

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Pesquisa Operacional 2012-2
Trabalho Prático - Parte 1

Corte de Barras de Aço

Considere uma empresa que realiza corte de barras de aço. Esta empresa realiza a compra de grandes barras de aço, todas com o mesmo comprimento W e vende barras de comprimentos w_i distintos. Assuma $w_i \leq W$ e w_i inteiro para cada $i = \{1, 2, \dots, m\}$. Em particular, b_i barras de comprimento w_i devem ser produzidas.

Uma barra de tamanho W pode ser cortada de diversos modos. Cada modo de cortá-la é chamado um padrão de corte. Por exemplo, uma barra com comprimento 70 pode ser cortada pelo seguinte padrão de corte: 3 barras de tamanho 20, 1 barra de tamanho 6 com uma sobra igual a 4.

Em geral, um padrão de corte j pode ser representado por um vetor coluna A_j , onde a i -ésima entrada a_{ij} indica a quantidade de barras de tamanho w_i produzidas pelo padrão j . Para que um determinado padrão seja viável, suas componentes devem ser inteiras não-negativas e a seguinte restrição deve ser atendida:

$$\sum_{i=1}^m a_{ij} w_i \leq W \quad (1)$$

Seja n o número de todos os padrões viáveis e considere a matriz $A_{m \times n}$ formada por colunas A_j , $j = 1, \dots, n$.

A meta da empresa é atender a demanda de seus clientes comprando uma quantidade mínima de barras de tamanho W . Seja x_j a quantidade de barras cortadas pelo padrão de corte j . Logo, deve-se resolver o seguinte problema:

$$\min \sum_{j=1}^n x_j \quad (2)$$

$$\text{sujeito a } \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \quad (3)$$

$$x_j \geq 0 \text{ e inteiro.} \quad j = 1, \dots, n. \quad (4)$$

A princípio, considere a versão não-inteira deste problema. Ou seja:

$$\min \sum_{j=1}^n x_j \quad (5)$$

$$\text{suj. a } \sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, \dots, m, \quad (6)$$

$$x_j \geq 0 \quad j = 1, \dots, n. \quad (7)$$

Como o número de padrões de corte (n) pode ser muito grande, é impraticável o armazenamento da matriz A completa. Uma forma de resolver o problema (5-7) é utilizar o método simplex revisado, que armazena apenas a matriz base, e a técnica de geração de colunas que permite encontrar as colunas de A somente quando são necessárias. O algoritmo para resolvê-lo, portanto, fica do seguinte modo:

1. Construa uma solução viável básica inicial: Para $j = 1, \dots, m$, faça o padrão de corte j conter apenas 1 barra com tamanho w_j . Desse modo, obtém-se uma solução básica viável para o modelo de programação linear (5-7) com apenas m padrões de corte.
2. Em cada iteração do método simplex revisado, deve-se encontrar a coluna não-básica de menor custo reduzido $\bar{c}_j = 1 - p'A_j$ onde $p' = c_B B^{-1}$. Ao invés de avaliar todas as colunas não básicas, pode-se resolver o seguinte problema de precificação:

$$\max \sum_{i=1}^m p_i a_i \quad (8)$$

$$\text{suj. a } \sum_{i=1}^m w_i a_i \leq W \quad (9)$$

$$a_i \geq 0 \text{ e inteiro.} \quad i = 1, \dots, m. \quad (10)$$

Com base na restrição (1), este problema de precificação tem como objetivo, encontrar um padrão de corte viável com menor custo reduzido.

3. Se a solução ótima do problema de precificação for menor ou igual a 1, todos os custos reduzidos são maiores ou iguais a zero e, portanto, temos uma solução ótima. Caso contrário, a coluna A_j correspondente à solução de (8) entra na base.

O problema de precificação apresentado acima é o clássico problema da mochila inteira, onde p_i é o valor e w_i o peso do i -ésimo item. O objetivo é preencher uma mochila com valor máximo dos itens sem exceder sua capacidade W . Por programação dinâmica, este problema pode ser resolvido como segue. Seja \mathbb{O} o conjunto dos itens que formam a solução ótima.

- Se o item $n \notin \mathbb{O}$ então $otimo(n, W) = otimo(n - 1, W)$.
- Se $n \in \mathbb{O}$ então $otimo(n, W) = p_n + otimo(n - 1, W - w_n)$,

Nesta recursividade, é necessário resolver o problema $otimo(n, w)$ onde w pode ser menor que algum w_i . Neste caso, temos $otimo(n, w) = otimo(n - 1, w)$. Caso contrário, obtemos a solução ótima dentre as duas opções acima. Em resumo, temos a seguinte recorrência:

Se $w < w_i$ então $otimo(i, w) = otimo(i - 1, w)$
 senão $otimo(i, w) = \max \{ otimo(i - 1, w), p_i + otimo(i - 1, w - w_i) \}$

Atividades:

1. Desenvolva um programa em C/C++ para resolver o problema (5-7).
2. No final, mostre as barras que foram adquiridas com seus respectivos padrões de corte, no seguinte formato:

```
PADROES DE CORTE:
BARRA 1: [ a_{11}, a_{21}, ..., a_{m1} ] QDE_COMPRADA: x_1
BARRA 2: [ a_{12}, a_{22}, ..., a_{m2} ] QDE_COMPRADA: x_2
...
BARRA m: [ a_{1m}, a_{2m}, ..., a_{mm} ] QDE_COMPRADA: x_m
```