Problema da mochila

Thiago Rafael Becker

Novembro de 2011

Introdução

Definição do problema. O problema, em sua forma original, é definido como: *Dado um conjunto de itens, cada um com um peso e um valor, como encher uma mochila como tal que o peso total não ultrapasse o limite de peso da mochila, e o valor total seja o maior possível.*

Importância do problema. Problemas similares aparecem em economia (alocação de recursos com limitação de investimento, busca de estratégias com limitação de risco), combinatória, matemática aplicada, logística (*bin packing*), criptografia (*subset sum problem*), corte de materiais com despedício mínimo.

Problema de decisão. O problema, como proposto, não serve para a análise de sua classe de complexidade, pois não é um problema de decisão. Para torná-lo um problema de decisão, estabelecemos um limite mínimo V ao valor máximo dos itens na mochila, levando ao seguinte problema de decisão: *Dado um conjunto de n itens I, cada um com um peso p_i e um valor v_i $(i \in I)$, é possível encher uma mochila de forma que o peso total do itens não ultrapasse o limite de peso P e o valor total seja pelo menos V.*

Problema da mochila-0,1. Este problema surge quando os itens são indivisíveis. De tal forma, para cada item no conjunto, a decisão é de colocá-lo na mochila (1) ou não (0).

Problema da mochila-0,1

Com o problema de decisão definido acima, podemos definir o problema com as seguintes restrições

$$\left(\sum_{i \in I} x_i v_i \right) \geq V \tag{1}$$

$$\left(\sum_{i \in I} x_i p_i \right) \leq P \tag{2}$$

Nas restrições acima, $x_i \in \{0, 1\}$.

Solução por força bruta

Como existem n itens, existem 2^n subconjuntos de I. O algoritmo-solução procura por cada um dos subconjuntos pela solução ótima, o que levaria a um problema de complexidade $c_p^{\leq} = O(2^n)$.

Solução em tempo pseudo-polinomial (programação dinâmica)

Por programação dinâmica, é possível obter uma solução de complexidade $O(nP)$.

Subestrutura ótima. Considere a carga mais valiosa $S = \{s_1, s_2, s_3, ..., s_n\}, S \subseteq I$ que pesa no máximo P. Se removermos o item j de S, a carga restante deve ser a mais valiosa que pesa no máximo $P - p_j$ obtida de $n - 1$ itens de $S - j$.

Esta subestrutura ótima pode ser traduzida para a função recursiva abaixo.

$$B[i,w] = \begin{cases} 0 & \text{se } i = 0 \text{ ou } w = 0 \\ B[i-1,w] & \text{se } p_i > w \\ max(B[i-1,w], B[i-1,w-p_i] + v_i) \end{cases} \tag{3}$$

Nesta equação, B é uma matriz de tamanho $(n+1) \times (P+1)$ que irá conter os resultados intermediários do cálculo.

A resposta para o problema de decisão se encontrará na célula $B[n,P]$ da matriz resultante. Se $B[n,P] \geq V$, o problema está satisfeito.

Algoritmo. O pseudo-código da solução encontra-se abaixo.

```
for w = 0 → P do
  B[0, w] = 0
end for
for i = 1 → n do
  B[i, 0] = 0
end for
for i = 1 → n do
  for w = 0 → P do
    if p_i ≤ w then
      if v_i + B[i − 1, w − p_i] > B[i − 1, w] then
        B[i, w] = v_i + B[i − 1, w − p_i]
      else
        B[i, w] = B[i − 1, w]
      end if
    else
      B[i, w] = B[i − 1, w]
    end if
  end for
end for
return B[n, P]
```

Complexidade do problema

A complexidade do algoritmo acima é $O(nP)$, na qual n é a quantidade de elementos do conjunto de entrada, e P é o peso máximo suportado pela mochila. É, portanto, pseudo-polinomial, pois não depende exclusivamente do tamanho do conjunto que compõe o preoblema, mas também dos parametros numéricos do problema.

Prova da plenitude-NP

Subset sum. O problema *subset sum* será usado para provar que problema da mochila é NP-Completo. O problema é proposto da seguinte maneira: *Dado um conjunto de inteiros X, existe um subconjunto S de X (X incluso) tal que $(\sum S_i) = c$?*

Caso especial de problema da mochila. O caso especial do problema da mochila usado para a prova será tal que $P = V$ e $(\forall i \in I)(v_i = p_i)$. Com essa equivalencia, é possível transformar a restrição 2 na seguinte restrição:

$$\left(\sum_{i \in I} x_i v_i \right) \leq V \tag{4}$$

e igualando as restrições 1 e 4 pelo somatório comum, chegamos a conclusão de que a restrição do caso especial é

$$V \leq \left(\sum_{i \in I} x_i v_i \right) \leq V \Rightarrow \left(\sum_{i \in I} x_i v_i \right) = V \tag{5}$$

Analogamente,

$$\left(\sum_{i \in I} x_i p_i\right) = P \tag{6}$$

Este problema é equivalente ao problema de *subset sum*, Portanto o problema da mochila é NP-Completo.