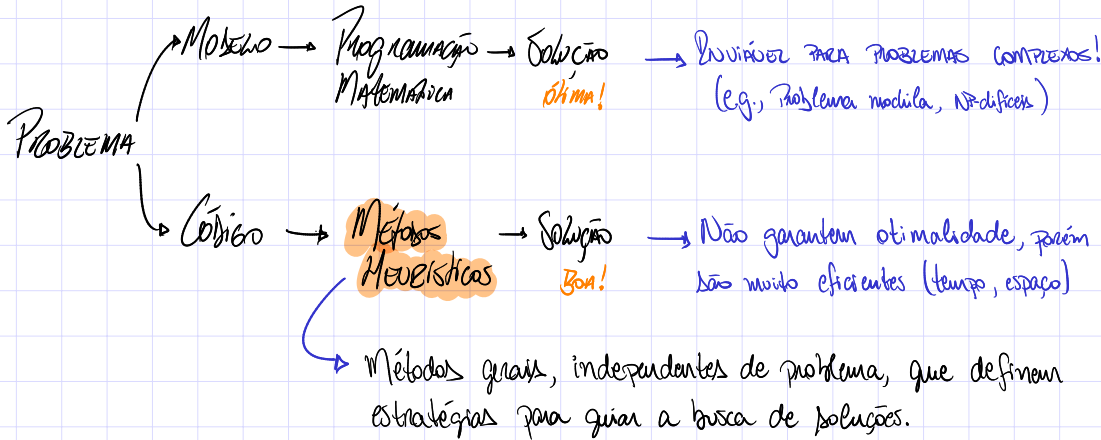


Métodos Heurísticos: Introdução e Construção

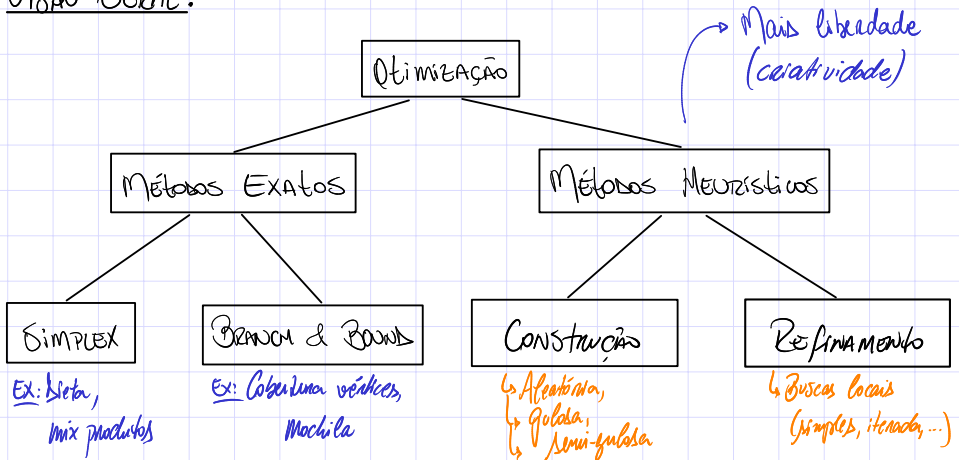


CAP. 1, 2, 4
Sec. 3.1

CAP. 2



Visão Geral:

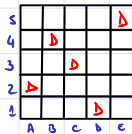


Representação de Soluções

- ↳ Específica para cada problema
- ↳ Define o tamanho do espaço de soluções
- ↳ " a complexidade das operações
- ↳ Permite excluir (algumas) soluções inviáveis

Exemplos:

- n-queens

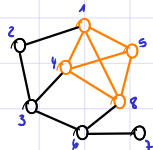


PE: $X_{ij} \in \{0, 1\} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$R = (2, 4, 3, 1, 5)$
 $\begin{matrix} A & B & C & D & E \end{matrix}$

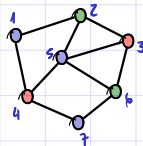
↳ + Matriz repetição! $\rightarrow [2, 4, 3, 1, 5]$

- Clique máximo (max clique)



$C = (1, 0, 0, 1, 1, 0, 0, 1)$
 $\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix}$

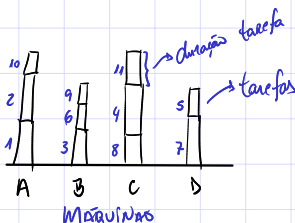
- Coloração de vértices



Seja $\{1, 2, \dots, c\}$ o conjunto de cores

$C = (1, 3, 2, 2, 1, 3, 1)$
 $\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix}$

- Pm II Cmax



$T = (A, A, B, C, D, B, B, C, B, A, C)$
 $\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{matrix}$

Heurística Construtiva: Inicia com uma solução vazia e iterativamente adiciona componentes/elementos até que uma solução completa é produzida.

↳ Ou seja, constrói uma solução "do zero"!

CONSTRUÇÃO (G):

$S \leftarrow \emptyset$

$C \leftarrow$ elementos candidatos

Enquanto $C \neq \emptyset$ e Solução não Completa FAÇA

$c \leftarrow$ Selecciona $c \in C$ Conforme estratégia G

 SE c pode ser inserido em S ENTÃO

$S \leftarrow S \cup \{c\}$

$C \leftarrow C \setminus \{c\}$

RETORNA S

ESTRATÉGIA (G)	DESCRIÇÃO	ALGORITMO
ALEATÓRIA	Selecciona um elemento aleatoriamente	Construção aleatória
GULOSA	Selecciona o melhor elemento	Construção gulosa
SEMI-GULOSA	Selecciona um dos K melhores elementos	Construção semi-gulosa

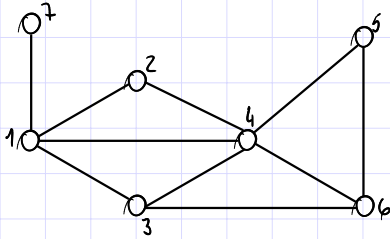
↳ Parâmetro K :
1) Constante (ex: $K=10$)
2) Percentual da solução (ex: $K=10\%$ elementos)

Para a construção (semi) gulosa, os elementos devem ser ranqueados conforme uma função heurística f .

- ↳ Se f não depende da solução parcial $\rightarrow f$ é estática,
- ↳ Caso contrário $\rightarrow f$ é dinâmica / ABERTATIVA.

↳ Neste caso, f é recalculada a cada iteração.

Exemplo: Cobertura de Vértices



Representação: Vetor binário

Elementos: $\{1, 2, 3, 4, 5, 6, 7\}$

Estratégia (G): Global

Função Heurística (F): grau do vértice

↳

Vértice	grau
1	4
2	2
3	3
4	5
5	2
6	3
7	1

Execução:

Iteração	Solução	f.o.	Completa?
1	{4}	1	x
2	{4, 1}	2	x
3	{4, 1, 3}	3	x
4	{4, 1, 3, 6}	4	✓

↳ Desempenho?

- Escolha aleatória
- Segunda função heurística

↳ Executar construção gulosa com $f =$ número de arestas não cobertas ligadas ao vértice
↳ função dinâmica/adaptativa!

CONSTRUÇÃO PARA O PROBLEMA DA MOCHILA

Instância: Capacidade = 113

Item	Weight	Profit
1	32 lbs.	727 \$
2	40 lbs.	763 \$
3	44 lbs.	60 \$
4	20 lbs.	606 \$
5	1 lbs.	45 \$
6	29 lbs.	370 \$
7	3 lbs.	414 \$
8	13 lbs.	880 \$
9	6 lbs.	133 \$
10	39 lbs.	820 \$

Sol. ótima

(1,0,0,1,0,0,1,1,1,1)

$Z = 3580$

Peso total: 113

Heurística 1: Seleciona o item de maior profit que cabe na mochila.

Repete até não haver mais itens que caibam na mochila.

Resultado: (0,1,0,1,1,0,0,1,0,1), $Z = 3114$

Ordenando pelo Profit:

Heurística 1

Item	Weight	Profit
8	13 lbs.	880 \$
10	39 lbs.	820 \$
2	40 lbs.	763 \$
1	32 lbs.	727 \$
4	20 lbs.	606 \$
7	3 lbs.	414 \$
6	29 lbs.	370 \$
9	6 lbs.	133 \$
3	44 lbs.	60 \$
5	1 lbs.	45 \$

Σp	Σw
✓ 880	13
✓ 1700	52
✓ 2463	92
x -	-
✓ 3069	112
x -	-
x -	-
x -	-
x -	-
✓ 3114	113

Resultado:

Step	Item	Add ?	Total weight	Total profit
1	8	yes	13 lbs.	880 \$
2	10	yes	52 lbs.	1700 \$
3	2	yes	92 lbs.	2463 \$
4	1	no		
5	4	yes	112 lbs.	3069 \$
6	7	no		
7	6	no		
8	9	no		
9	3	no		
10	5	yes	113 lbs.	3114 \$

$S = (1,1,1,0,1,0,0,0,0,1)$

$Z = 3114$

Peso total = 113

Heurística 2: Igual à anterior, mas seleciona os itens pela sua eficiência, i.e. pelo custo-benefício $\frac{\text{Profit}}{\text{Weight}}$.

Ordenando pela eficiência:

Ver Heurística 2

Item	Weight	Profit	Efficiency	Σp	Σw
7	3 lbs.	414 \$	138.00 \$ per pound	✓ 414	3
8	13 lbs.	880 \$	67.69 \$ per pound	✓ 1299	16
5	1 lbs.	45 \$	45.00 \$ per pound	✓ 1339	17
4	20 lbs.	606 \$	30.30 \$ per pound	✓ 1945	37
1	32 lbs.	727 \$	22.72 \$ per pound	✓ 2672	69
9	6 lbs.	133 \$	22.17 \$ per pound	✓ 2805	75
10	39 lbs.	820 \$	21.03 \$ per pound	X —	—
2	40 lbs.	763 \$	19.08 \$ per pound	X —	—
6	29 lbs.	370 \$	12.76 \$ per pound	X 3175	104
3	44 lbs.	60 \$	1.36 \$ per pound	X —	—

Resultado:

Step	Item	Add ?	Total weight	Total profit
1	7	yes	3 lbs.	414 \$
2	8	yes	16 lbs.	1294 \$
3	5	yes	17 lbs.	1339 \$
4	4	yes	37 lbs.	1945 \$
5	1	yes	69 lbs.	2672 \$
6	9	yes	75 lbs.	2805 \$
7	10	no		
8	2	no		
9	6	yes	104 lbs.	3175 \$
10	3	no		

$$S = (1, 1, 1, 1, 1, 1, 0, 0, 1, 0)$$

$$Z = 3175$$

$$\text{Peso total} = 104$$

Construção Repetida: Constrói N soluções usando uma heurística construtiva não-determinística (semi-gulosa). Ao final, retorna a melhor solução. (incumbente)

Construção Repetida (N , Construção):

$S^* \leftarrow \text{Construção}$

PARA $i \leftarrow 1, \dots, N$ FAÇA

$S \leftarrow \text{Construção}$

SE S é melhor que S^* ENTÃO

$S^* \leftarrow S$

RETORNA $S^* \rightarrow \text{solução incumbente}$

Em vez de um número de iterações N , pode-se usar um tempo limite de execução, ou outro critério de terminação, como número máximo de iterações em estagnação (i.e., sem melhoria).

Guloso Iterativo (Iterated Greedy): Iterativamente aplica uma destruição à solução atual, seguida de uma etapa de Reconstrução. O parâmetro Δ define o tamanho da destruição.

Iterated Greedy (Δ , Destruição, Construção, N):

$s^* \leftarrow \text{Construção}$

$s \leftarrow s^*$

PARA $i \leftarrow 1, \dots, N$ FAÇA

$s \leftarrow \text{Destruição}(s, \Delta)$

$s \leftarrow \text{Construção}(s)$

SE s é melhor que s^* ENTÃO

$s^* \leftarrow s$

RETORNA s^*

→ Ou outro critério

Reconstrução a partir de uma solução parcial.
(destruída parcialmente)

→ Melhor solução encontrada (incumbente)

Destruição: Remove Δ elementos da solução

↳ Ex: Remove a cor de Δ vértices → Destruição
e atribui novas cores → Reconstrução
(coloração de vértices)

Ex: Sete Δ itens para os → Destruição
e atribui novos valores → Reconstrução
(problema da mochila)

Resumo:
Construção repetida → Construção independente
Iterated greedy → Usa informações da solução construída antes → Construção dependente.