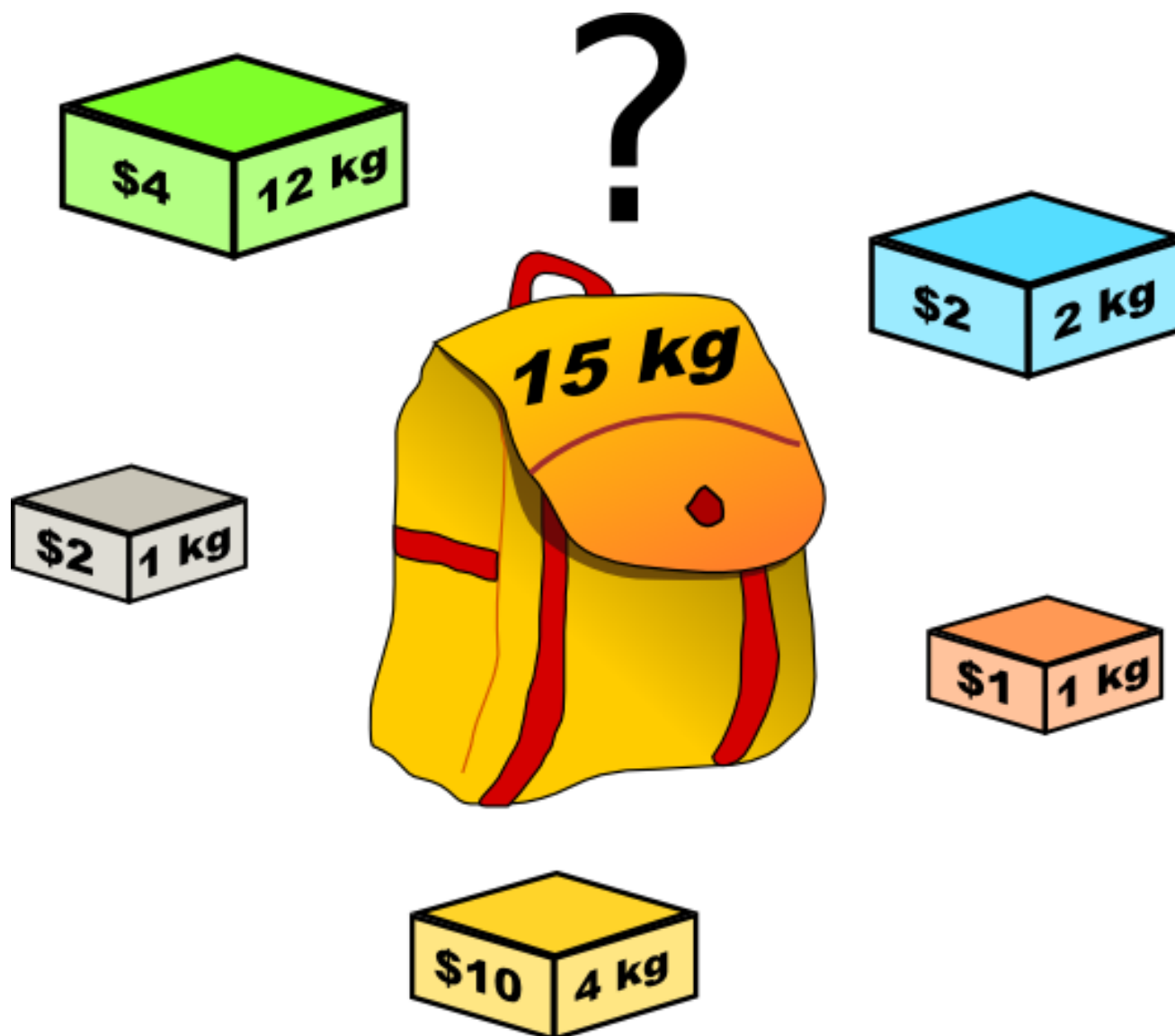


# **Algoritmos Genéticos para o Problema da Mochila**

Estéfane G. M. de Lacerda  
UFRN/DCA  
Setembro/2008

# O Problema da Mochila



# O Problema da Mochila

- Dados um conjunto de  $n$  objetos e uma mochila com:
  - $c_j$  = benefício do objeto  $j$
  - $w_j$  = peso do objeto  $j$
  - $b$  = capacidade da mochila
- Determinar quais objetos devem ser colocados na mochila para maximizar o benefício total de tal forma que o peso da mochila não ultrapasse sua capacidade.

# O Problema da Mochila zero-um

*(do inglês, 0-1 knapsack problem)*

$$\text{Maximizar } z = \sum_{j=1}^n c_j s_j$$

$$\text{Sujeita a } \sum_{j=1}^n w_j s_j \leq b$$

$$s_j \in \{0, 1\}$$

Uma solução  $s$  é um vetor de uns e zeros.

Se o objeto  $j$  está na mochila então  $s_j = 1$ , caso contrário  $s_j = 0$ .

# Algoritmo Genético

- **Cromossomo**

- A solução  $s$  (um vetor de uns e zeros) é naturalmente representada por um cromossomo binário.

- **Operadores binários padrão**

- Crossover de 1-ponto (ou 2-pontos, etc)
  - Mutação (invertendo os bits)

# Uma Instância do Problema da Mochila

Objeto ( $j$ )	1	2	3	4	5	6	7	8
Benefício ( $c_j$ )	3	3	2	4	2	3	5	2
Peso ( $w_j$ )	5	4	7	8	4	4	6	8

Capacidade da mochila:  $b = 25$

---

**11001110 (cromossomo válido)**

peso =  $5+4+4+4+6 = 23 \leq 25$

função objetivo =  $3+3+2+3+5 = 16$

**11111001 (inválido)**

peso =  $36 > 25$

Função objetivo = 16

# Como Lidar com Indivíduos Inválidos?

- Solução 1 – reparar o indivíduo
- Solução 2 – penalizar a função objetivo

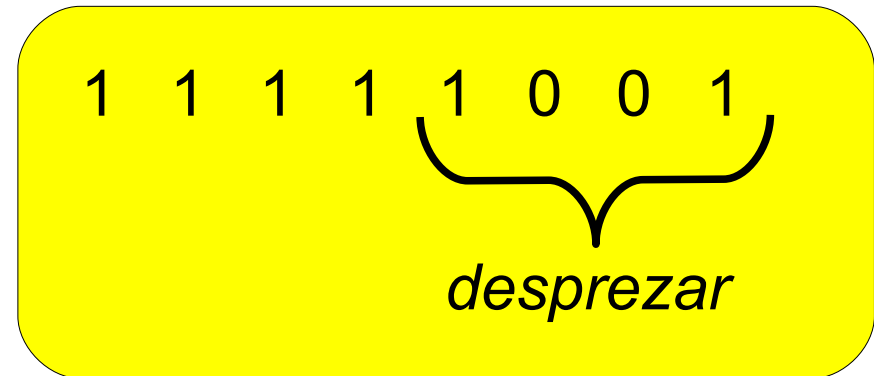
# Reparando o Indivíduo

- Indivíduo inválido

- 1 1 1 1 1 0 0 1
- peso = 36 > 25
- Função objetivo = 16

- Indivíduo “reparado”

- 1 1 1 1 0 0 0 0
- Peso = 24 (ok!)
- Função objetivo = 12



*visitar cada bit da esquerda para a direita e desprezar os bits que invalidam a solução.*



# Reparando o Indivíduo

- Por qual ordem dos bits devem ser visitados?
  - Da esquerda para direita?
  - No sentido oposto?
  - Aleatoriamente?
- Algoritmo Guloso
  - Visitar primeiro os bits com a maior razão benefício/peso;
  - Pode produzir melhores resultados.

# Penalizando a Função Objetivo

Um exemplo de penalidade é:

$$f(s) = \sum_{j=1}^n c_j s_j - \alpha \times \max\left(0, \sum_{j=1}^n s_j w_j - b\right)$$

Onde  $\alpha$  é um coeficiente de penalidade igual a:

$$\alpha = \sum_{j=1}^n c_j = 14$$

Objetos que ultrapassam a capacidade da mochila são penalizados.

# Penalizando a Função Objetivo

- **Exemplo**

- 1 1 1 1 1 0 0 1
- peso = 36 > 25
- Função original = 16
- Função com penalidade =  $16 - 14 \times (36 - 25) = -138$