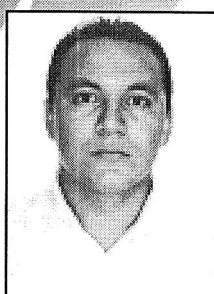


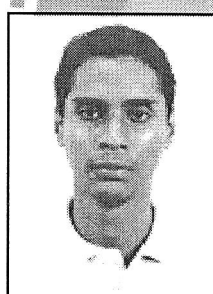
DESENVOLVIMENTO DE UMA FERRAMENTA COMPUTACIONAL PARA A MODELAGEM MATEMÁTICA DO PROBLEMA DE CORTES DE BARRAS DE AÇO PARA A INDÚSTRIA DA CONSTRUÇÃO CIVIL



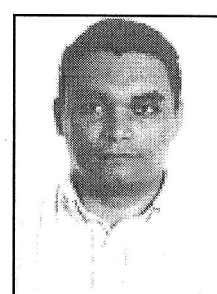
(1)



(2)



(3)



(4)

Yvelyne Bianca Iunes Santos (1)

Professora do Curso de Eng. de Produção da UEPA.

Léony Luis Lopes Negrão (2)

Graduado em Eng. de Produção pela Universidade do Estado do Pará (UEPA)

Bruno Rafael Dias de Lucena (3)

Graduado em Eng. de Produção pela Universidade do Estado do Pará (UEPA)

Denilson Ricardo de Lucena Nunes (4)

Mestre em Eng. Civil pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

DESENVOLVIMENTO DE UMA FERRAMENTA COMPUTACIONAL PARA A MODELAGEM MATEMÁTICA DO PROBLEMA DE CORTES DE BARRAS DE AÇO PARA A INDÚSTRIA DA CONSTRUÇÃO CIVIL

RESUMO:

A formulação do problema de corte de barras de aço para o uso na indústria da construção civil é o primeiro passo na solução de um problema maior: a otimização do uso destas barras. Este artigo apresenta um algoritmo para ser implementado computacionalmente com o objetivo de formular o problema de corte de modo que este possa ser solucionado por qualquer software resolvidor de programação linear.

1. INTRODUÇÃO E DESCRIÇÃO DO PROBLEMA

O desenvolvimento da Programação Matemática em geral e em especial da programação linear tem auxiliado em larga escala a Engenharia Civil. Segundo o Instituto McKinsey (1999), a indústria da construção civil emprega no Brasil 6,1% dos trabalhadores formais do país, o que denota a importância de um fortalecimento do setor.

Na construção civil brasileira o nível de desperdício de materiais e mão-de-obra é alto se comparado a outros setores industriais de igual importância, como a indústria metalúrgica ou outras indústrias de transformação. Isto se deve amplamente ao nível técnico do corpo operacional de base destas indústrias (operários) e a dificuldade dos níveis de gerência em definir suas tarefas no caso do setor da construção. Meseguer (1991),

comenta ainda que é difícil manter a constância nas características dos materiais e dos processos, o que ratifica a afirmação acima citada.

Na atividade de corte de barras de aço para a montagem de armaduras para estruturas de concreto armado, isto não é diferente. Muitas empresas de cálculo estrutural utilizam softwares para o dimensionamento de tais estruturas. Estes softwares geram tabelas com as dimensões das barras necessárias a execução das estruturas de concreto armado dimensionadas. Esta planilha de necessidades é geralmente enviada ao responsável pelo corte, que, empiricamente, corta as barras de aço para obter as peças desejadas.

É importante perceber, que nestes casos apesar da perda considerada de 12% (GOLDMAN, 1997), a intuição dos operários do corte não possui a precisão e a confiabilidade de manter este

nível máximo de perda, ocasionando assim custos excessivos proporcionais à dimensão da obra em questão.

O problema poderá ser resolvido através das ferramentas matemáticas da programação linear inteira, entretanto, isto implica elevados custos com pesquisa e desenvolvimento de projetos de corte.

A solução computacional é então ainda a mais visada, sabe-se que o problema de corte unidimensional é do tipo NP-difícil (CINTRA & WAKABAIASHI, 1998), o que quer dizer que para um elevado número de dados será exigida uma alta capacidade computacional e, Bazaraa (1943) afirma que não é possível definir computacionalmente a enumeração de cada possibilidade de corte antecipadamente.

Para fugir deste impasse considera-se importante que seja definido um valor limitante para diferentes peças para uma obra civil. Já que mesmo para construções de médio e grande porte como por exemplo prédios, a execução da atividade de corte é feita por partes.

Esta limitação torna possível a modelagem computacional de um sistema que formule o problema matemático que pode ser resolvido em outros softwares matemáticos como o LINDO®, LINGO®, além de outros.

2. CONSIDERAÇÕES SOBRE PROGRAMAÇÃO LINEAR INTEIRA

Hillier & Lieberman (1988) resumem as fases usuais para um estudo de pesquisa operacional:

1. Formulação do problema;
2. Construção de um modelo matemático para o problema em estudo;

3. Derivação de uma solução a partir do modelo;
4. Teste do modelo e da solução derivada dele;
5. Estabelecimento de controles sobre a situação;
6. Pôr a solução para funcionar: implementação.

Este artigo trabalha para atingir com satisfação a segunda fase do estudo do problema de corte de barras.

No problema de cortes de barras todas as soluções encontradas serão números inteiros o que classifica o problema como de *programação linear inteira pura* (WINSTON, 2000).

O problema é modelado como um problema de programação linear sendo que as variáveis que denotam a quantidade de cada tipo de corte que deverá ser executada serão restritas a valores inteiros.

3. MODELAGEM DO PROBLEMA

O problema está suficientemente definido: As empresas de construção civil precisam de um plano de corte de barras a ser seguido por seus funcionários do setor de modo que a perda seja mínima.

3.1. Objetivo Geral

Dentre as possibilidades de corte para determinada planilha de necessidades definir quais serão executadas e em que quantidade de modo a minimizar as perdas nesta atividade.

3.2. Objetivos Específicos

- Desenvolver sistema computacional de modo a formular o problema matemático;
- Processar estas formulações no software LINDO® para gerar os planos de corte e testar a viabilidade do modelo.

3.3. Restrições

O problema terá como restrições o número máximo de peças de cada tamanho que deverão ser cortadas.

3.4. Apresentação Matemática do Problema

Segundo Bazaraa (1943), a notação matricial do seguinte problema de corte

$$\text{Min } \mathbf{s} \times \mathbf{x}$$

S.a

$$\mathbf{A} \times \mathbf{x} = \mathbf{b}$$

$$x_i \geq 0 \quad i = 1, 2, \dots, n$$

$$x_i \text{ inteiro} \quad i = 1, 2, \dots, n$$

descreve, com clareza, que a dificuldade deste problema é que o número de possíveis modelos de corte n é muito grande, e também não é computacionalmente praticável enumerar cada teste de padrões de corte e seu vetor coluna $\mathbf{a}_j$ antecipadamente.

O presente trabalho apresenta uma solução viável para que os modelos de corte sejam obtidos computacionalmente, fica claro, então, que desejamos encontrar a matriz $\mathbf{a}_{ij}$ que representa a quantidade de comprimentos i no tipo de corte j , o vetor $\mathbf{b}$ já é conhecido pois representa a quantidade de comprimentos i demandados.

3.4. Definição de Variáveis

Serão declaradas variáveis de entrada para serem utilizadas no modelo computacional:

$Npos$ – Número de posições de corte solicitadas pela planilha de necessidades.

Pos_i – Vetor posição de corte i . Alguns softwares de cálculo estrutural chamam de posição de corte às diferentes dimensões de barra que

deverão ser obtidas, esta denominação será utilizada no algoritmo desenvolvido.

b_i – Vetor quantidade demandada posição de corte i .

T – Dimensão padrão das barras que serão cortadas em metro.

As demais variáveis são derivadas destas e são:

$QPos_Corte_{ij}$ – Matriz que representa a quantidade de posições i encontradas em cada corte do tipo j .

Util – dimensão linear utilizada pelo tipo de corte.

$Sobra_j$ – dimensão linear da sobra para o corte j em metro.

4. METODOLOGIA DE RESOLUÇÃO

Primeiramente, deve-se observar que os cortes válidos serão aqueles que atendem as seguintes inequações matriciais.

$$\sum_{j=1}^n (QPos_Corte_{ij} \times Pos_i) > 0$$

$$\sum_{j=1}^n (QPos_Corte_{ij} \times Pos_i) \leq T,$$

$$i = 1, 2, 3, 4, \dots, NPos$$

Tais que:

$Npos$ – é o número de posições de corte solicitado pelo usuário;

n – é o número de possibilidades de corte para uma barra de aço;

Para cada conjunto de valores que satisfizerem as condições acima deverão ser armazenadas as variáveis: $Sobra_j$ e $QPos_Corte_{ij}$. Sendo que

a variável $Sobra_j$ receberá a seguinte expressão:

$$Sobra_j = T - \sum_{i=1}^n (QPos_Corte_{ij} \times Pos_i),$$

$i = 1, 2, 3, 4, \dots, NPos.$

Utilizando-se o conteúdo das variáveis é possível montar a função objetivo de minimização das sobras e as restrições.

5. DESENVOLVIMENTO DO ALGORITMO COMPUTACIONAL

O algoritmo deverá ser capaz de armazenar todas as possibilidades de corte com suas respectivas sobras, montar a função objetivo e as restrições e armazená-las em um arquivo de texto para ser exportado a um programa de resolução de problemas de programação linear, como por exemplo o LINDO®.

5.1. Módulo de Recepção das Variáveis de Entrada

Algoritmo passo-a-passo para a recepção de variáveis:

Passo 1. Receber a dimensão padrão da barra de aço em T ;

Passo 2. Receber o número de posições de corte solicitado pela planilha de necessidades em $NPos$;

Passo 3. Se $NPos \leq 0$, volte para o passo 1;

Passo 4. Receber a dimensão da posição i em Pos_i ;

Passo 5. Se $Pos_i \leq 0$ ou $Pos_i > T$, voltar para o passo 4;

Passo 6. Receber a demanda da posição i em d_i ;

Passo 7. Se $d_i \leq 0$ voltar para o passo 6;

Passo 8. Incrementar i recebendo $+ 1$;

Passo 9. Se $i > Npos$ abandonar módulo, senão voltar para o passo 4.

5.2. Módulo Principal

Algoritmo passo-a-passo para a determinação das variáveis $Qpos_Corte_{ij}$, e $Sobra_j$:

Passo 1. Defina um vetor $Teste$ de $Npos$ linhas, com todos os valores iguais a 0;

Passo 2. Faça os passos de 3 a 5 variando cada elemento $Teste_i$ de 0 até o número que representa

a parte inteira da razão $\left(\frac{T}{Pos_i} \right)$, começando pelo

índices i que vale $Npos$ e retrocedendo a cada variação de um nível mais interno, ou seja, até Pos_1 ;

Passo 3. Defina $Util = \sum_{i=1}^{Npos} Teste_i^T \times Pos_i$

Passo 4. Se $0 < Util \leq T$ e $Teste_i \neq Teste_{i-1}$, para todo i , então vá para passo 5, senão vá para passo 7;

Passo 5. $Qpos_Corte_{ij}$ recebe $Teste_i$ para todo i e $Sobra_j$ recebe $(T - Util)$;

Passo 6. Incremente J recebendo $+ 1$;

Passo 7. Se todas as combinações foram testadas, vá para o passo 8, senão vá para o passo 3;

Passo 8. Finalizar módulo.

Ao final deste módulo a matriz $Qpos_Corte_{ij}$ e o vetor $Sobra_j$ estarão completos sendo suficientes para estabelecer a função objetivo e as restrições do problema de corte.

6. APLICAÇÃO DA METODOLOGIA

Apresentar um caso real textualmente resultaria em uma complexidade pouco tratável. O exemplo aqui apresentado é uma abstração simplificadora, retirada de PRADO(1999).

Suponha-se que são possuídas barras de 6m de comprimento que devem ser convenientemente cortadas para obtermos barras menores, nos seguintes tamanhos.

- 50 barras de 2m
- 60 barras de 3m
- 90 barras de 4m

A partir desta planilha de necessidades, o objetivo é formular matematicamente este trabalho utilizando o algoritmo apresentado:

É óbvio que o módulo de recepção dos dados de entrada será satisfeito quando os dados acima forem corretamente inseridos. Os dados serão assim armazenados:

Variável	Conteúdo	Variável	Conteúdo
$NPos$	3	T	6
Pos_1	2	b_1	50
Pos_2	3	b_2	60
Pos_3	4	b_3	90

Quadro 1 – Inicialização das Variáveis de Entrada

O módulo principal é acionado, então, e o vetor *Teste* é definido com três linhas e todos os valores iguais a zero.

$$Teste = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A seguir, determina-se o limite de variação para cada linha do vetor teste:

Elementos	Razão $\left(\frac{T}{Pos_i} \right)$	Variação até:
$Teste_1$	$\left(\frac{6}{2} \right) = 3,0$	3
$Teste_2$	$\left(\frac{6}{3} \right) = 2,0$	2
$Teste_3$	$\left(\frac{6}{4} \right) = 1,5$	1

Quadro 2 – Limites de variação para cada elemento $Teste_i$

Os vetores colunas testados para compor a matriz *Qpos_Corte* serão todos as intermediárias entre os limites de variação de *Teste*, incluindo estes limites:

$$Teste = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Para cada um dos 24 vetores acima a variável *Util* que receberá a expressão VO assumirá os seguintes valores:

$$Util = [0|4|3|7|6|1|0|2|6|5|9|7|1|1|4|8|7|1|1|0|1|4|6|1|0|9|1|3|1|2|1|6]$$

Todos os vetores que geraram na variável *Util* um valor que satisfaçam a condição $0 < Util$ VO serão armazenados na matriz *Qpos_Corte* e o vetor linha *Sobra* receberá o valor da expressão $(T - Util)$ para cada *j*.

		Possibilidades de Corte							
	$i \backslash j$	1	2	3	4	5	6	7	8
Posição	1	0	0	0	1	1	1	2	3
Posição	2	0	1	2	0	0	1	0	0
Posição	3	1	0	0	0	1	0	0	0
Sobra	-	2	3	0	4	0	1	2	0

Quadro 3 – Matriz *Qpos_Corte* e vetor *Sobra*.

O problema está, então, convenientemente formulado da seguinte forma:

$$\text{Min } \textit{Sobra} \times \mathbf{x}$$

S.a

$$\textit{Qpos_Corte} \times \mathbf{x} = \mathbf{b}$$

$$x_i \geq 0 \quad i = 1, 2, \dots, n$$

$$x_i \text{ inteiro} \quad i = 1, 2, \dots, n$$

Implementando-se os valores obtidos no algoritmo, têm-se:

$$\text{Min } Z = [2 \ 3 \ 0 \ 4 \ 0 \ 1 \ 2 \ 0] \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

S.a.

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 3 \\ 0 & 1 & 2 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 50 \\ 60 \\ 90 \end{bmatrix}$$

$$x_i \geq 0 \quad i = 1, 2, \dots, 8$$

$$x_i \text{ inteiro} \quad i = 1, 2, \dots, 8$$

Esta formulação permite que o problema seja solucionado por qualquer software resolvidor de programação linear.

7. DISCUSSÃO SOBRE A METODOLOGIA EMPREGADA

Percebe-se que o número de iterações realizadas será tão maior quanto maior for o tamanho da barra e quanto menores forem os tamanhos das peças que compõem a planilha de necessidades.

O tempo de processamento do algoritmo é diretamente proporcional ao número de iterações, logo, são importantes, e vêm sendo desenvolvidas, pesquisas para a diminuição deste número sem, no entanto, comprometer a integridade da resposta.

A divisão do problema em problemas menores reduz o potencial de otimização entretanto ganha tempo e pode satisfazer as necessidades das construtoras.

8. CONCLUSÕES

Este trabalho apresentou um algoritmo para a formulação do problema de corte para barras de aço na construção civil. Percebe-se que esta metodologia permite formular matematicamente rapidamente o referido problema, o que seria inviável sem o uso da ferramenta computacional.

É necessário entretanto, que continuem sendo desenvolvidas pesquisas que permitam diminuir o número de iterações realizadas e aumentar a velocidade de execução para quaisquer casos.

A criação de um módulo para a resolução do problema de corte é de essencial importância para facilitar a tarefa do usuário e não depender das limitações dos softwares resolvidores existentes.

REFERÊNCIAS

- BAZARAA, M. S. *Linear Programming and Network Flows*, John Wiley & Sons. 3ª Edição. New York, 1943.
- CINTRA, G.; WAKABAYASHI, Y. Um algoritmo híbrido para o problema de corte unidimensional. *Anais: XXX Simpósio Brasileiro de Pesquisa Operacional*. IME, São Paulo: 1998.
- GOLDMAN, P. *Introdução ao planejamento e controle de custos na construção civil brasileira*. 3. ed. Pini, São Paulo: 1997.
- HILLIER, F. S.; LIEBERMAN, G. J. *Introdução a Pesquisa Operacional*. Ed. Campus. Rio de Janeiro, 1988.
- INSTITUTO MCKINSEY. *Produtividade no Brasil: a chave do desenvolvimento acelerado*. Campus, Rio de Janeiro: 1999.
- MESEGUER, A. G. *Controle e garantia da qualidade na construção*. Traduzido por Antonio Carmona Filho, Paulo Roberto do Lago Helene e Roberto José Falcão Bauer. Sinduscon-SP, São Paulo: 1991.
- PRADO, D.S. (1999) - *Programação Linear*. Editora de Desenvolvimento Gerencial. 3ª Edição. Belo Horizonte.
- WINSTON, W. L. *Operations Research: applications and algorithms*. 3. ed. IPT, California, 2000.

ANEXO

Foi criado um software na linguagem de programação PASCAL, utilizando-se o algoritmo apresentado. Abaixo apresentam-se algumas telas deste software cujos dados estão relacionados ao exemplo resolvido no presente trabalho:

```
Dados Fornecidos:
Dimensao da Barra Padrao: 6.00
Numero de Posicoes de Corte: 3
Dimensao da Posicao 1: 2.00
Dimensao da Posicao 2: 3.00
Dimensao da Posicao 3: 4.00
```

Figura 1 – Tela de recepção de dados de entrada

```
Tipo 1 Pos1 = 0 Pos2 = 0 Pos3 = 1 Sobra: 2.00
Tipo 2 Pos1 = 0 Pos2 = 1 Pos3 = 0 Sobra: 3.00
Tipo 3 Pos1 = 0 Pos2 = 2 Pos3 = 0 Sobra: 0.00
Tipo 4 Pos1 = 1 Pos2 = 0 Pos3 = 0 Sobra: 4.00
Tipo 5 Pos1 = 1 Pos2 = 0 Pos3 = 1 Sobra: 0.00
Tipo 6 Pos1 = 1 Pos2 = 1 Pos3 = 0 Sobra: 1.00
Tipo 7 Pos1 = 2 Pos2 = 0 Pos3 = 0 Sobra: 2.00
Tipo 8 Pos1 = 3 Pos2 = 0 Pos3 = 0 Sobra: 0.00
Fim do Arquivo
```

Figura 2 – Tela de apresentação dos dados de saída