



Resumo

Nesse trabalho implementamos uma modificação na técnica de geração de colunas de Gilmore e Gomory, proposta por Haessler, para um problema de corte unidimensional, com o objetivo de minimizar as perdas nos processos de corte. Este é um problema típico de programação linear aplicado em indústrias têxteis e metalúrgicas, dentre outras, onde se desperdiçam muito material devido a uma escolha não adequada de padrões de corte. Os modelos matemáticos envolvidos são de programação linear inteira e o grande número de variáveis envolvidas dificulta sua resolução. Uma solução para o problema é expressa por um conjunto de padrões de corte e pelas quantidades de produção associadas. Matematicamente, estes padrões de corte são colunas geradas ao se resolver um problema da mochila. Foi feito um estudo do problema de corte unidimensional, do método simplex revisado, do problema da mochila e da técnica de geração de colunas proposta por Gilmore e Gomory. Posteriormente, foi estudada a proposta de Haessler, que faz modificações nos padrões gerados a partir de um problema da mochila canalizado. A implementação computacional foi feita com o uso do pacote computacional AIMMS. Foi feita uma comparação entre a técnica original e a modificação estudada através de vários problemas gerados.

Problema de Corte

O problema de corte utilizado parte da idéia que um produtor tem que atender um pedido de m itens, que serão cortados em pedaços de larguras l_i , em quantidades d_i , $i=1,2,...,m$, de uma barra de comprimento L , com o objetivo considerado de minimizar a perda.
 Na solução do problema temos um conjunto de padrões de corte a_j , $a_j=(a_{1j}, \dots, a_{mj})^T$, onde a_{ij} representa o número de vezes que a largura i aparece no padrão j . Cada padrão de corte é obtido através da resolução de um problema da mochila. Sendo n o número total de padrões viáveis e x_j a quantidade cortada de cada padrão, temos a formulação seguinte:

$$\text{Minimizar } \sum_{j=1}^n x_j$$

$$\text{Sujeito à } \sum_{j=1}^n a_{ij} x_j \geq d_i$$

$$x_j \geq 0 \text{ e inteiro}$$

Algoritmo de Gilmore Gomory

Para descrição do método de Gilmore e Gomory, necessitamos definir algumas variáveis e parâmetros. O método simplex utiliza apenas soluções básicas, ou seja, podemos particionar nossa matriz de padrões de corte A em $A=(B,N)$ onde B são as colunas básicas e N as colunas não básicas de A . Estendemos esta partição à variável x , onde $x=(x_B, x_N)^T$ e aos custos, $c=(c_B, c_N)^T$. Chamamos de π , o vetor das variáveis duais, isto é, $\pi^T = c_B^T B^{-1}$. Assim podemos apresentar o algoritmo do método Simplex como descrito abaixo:

- Determinar uma solução básica factível inicial
- Testar a otimalidade:
 - Determinar a solução básica dual
 - Determinar o menor custo relativo:

$$c_i - \pi^T a_i = \text{mínimo}(c_j - \pi^T a_j), j \in K$$
, onde K é o conjunto dos índices das variáveis não básicas.
 - Se $c_i - \pi^T a_i \geq 0$, estamos na solução ótima, caso contrário o algoritmo continua.
- Determinar a direção simplex: $y = -B^{-1} a_i$
- Teste da razão:

$$x_B = \text{mínimo}(\frac{d_i - \sum_{j \in B} a_{ij} x_j}{y_i}, y_i < 0)$$
. Se $y_i \geq 0$, então o problema não tem solução ótima finita.
- Atualizar a partição básica:
 - entra na base;
 - sai da base;
 - a coluna de e é substituída pela coluna de i ;
 - atualizar os valores de e e da função objetivo.
- Voltar ao passo 2.

A técnica de geração de colunas de Gilmore e Gomory é acoplada ao algoritmo anterior no passo 2.2, ou seja, para verificar a otimalidade da solução básica. Neste ponto, determinamos o menor custo relativo: $c_i - \pi^T a_i = \text{mínimo}(c_j - \pi^T a_j)$. No problema de corte a coluna e é um padrão de corte, devendo, portanto, satisfazer as restrições de capacidade e de integralidade.

Modificações propostas

Essencialmente, Haessler propôs três modificações para o algoritmo de Gilmore e Gomory. Em uma dessas modificações, ele utiliza o parâmetro u_i , que representa um limitante superior. Em seu artigo ele propõe que $u_i = \frac{d_i}{10}$. As modificações são as seguintes:

- A solução inicial, formada por padrões homogêneos, deve ser trocada por $a_{ii} = \text{mínimo}(\lfloor \frac{L}{l_i} \rfloor, u_i)$;
- O problema da mochila associado é:

$$\text{Maximizar } \sum_{i=1}^m \pi_i a_i$$

$$\text{Sujeito à } \sum_{i=1}^m l_i a_i \leq L$$

$$0 \leq a_i \leq u_i \text{ e inteiro}$$

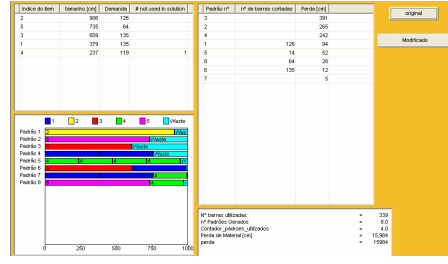
Com a inclusão destes limitantes superiores, temos agora um problema da mochila canalizado.

- Se $l_i > L$ e $\pi_i < \pi_{i+1}$, não desconsideraremos a_i na resolução do problema da mochila. De fato, a_i só será desconsiderado se, além disso, ocorrer também $u_i l_i + l_{i+1} > L$.

A modificação (iii), não foi utilizada, uma vez que para sua implementação, seria necessária a modificação "dentro" do próprio software. Assumindo que há soluções ótimas alternativas (mesma perda), a tendência de Gilmore e Gomory é encontrar uma solução que possua padrões com maiores. Se colocarmos uma restrição sobre o número de vezes que um tamanho pode aparecer em um padrão, isto forçará o aparecimento de padrões com menores, e que poderão ser repetidos mais vezes, o que poderá facilitar o arredondamento da solução.

AIMMS

AIMMS foi o software utilizado para a implementação computacional de nosso modelo. Por se tratar de um software de bom nível para resolução de problemas matemáticos, este pacote computacional é um ótimo recurso tanto para quem objetiva programar os modelos tanto para quem apenas o utiliza como ferramenta para obter uma solução para o problema de seu interesse. Abaixo temos uma interface típica em que o usuário necessita apenas entrar com dados para obtenção de seus resultados.



Dificuldades

Um dos maiores problemas encontrados foi devido ao pouco detalhamento do artigo estudado. O autor, em nenhum momento, explica o motivo da escolha de seu limitante superior, uma das principais modificações. O AIMMS apesar de ser um software disponível no laboratório de matemática da UNICAMP, trata-se de um pacote computacional pago, fazendo com que nosso ambiente de desenvolvimento se restringisse apenas no laboratório.

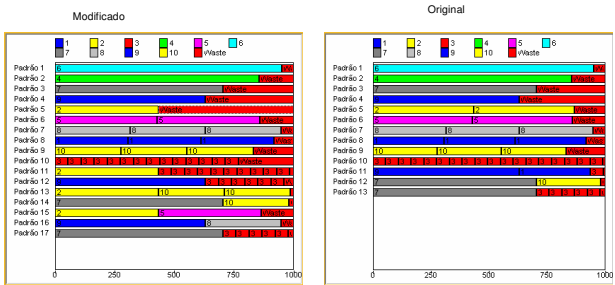
Resultados

Os resultados foram obtidos a partir da geração aleatória de 36 tipos de problemas, assim tentando envolver todas as classes de tamanho de itens e de demanda. Para cada problema resolvemos utilizando a técnica original de Gilmore e Gomory e a modificação proposta por Haessler. Abaixo seguem todos padrões gerados a partir de um problema com barras de 1000 cm onde:

- demanda varia entre (0, 150) itens ;
- tamanho dos itens entre (0,1000) cm ; e
- número de itens m igual a 10

Vale lembrar que para a solução do problema necessitamos apenas de m padrões de corte, sendo que, nem todos os padrões gerados serão utilizados.

Nessa solução o número de padrões gerados foram de 17 e 13 com uma perda de 30198 cm e de 30208 cm para o problema modificado e original, respectivamente.



Conclusões

No projeto apresentado, pudemos mostrar resultados obtidos através de uma técnica comumente utilizada para gerar padrões de corte e uma proposta de modificação publicada anos depois. Os resultados esperados não foram obtidos em todos os casos. Estudando os motivos que levaram a esse efeito, chegamos a algumas conclusões:

Os recursos computacionais que temos em mãos são, teoricamente, superiores aos utilizados na época da edição dos artigos (1980); assim, os tamanhos de problemas que geramos talvez tenham sido muito diferentes dos testados anteriormente, já que o autor, quando propôs sua modificação, não apresentou os dados utilizados dos problemas. Assim, não sabemos nem os tamanhos dos problemas que ele resolveu, nem a relação existente entre o tamanho da barra, as demandas e os comprimentos dos itens. De acordo com o único exemplo apresentado no artigo, conseguimos chegar exatamente ao resultado que foi publicado. Daí concluímos que nossa implementação estaria adequada.

Haessler não explica em nenhuma parte de seu artigo o motivo de fazer esta escolha para o limitante superior apresentado $u_i = \frac{d_i}{10}$. Assim, ficamos sem aparato para saber em qual classe de problemas esse limitante é conveniente. Utilizamos o u_i sugerido, mas isto não se mostrou adequado.

Sabendo de alguma relação entre o valor 10 (utilizado) e os tamanhos das demandas, ou os comprimentos dos itens, talvez pudéssemos fazer uma escolha mais adequada dos limitantes, para as classes de problemas que trabalhamos. Fizemos outras escolhas para u_i , mas não obtivemos melhorias muito significativas para nossos resultados.

Com isso concluímos que a proposta apresentada no artigo estudado é interessante, porém, por ser um artigo muito pequeno (aproximadamente 3 páginas) e pouco detalhado ficamos sem material para poder gerar dados mais significativos. Quanto ao pacote computacional utilizado, foi de bom proveito, já que se trata de um software matemático de alto nível e que contribuiu para um maior entendimento de funcionamento desse tipo compilador.