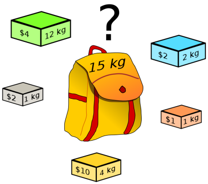
Problema da Mochila Binária

Fernando Gomes, Leonardo Holtz

Universidade Federal do Rio Grande do Sul



Sumário

Definição intuitiva

Dado um conjunto de itens, com cada item tendo um peso e um custo, qual a escolha de itens tal que a soma de seus pesos é menor que a capacidade de uma mochila a soma de seus custos é a maior possível?

Definição matemática: problema de otimização

Dados uma mochila com capacidade máxima W, um conjunto de n itens $x_1, x_2, ..., x_n$, cada um com um peso w_i e um valor v_i:

$$\text{maximizar} \sum_{i=1}^{n} v_i x_i$$

$$\text{sujeito a } \sum_{i=1}^{n} w_i x_i \leq W \text{ e } x_i \in \{0, 1\}$$

Neste caso, x_i representa o número de instâncias do item i dentro da mochila.

Definição matemática: problema de decisão

Dados uma mochila com capacidade máxima W, um valor mínimo V, um conjunto de n itens $x_1, x_2, ..., x_n$, cada um com um peso w_i e um valor v_i:

$$\text{maximizar} \sum_{i=1}^{n} v_i x_i \text{ t.q.}$$

$$\sum_{i=1}^{n} w_i x_i \leq W \text{ e } x_i \in \{0, 1\}$$

$$\sum_{i=1}^{n} v_i x_i \geq V \text{ e } x_i \in \{0, 1\} \text{ ?}$$

O problema de decisão se torna: "Existe uma seleção de x_i que satisfaça essa definição?"

Aplicações

Estudar na noite de anterior de uma prova:
Um estudante que tem uma prova no dia seguinte e tem um conjunto de capítulos para estudar, cada qual com uma probabilidade de cair na prova e um tempo necessário para estudo. Selecionar quais capítulos estudar antes da prova é análogo ao problema da mochila binária, e como veremos em breve, é *NP-completo*.
Outros exemplos:

- Gerenciadores de download
- Programas espaciais

Algoritmo de verificação

Para provar que PM[1] $\in$ NP, precisamos mostrar que existe um algoritmo capaz de verificar se um certificado do PM é correto em tempo polinomial.

[1] PM = Problema da mochila binária

Algoritmo em pseudocódigo

P: conjunto de pesos p_i
V: conjunto de valores v_i
VMIN: valor minímo para o problema de decisão
W: carga máxima da mochila
n: tamanho do problema
CERTIFICADO: seleção $x_1, ..., x_n$ $t.q.$ $x \in \{0, 1\}$

Pré-algoritmo

```
VerificaPM(P, V, VMIN, W, n, CERTIFICADO)
    ...
    returns SIM or NÃO
```

Algoritmo completo

```
VerificaPM(P, V, VMIN, W, n, CERTIFICADO)
    PesoTotal = 0
    ValorTotal = 0
    for i in 1:n
        PesoTotal += CERTIFICADO[i] * P[i]
        ValorTotal += CERTIFICADO[i] * V[i]
    endfor
    if PesoTotal <= W && ValorTotal > VMIN
        return True
    else
        return false
```

Análise de complexidade

- As atribuições das linhas 2 e 3 tem complexidade $\mathcal{O}(1)$.
- O laço *for* executa n vezes, e a complexidade das atribuições e multiplicações são constantes. A complexidade total do laço é $\mathcal{O}(n)$
- A complexidade total do algoritmo é, portanto, $\mathcal{O}(n)$.
- O problema da mochila (como problema de decisão) pode, então, ser verificado em tempo polinomial. Com isso concluímos que pertence ao conjunto NP.

Soma de subconjuntos

- Para demonstrar que PM $\in$ NP-difícil, utilizaremos uma redução do problema da soma de subconjuntos, que já sabemos que é NP-completo, para o problema da mochila.
- Note que estamos utilizando problemas de decisão, mas se demonstrarmos que eles pertencem a NP-difícil, é evidente que o problema de otimização associado também será NP-difícil (o problema de decisão é mais "fácil").

Soma de subconjuntos

Dado um conjunto de n números inteiros positivos $a_1, a_2, ..., a_n$, e um valor $t \geq 0$, existe um subconjunto destes números tal que:

$$\sum_{i=1}^{n} a_i x_i = t \text{ e } x_i \in \{0, 1\}$$

x_i indica se a_i faz parte do subconjunto (1) ou não (0)

O problema de decisão é similar ao anterior: "Existe uma seleção de a_i que satisfaça essa definição?"

Redução inst A para inst B

A redução das instâncias do Problema da Mochila para a Soma de subconjuntos é trivial, já que devemos somar um subconjunto de valores a partir de todo um conjunto em ambos os casos.

Para todos os n itens $x_1, x_2, ..., x_n$ fazemos a seguinte transformação:

$$w_i = a_i$$

$$v_i = a_i$$

Para a capacidade máxima W e o valor mínimo V, fazemos a seguinte transformação:

$$W = t$$

$$V = t$$

Redução inst A para inst B

Com isso, conseguimos mapear as exatas mesmas respostas do problema de decisão da Soma de subconjuntos para o problema de decisão da Mochila.

$\sum_{i=1}^{n} w_i x_i \leq W$ terá as mesmas respostas (sim ou não) que $\sum_{i=1}^{n} a_i x_i = t$

$\sum_{i=1}^{n} v_i x_i \geq V$ terá as mesmas respostas (sim ou não) que $\sum_{i=1}^{n} a_i x_i = t$

Algoritmo de redução

n: tamanho do problema

t: soma total formada por um subconjunto dos valores a_i

a: vetor com os valores a_i

Algoritmo de Redução

```
ReducaoSSCparaPM(n, t, a)
    W = t
    V = t
    for i in 1:n
        w[i] = a[i]
        v[i] = a[i]
    endfor
    return n, w, v, W, V
```

Análise da complexidade

- As atribuições das linhas 2 e 3 tem complexidade $\mathcal{O}(1)$.
- Nas linhas 4 a 7, o laço *for* executa n vezes, e a complexidade das atribuições são constantes. A complexidade total do laço é $\mathcal{O}(n)$.
- O retorno na linha 8 é constante.
- A complexidade total do algoritmo é, portanto, $\mathcal{O}(n)$.
- Com isso, provamos que o Problema da Mochila $\in$ NP-difícil.
- Já que provamos que o Problema da Mochila $\in$ NP e que o Problema da Mochila $\in$ NP-difícil, automaticamente provamos que ele também $\in$ NP-completo.