UM ALGORITMO MEMÉTICO NO PROBLEMA DO CORTE UNIDIMENSIONAL INTEIRO

Angelo Aliano Filho[1] **e Antônio Carlos Moretti**[2]
Universidade Estadual de Campinas - IMECC
E-mail: [1]angeloaliano@hotmail.com [2]moretti@ime.unicamp.br

RESUMO

Este trabalho visou solucionar o PCUI utilizando uma metaheurística híbrida, um Algoritmo Genético seguido seguida de uma Busca Local. Juntamente com o método desenvolvido, duas herísticas foram desenvolvidas (i) uma que constroi padrões de corte e (ii) outra que elabora uma solução para o problema. Afim de comparar e avaliar a solução da metaheurística, implementou-se o clássico algoritmo de Geração de Colunas, e em seguida, resolveu-se o Problema Linear Inteiro com as colunas obtidas, usando-se o *Branch-and-Bound*. Os resultados e teste preliminares demostraram eficiência e robustez do método aqui desenvolvido, tendo em vista a qualidade das soluções encontradas e o tempo de processamento.

PALAVRAS-CHAVE: Problema do Corte, Otimização Combinatória, Algoritmo Memético.

1. INTRODUÇÃO

O PCUI é um dos problemas combinatórios mais estudados, devido, principalmente, a sua aplicabilidade no mundo da engenharia de produção, fazendo parte do planejamento de uma diversidade de indústrias que cortam papel, móveis, vidro, plásticos, tecido, entre outras. Apesar de ser muito simples de ser entendido, este problema têm um elevado nível de complexidade, sendo classificado na literatura como $\mathcal{NP}$-difícil Garey e Johnson (1996).

A tarefa de se obter uma solução ótima exata para o PCUI utilizando métodos clássicos de otimização, é desafiante. Isso se deve, principalmente, à integrabilidade e o elevado número de variáveis decisórias envolvidas neste problema. Alguns poucos algoritmos exatos para encontrar a solução ótima inteira do PCUI são conhecidos na literatura, como Carvalho (1999) e Vanderbeck (1996). No entanto, estes métodos podem ser aplicados apenas a problemas de pequeno porte.

Sendo assim, metodologias aproximativas têm sido desenvolvidas nas últimas cinco décadas para este problema. O primeiro trabalho a tratar o PCUI de maneira não exata foi Gilmore e Gomory (1961), que propuseram uma técnica de GC para obtenção de uma solução ótima contínua. Outros métodos não-exatos têm sido desenvolvidos e uma breve revisão é dada em Wascher e Gau (1996). Ressalta-se também o uso de Heurísticas de Arredondamento, proposto por Poldi e Arenales (2006), e algumas metaheurísticas específicas, desenvolvidas por Gehring e Bortfeldt (1998), Golfeto et al. (2008), Golfeto et al. (2009), entre tantos.

Mesmo que a otimalidade não seja garantida, o objetivo deste trabalho é apresentar um método heurístico de solução alternativo que forneça soluções “boas” e rápidas para este problema.

2. MODELAGEM MATEMÁTICA

Para modelar o PCUI, considere uma bobina mestre em estoque, de tamanho L (comprimento padrão), e m o número de itens demandados. Cada item tem comprimento

$l_i < L$ e ao menos d_i unidades precisam ser produzidas afim de atender a demanda necessária, $i = 1, ..., m$. O objetivo consiste em minimizar o número de peças a serem cortadas afim de atender à esta demanda mínima exigida. Do ponto de vista operacional, aqui apenas faz sentido cortar um número inteiro de vezes desta bobina mestre, o que torna este problema difícil de ser resolvido.

Uma instrução que lista os itens demandados a serem cortados da bobina principal é chamada de *padrão de corte*, que pode ser associado ao vetor m-dimensional

$$\mathbf{a}_j = (a_{1j}, a_{2j}, ..., a_{mj})^T,$$

onde cada entrada $a_{ij} \in \mathbb{N}$, denota a quantidade do item i presente no padrão j, $i = 1, ..., m$.

Definicão 2..1. *Um padrão de corte* $\mathbf{a}_j \in \mathbb{N}^m$ *é admissível se satisfaz as seguintes condições:*

$$\begin{aligned} \sum_{i=1}^{m} l_i \cdot a_{ij} &\leq L, & (1) \\ \sum_{i=1}^{m} l_i \cdot a_{ij} &\geq L - \Delta, & (2) \\ \sum_{i=1}^{m} a_{ij} &\leq F, & (3) \end{aligned}$$

onde $\Delta = \min_{1 \leq i \leq m} \{l_i\}$ *e* F *é o número de facas máximo permitido para produzí-lo.*

Suponha que existam n maneiras distintas de se formar todas as combinações de corte possíveis atendendo as restrições (1)-(2). Como visto, mesmo para m da ordem de dezenas, o valor de n pode facilmente atingir a casa dos milhões ou bilhões, desde que a razão L/l_i fique suficientemente pequena.

O modelo para o PCUI é apresentado a seguir:

$$\begin{aligned} \text{Minimize} \quad & z = \sum_{i=1}^{n} x_j & (4) \\ \text{sujeito a} \quad & \sum_{i=1}^{m} a_{ij} \cdot x_j \geq d_i, \; i = 1, ...m, & (5) \\ & x_j \in \mathbb{N}, \; j = 1, ..., n. & (6) \end{aligned}$$

A variável decisória deste problema é x_j, e denota a frequência do padrão j a ser utilizado, $j = 1, ..., n$. O conjunto de restrições (5) diz respeito ao atendimento mínimo da demanda exigida para cada item i.

Como salientado na seção anterior, este modelo tem dois agravantes, e que dificultam a obtenção da solução ótima exata $\mathbf{x}^* = (x_1^*, x_2^*, ..., x_n^*)^T$: (i) as condições de integrabilidade (6) e (ii) o elevado valor de n, o que torna inviável resolver o PLI (4)-(6) usando os métodos exatos clássicos como *Branch-and-*$\{$*Bound,Cut,Price*$\}$.

Afim de encontrar um limitante inferior $\underline{z}$ para o valor da função objetivo e avaliar a qualidade das soluções heurísticas, aplicou-se o algoritmo clássico GC e em seguida resolveu o PLI com as colunas de Gilmory-Gomory encontradas.

3. O ALGORITMO MEMÉTICO

Em virtude da elevada complexidade do PCUI e na impossibilidade de solucioná-lo via métodos exatos, apresenta-se uma metaheurística alternativa, afim de obter soluções admissíveis que aproximam $\mathbf{x}^*$ num tempo computacional aceitável. Para isto, foram desenvolvidas duas heurísticas: uma para construir padrões admissíveis, respeitando as restrições (1)-(3) e a outra para construir soluções factíveis para o problema atendendo (5).

A codificação de uma solução **S** para o PCUI é composta por uma matriz $\mathbf{A} \in \mathbb{N}^{m \times m}$, onde a coluna $\mathbf{a}_j$ é o j-ésimo padrão de corte, acoplada um vetor linha $\mathbf{x}^T \in \mathbb{N}^{1 \times m}$, cuja componente j indica a frequência do padrão $\mathbf{a}_j$ nesta solução, ou seja, $\mathbf{S} = \left[\frac{\mathbf{A}}{\mathbf{x}^T} \right]$.

Sendo assim, esses algoritmos fazem com que o método desenvolvido busque uma solução para o problema somente no espaço admissível. Os passos destes dois algoritmos são apresentados na próxima subseção.

3.1. HEURÍSTICAS CONSTRUTIVAS

O primeiro procedimento constroi padrões admissíveis e tem como dados de entrada m, F, l_i, L. Para evitar um laço infinito, um vetor **v** é inicializado, e indica a ordem dos itens que têm preferência de escolha para entrar no padrão a ser formado. Um pseudocódigo deste algoritmo é apresentado na Figura (1).

Procedimento 1: construção de padrões

Input: $\{m, F, l_i, L\}$
$Sobra = L$
Enquanto $Sobra > \Delta$
  Inicialize **v**
  Para i = 1 até m
    Tome o i-ésimo item de **v**
    Calcule a_{ij}: quantas vezes este item pode ser adicionado em uma fração da sobra deste padrão
    Se o número de facas não foi excedido, aloque a_{ij} na posição correspondente **Fim-Se**
    Atualize $Sobra = L - \sum_{k=1}^{i} l_k \cdot a_{kj}$
  Fim-Para
Fim-Enquanto
Output: $\mathbf{a}_j$

Figura 1: Algoritmo para construir padrões admissíveis

Apresenta-se na Figura (2) o procedimento que constroi soluções admissíveis para o problema.

Este algoritmo constroi soluções admissíveis ao PCUI quanto às restrições de demanda e que contenham, no máximo, m padrões. Caso a demanda tenha sido atendida mesmo antes de terem usado as m colunas de **S**, completa-se a dimensão desta matriz com colunas nulas. Isto pode ser útil quando se interessa em soluções que utilizem um menor *setup*.

Usando-se os **Procedimentos 1** e **2**, pode-se descrever o método Memético. Inicialmente, aplica-se um Algoritmo Genético (AG) clássico, iniciando-o com uma população com P indivíduos e G gerações. Em seguida, avalia-se cada indivíduo **S** segundo a função f:

$$f(\mathbf{S}) = \sum_{i=1}^{m} x_j + 10 \cdot \left(\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} \cdot x_j - d_i \right). \tag{7}$$

Procedimento 2: construção de soluções

Input: $\{m, F, l_i, L, \mathbf{d}\}$
$\mathbf{S} = \varnothing$
producao $= 0$
Falta $=$ **producao** $-$ **d**
Enquanto Falta < 0
Certifique os itens que ainda faltam para completar a demanda
Aloque, nas primeiras posições do vetor **v**, os itens cujas demandas não foram atendidas
Construa um padrão $\mathbf{a}_j$ utilizando o **Procedimento 1**
Calcule

$$x_j = \min_i \left\{ \left\lceil \frac{-Falta_i}{a_{ij}} \right\rceil : a_{ij} \neq 0 \wedge sobra_i > 0 \right\}$$

producao $=$ **producao** $+ x_j \cdot \mathbf{a}_j$
Falta $=$ **producao** $-$ **d**
$\mathbf{S} = \mathbf{S} \cup \left[\dfrac{\mathbf{a}_j}{x_j} \right]$
Fim-Enquanto
Output: **S**

Figura 2: Algoritmo para construir soluções admissíveis

Nesse sentido, prioriza-se as soluções com a menor superprodução possível. A melhor solução encontrada (elite) é preservada dos próximos operadores deste método. Em seguida, selecina-se 80% destes indivíduos, usando-se a roleta viciada, para realizar o ***Crossover*** **Uniforme**.

Dados dois indivíduo pareados, para obter novos indivíduos (filhos), realizaram-se cortes verticais em posições aleatórias nestas matrizes, afim de sempre preservar a factibilidade dos padrões de corte construídos. Entretanto, a factibilidade com relação à demanda pode ser perdida ao se gerar um filho. Se isto acontecer, a linha que contém as frequências (vetor $\mathbf{x}^T$) é recalculada utilizando os padrões que formam o novo indivíduo.

Após esta etapa, realiza-se o operador **Mutação** com probabilidade $\sigma(g)$

$$\sigma(g) = \frac{0,01}{0,01 + e^{-g/10}}, \tag{8}$$

onde $g = 1, ..., G$ é a geração corrente. Isso significa que este operador atuará mais intensamente nas gerações finais, impedindo e adiando a formação de uma população com características similares. A mutação, se ocorrer numa dada geração, escolhe 10% da população, e com probabilidade de 50% modifica os gens (que é uma coluna do tipo $(\mathbf{a}_j|x_j)^T$) destes indivíduos. Caso a factibilidade seja perdida, a mesma é recuperada no mesmo procedimento.

Após o AG finalizar sua busca, a elite encontrada serve como solução inicial para uma Busca Local clássica (BL) clássica, A única peculiaridade é a topologia da estrutura de vizinhança aqui empregada. Dada uma solução corrente **S**, define-se como um vizinho **V** uma outra solução que tem os $m-1$ padrões d original e o remanescente é gerado pelo **Procedimento 1**, e nesse caso, o vetor das frequencias para esta nova solução é calculado. Nessa BL encarra-se a busca quando o número de iterações (comparações) for igual à θm, θ um parâmetro.

A principal ideia de uma metaheurística híbrida é formar um método conjunto, capaz de pesquisar soluções mais eficientes do que se apenas um método (AG ou BL) fosse empregado.

4. PERSPECTIVAS FUTURAS

Este trabalho visou solucionar o PCUI utilizando a metaheurística Memética com duas herísticas construtivas. Afim de comparar e avaliar a solução da metaheurística, implementou-se o clássico algoritmo de Geração de Colunas, e em seguida, resolveu-se o Problema Linear Inteiro com as colunas obtidas, usando-se o *Branch-and-Bound*. Os resultados preliminates demostraram eficiência e robustez do método aqui desenvolvido, tendo em vista a qualidade das soluções encontradas e o tempo de processamento. Além disso, o método implementado obteve soluções mais interessantes que as fornecidas pelo concorrente, no tocante à superprodução, ao número de padrões utilizados e o desperdício. Isto pode ser entendido como um sub-produto do método apresentado, sendo uma ferramenta simples, fácil de ser implementada e do ponto de vista computacional, viável.

Como perspectiva futura de novos trabalhos, propõe-se utilizar estes algoritmos heurísticos para tratar o PCUI com múltiplos objetivos.

O trabalho presente é está ainda em andamento e é fruto das primeira pesquisas do aluno envolvido no seu programa de Doutorado. Na apresentação do trabalho, caso seja aceito, resultados mais conclusivos e detalhados serão apresentados.

AGRADECIMENTOS

Este trabalho tem apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, processo 2013/06035-0.

Referências

Carvalho, J. M. V. (1999). Exact Solution of Bin-packing Problems Using Column Generation and Branch-and-bound. *Annals of Operations Research*, 86, 629–659.

Garey, E. C. M.; Johnson, D. (1996). *Approximation Algorithms for Bin Packing: A Survey.* Boston. PWS. Approximation algorithms for NP-hard problems.

Gehring, H.; Bortfeldt, A. (1998). A Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operations Research*, 4, 401–418.

Gilmore, P. C.; Gomory, R. E. (1961). A Linear Programming Approach to the Cutting-Stock Problem. *Oper Res*, 9, 848–859.

Golfeto, R. R.; Moretti, A. C.,; Sales, L. L. N. (2008). A Grasp Metaheuristic for the Ordered Cutting Stock Problem. *Revista Chilena de Ingeniería (En línea)*, 16, 421–427.

Golfeto, R. R.; Moretti, A. C.,; Sales, L. L. N. (2009). A Genetic Symbiotic Algorithm Applied to the One-dimensional Cutting Stock Problem. *Pesquisa Operacional (impresso)*, 9, 365–382.

Poldi, K. C.; Arenales, M. N. (2006). Heurísticas para o Problema do Corte Unidimensional Inteiro. *Pesquisa Operacional*, 26, 473–492.

Vanderbeck, F. (1996). An Exact Algorithm for ip Column Generation. *Operations Research Letters*, 19, 151–159.

Wascher, G.; Gau, T. (1996). Heuristic for the Integer One-dimensional Cutting Stock Problem: a Computational Study. *OR Spektrum*, 18, 131–144.