#### ***Column Generation***

#### **Uma Aplicação Bem Sucedida**

##### Arnaldo Ferreira Braga Neto

[arnaldoneto@sol.com.br](mailto:arnaldoneto@sol.com.br)

##### João Pedro Coutinho Castelo Branco

[pbranco@amcham.com.br](mailto:pbranco@amcham.com.br)

##### Hugo Tsugunobu Yoshida Yoshizaki

Departamento de Engenharia de Produção , Escola Politécnica , USP

###### *ABSTRACT*

*This work describes the effort to build a decision support tool to make systematic a unidimensional bar cutting process. This tool was based on a mathematical method called Column Generation. Modifications were made on the method to make it perform in an adequate way, lowering process losses and obtaining significant improvement in the raw material stock.*

###### Pesquisa Operacional

***Key words:*** *Operational Research, Column Generation; Unidimensional Cut Optimization.*

# Introdução

Este trabalho foi realizado numa indústria de equipamentos eletrônicos e de medição instalada na região metropolitana de São Paulo. O objetivo principal foi obter um método matemático e, consequentemente, uma ferramenta computacional para otimização de um dos processos de fabricação que utiliza uma matéria prima cara e que apresentava alto índice de refugo – o corte de perfis.

Assim, este artigo se dedica a descrever a experiência bem sucedida da utilização do *Método de Geração de Colunas* para a solução de um problema real de corte unidimensional de perfis de tamanho fixo. Vale ressaltar que a solução aqui descrita corresponde ao caso mais complexo encontrado na empresa. Um outro caso semelhante também foi resolvido com o método de geração de colunas, com menos condições de contorno mas com resultados igualmente satisfatórios.

Além disso, foi feita uma comparação com o método *Branch&Bound*, cujos resultados foram bastante inferiores.

Este projeto foi desenvolvido como trabalho do curso de Pesquisa Operacional em nível de graduação pelos dois primeiros autores, orientados pelo terceiro, no ano de 1997.

# Descrição do Problema

O processo de corte em questão é uma etapa na fabricação de uma família de instrumentos de medição de deslocamento linear de alta precisão, aqui denominados como réguas. Para a fabricação dessas réguas, são importados perfis de tamanho fixo que são cortados em pedaços menores, em função da necessidade do cliente. Posteriormente passam por outras etapas até chegar ao produto acabado.

Há uma série de tipos de réguas que passam por essa etapa de corte, e para cada tipo deve haver uma ferramenta diferente em função das diferentes condições de contorno. No caso aqui descrito tem-se um perfil de tamanho 1345 mm a partir do que se devem cortar réguas nos tamanhos a seguir (em mm):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 175 | 225 | 275 | 325 | 375 | 425 | 575 | 525 | 575 | 625 | 675 | 725 | 775 | 825 | 875 | 925 | 1025 | 1125 | 1245 | 1345 |

No entanto, há nesse caso perdas relacionadas ao projeto do produto. Ao cortar mais de uma régua por perfil, deve-se eliminar 25mm entre uma régua e outra. Deve-se ainda cortar 10 mm de cada ponta do perfil de 1345mm (Figura 1), exceção feita às duas maiores réguas (1245 e 1345).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Régua 1 |  | Régua 2 |  | Régua 3 |  |  |

10 mm

25 mm

25 mm

10 mm

|  |  |
| --- | --- |
|  | Perda de Projeto |
|  | Perda de Processo |

**Figura 1. Exemplo do corte de três réguas num perfil de 1345mm**

Dessa forma duas conclusões são imediatas:

* as duas maiores réguas (1245 e 1345) não oferecem oportunidade de otimização (pois 1245+25+175=1445 > 1345);
* podemos considerar, para efeito de modelagem, o perfil como tendo comprimento de 1325mm; entretanto, ao computarmos o aproveitamento, devemos considerar esses 20mm como perda de projeto.

Anteriormente, o mapa de corte era feito pelo mesmo operador que corta as réguas, contando apenas com uma calculadora simples e com sua experiência. Para efeito de comparação, pudemos contar com dados do estoque gerado pelo processo original para os últimos 6 meses de 1997 :



No final de Junho, havia 9,950m de estoque cortado que eram compostos por 11 réguas de 175mm, 10 réguas de 225mm e 21 réguas de 275mm.

Pode-se notar um claro problema de acúmulo de peças cortadas no estoque. Isso ocorre em função do excesso de réguas de pequeno comprimento que, periodicamente, são inutilizadas por falta de demanda. Além disso, quando se comparam o total de comprimento vendido com o total de comprimento utilizado tem-se:

|  |  |
| --- | --- |
| Total de material utilizado: | 357,005 m |
| Comprimento vendido: | 319,070 m |
| Perdas (%): | 10,63 % |

Outro dado importante é o histórico das vendas desse tipo de régua. A Figura 3 mostra as vendas por tamanho de régua nos anos de 1996 e1997:



# Escolha do Método

Primeiramente foi utilizado o Método *Branch&Bound* através da seguinte abordagem: o modelo deveria diminuir ao máximo as perdas do corte. A melhor forma, então, de se evitar perdas é concentrar as “perdas” em um único perfil, possibilitando que essa “perda concentrada” seja grande o suficiente para que possa ser utilizada posteriormente como uma régua. Assim, devemos maximizar a diferença entre a sobra de um dos perfis e as sobras dos outros perfis restantes:

Porém, o grande número de variáveis torna o método impraticável ao caso em questão, pois o tempo de processamento aumenta enormemente. Para se ter uma idéia, estimou-se que um Pentium® 100MHz demoraria anos de processamento ininterrupto para chegar à solução ótima. Dessa forma, para sua utilização, é necessário que se interrompa a resolução e que se trabalhe com uma solução que não é a ótima.

Outro motivo é que para cada lote a ser cortado, o modelo teria de ser codificado de forma diferente, ou seja, o operador teria que reprogramar o computador para determinar o melhor corte, o que torna todo o processo pouco ágil e suscetível a erros.

Além disso, para o melhor funcionamento do modelo é necessário que se acumule um volume considerável de réguas a serem cortadas, o que nem sempre é possível em virtude dos prazos apertados de entrega.

Dessa forma, foi empregado o método de Geração de Colunas, que parte de uma tabela de padrões de cortes possíveis ou desejáveis pré-determinada. Cada linha dessa tabela funciona como se fosse um perfil e os números nela contidos indicam a quantidade de cada tipo de régua a ser cortada. Posteriormente, através da resolução de um Problema de Programação Linear Inteira, selecionam-se quais os cortes que devem ser feitos e a quantidade que se deve fazer de cada um.

Portanto, mudou-se completamente a abordagem do problema, pois o fato de gerar uma tabela inflexível de cortes gera a seguinte mentalidade: “É melhor estocar que perder”, uma vez que nem todas as réguas cortadas serão vendidas. Assim, as adaptações feitas no modelo de Geração de Colunas fazem exatamente com que se evite optar por cortes que armazenem as réguas de menor demanda e por cortes que aumentem o estoque cortado já existente.

# Desenvolvimento do trabalho

O primeiro passo foi gerar a tabela de padrões de cortes no seguinte formato:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | R1 | R2 | R3 | . . . |
| X1 | n11 | n12 | n13 | . . . |
| X2 | n21 | n22 | n23 | . . . |
| X3 | n31 | n32 | n33 | . . . |
| .  .  . | .  .  . | .  .  . | .  .  . |  |

Onde:

Xi = padrão de corte i, ou seja, como devem ser cortadas as réguas num perfil i ;

Rj = indica a coluna relativa à régua do tipo i ;

nij = é o número de vezes que a régua Rj é cortada no padrão Xi .

Essa tabela foi obtida através de um algoritmo que deveria obedecer às seguintes restrições:

* listar somente os padrões de cortes com aproveitamento maior que 90% (considerando já as perdas de projeto e processo);
* não repetir padrões de corte, ou seja, garantir a unicidade de cada padrão.

Esse algoritmo foi codificado em C (vide anexo) e o programa em C gerou um arquivo \*.TXT que foi exportado para o Microsoft Excel e nele foram feitas modificações para que se criasse uma tabela mais apropriada para a utilização da planilha de cálculos.

Como forma de melhorar o desempenho do modelo, ainda foram retiradas algumas linhas da tabela de padrões. Como é conhecida a tendência de que se acumulem réguas de 175mm (comprimento cortado) e de 225mm (idem), alguns padrões que contém essas réguas foram eliminados. Somente um padrão com o comprimento 175mm e quatro com o de 225mm foram mantidos.

Após a obtenção da tabela de padrões de cortes, temos que o objetivo final é *minimizar a quantidade de perfis utilizada*. As variáveis de decisão são, portanto, os Xi.

min 

Onde:

p = número de padrões possíveis de corte.

Como restrições temos ainda:

* número de réguas cortadas de cada tipo deve ser maior ou igual ao número de réguas pedidas de cada tipo. Assim para cada j , ou seja, para cada tamanho de régua

.

Onde:

NRj = número de réguas do tipo j pedidas.

* número de padrões a serem utilizados devem ser inteiros e maiores ou iguais a zero:

 ; 

Entretanto, com o objetivo de evitar um estoque excessivamente grande de peças cortadas e também como forma de utilizar cortes com perdas menores, foram feitas as seguintes modificações:

* a cada padrão de corte é associada uma penalidade, na função objetivo, que leva em consideração a quantidade vendida de cada régua integrante daquele corte (pois réguas de pouca saída não podem ser estocadas) e as perdas percentuais que aquele corte gera. O cálculo dessa penalidade é feito da seguinte forma:



Onde:

r = número total de tipos de réguas;

C = uma constante para aumentar o peso dessa penalidade. O valor 10 tem-se mostrado apropriado;

Perdas(i) = Perdas percentuais do padrão de corte i;

ΣNRj = Total de réguas do tipo j vendidas entre Jan-1996 e Dez-1997.

* outra penalidade é associada a cada padrão de corte de tal forma que os estoques existentes de réguas cortadas influenciem na escolha do padrão. Assim, se houver uma certa quantidade de réguas já cortadas de um certo tamanho, o modelo deve evitar o acúmulo desta no almoxarifado. O cálculo desta penalidade é feito da seguinte forma:



Onde:

Ej = número de réguas estocadas do tipo j.

Obs: na grande maioria das vezes, a Penalidade2 é uma ordem de grandeza maior que a Penalidade1.

A Função Objetivo modificada fica:

min z = 

A partir dessas modificações, montou-se uma Planilha no Microsoft Excel. Utilizou-se então o programa *What’s Best®* para a resolução do problema.

# Resultados

Para testar o modelo, utilizamos os dados dos cortes nos últimos 6 meses de 1997 (os cortes podem ser considerados como sendo feitos quinzenalmente). Com isso, se o modelo tivesse sido utilizado, obteríamos o seguinte movimento no estoque:



No final de Dezembro, se o método proposto tivesse sido utilizado, haveria uma redução de 5,04% no estoque final de peças cortadas (de 40,305m para 38,275m). Além disso, ainda haveria um aumento de 1.700% na disponibilidade de perfis não cortados (de 1,345m para 24,210m).

Outro resultado importante é o da diminuição de perdas dado a seguir:

|  |  |
| --- | --- |
| Total de material utilizado: | 334,825 m |
| Comprimento vendido: | 319,070 m |
| Perdas (%): | 5,46 (%) |

# Conclusão

Pudemos concluir que a utilização do método de Geração de Colunas reduz as perdas em cerca de 5%. Entretanto, o resultado mais significativo foi a diminuição no acúmulo no estoque de peças cortadas e o aumento da disponibilidade de perfis não cortados.

Pode parecer à primeira vista que fixar um número limitado de padrões a serem cortados pode tirar a flexibilidade da ferramenta que utiliza o método de Geração de Colunas e, consequentemente, a flexibilidade do estoque de perfis. Entretanto, podemos perceber pelos resultados que isso não ocorre, pois a função objetivo modificada persegue dois objetivos: minimizar perdas e minimizar o estoque de peças cortadas.

Apesar das melhorias, podemos ainda esperar uma tendência de aumento do estoque que o modelo não resolve, mas minimiza. Isso se deve principalmente ao fato de não haver demanda suficiente para as três menores réguas cortadas, acumulando essas peças cortadas no estoque. Nem sempre é possível fazer o corte de réguas grandes sem que sobrem réguas pequenas. A solução seria, portanto, aumentar o tamanho do perfil.

Além disso, o modelo se mostrou robusto para a hipótese dos cortes não serem quinzenais. Testes nesse sentido mostraram resultados bastante similares.

# Bibliografia

[1] WINSTON, Wayne L. ; *Introduction to Mathematical Programming* . Duxbury Press , 1995

[2] Morábito, R., Arenales, M.; *Staged and Constrained Two-Dimensional Cutting Problems: a new approach*. Notas do ICMSC, n.º 126, 1992.

[3] Morábito, R., Arenales, M; A *Theory of the Generation of Cutting and Packing Patterns – Part I: Foundations*. Notas do ICMSC, n.º 31, 1996.

[4] Wakabayashi, Y., Ferreira, C. E., Miyazawa, F. K.; *Anais da 1ª Oficina nacional de Problemas de Corte e Empacotamento*. ICMSC, 1996.

[6] Morábito, R., Arenales, M.; *O Problema de Corte e Empacotamento e Aplicações Industriais – 2ª Oficina Nacional de PCE*. ICMSC, 1997.

[7] CASTELO BRANCO, J. E., BRAGA NETO, A. F., MOREIRA, F.F., ASAKA, F.V., *Projeto de Conclusão do Curso de Pesquisa Operacional,* 1997

# Agradecimentos especiais

Os autores gostariam de dedicar um agradecimento especial à Profa. Yoshiko Wakabayashi do Instituto de Matemática e Estatística da USP

# Anexo

Programa codificado em C para obtenção da tabela de padrões:

#include <stdio.h>

void main(void){

int a,b,c,d,e,f; /\*contadores\*/

int MEDIDAS[19], RODA[6];

int p,q; /\*contadores\*/

float aprov,perfil,soma;

printf("REGUAS DE PERFIL FINO \n");

printf("\n\n");

for(p=0;p<19;p++) scanf("%d", &MEDIDAS[p]); /\*entrada das medidas no vetor MEDIDAS\*/

for(p=0;p<19;p++) printf("%d ",MEDIDAS[p]);

for(p=0;p<6;p++) RODA[p]=0; /\*zeramento do vetor RODA\*/

perfil=1350; /\*perfil=1345+25 , pois foi acrescentado 25+105 no comprimento de cada régua\*/

printf("R1 R2 R3 R4 R5 R6 Aprov\n\n");

for(a=0;a<5;a++){ RODA[0]=MEDIDAS[a]; /\*Loop encadeado que faz a permutação das \*/

for(b=a;b<19;b++){ RODA[1]=MEDIDAS[b]; /\*MEDIDAS. \*/

for(c=b;c<19;c++){ RODA[2]=MEDIDAS[c];

for(d=c;d<19;d++){ RODA[3]=MEDIDAS[d];

for(e=d;e<19;e++){ RODA[4]=MEDIDAS[e];

for(f=e;f<19;f++){ RODA[5]=MEDIDAS[f];

soma=0; q=0;

for(p=0;p<6;p++){

soma=soma+RODA[p]; /\* Soma as medidas da permutação \*/

if(RODA[p]>0) q++; /\*Conta quantas réguas são cortadas \*/

}

aprov=((soma-q\*25)/1345)\*100; /\*Calcula o aproveitamento percentual\*/

if(soma<=perfil && aprov>90){ /\*Seleciona os cortes com aproveitamento>90% \*/

for(p=0;p<6;p++) printf("%d ",RODA[p]); /\*Imprime uma lista com as réguas \*/

printf("%f\n",aprov); /\*a serem cortadas naquele padarão\*/

}

} } } } } }

printf("\nFIM\tFIM\tFIM\n");

}

A saída desse programa é uma lista em formato \*.TXT da seguinte forma:

R1 R2 R3 R4 R5

0 0 0 0 175 1025 Soma=1225.000000 Aproveitamento=91.078064

0 0 0 0 175 1125 Soma=1325.000000 Aproveitamento=98.513008

0 0 0 0 225 925 Soma=1175.000000 Aproveitamento=87.360596

...