



UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica
Departamento de Matemática Aplicada

Monografia de MS777 (Projeto Supervisionado I) - 1s2019

UM ESTUDO SOBRE O PROBLEMA DE CORTE DE ESTOQUE COM CUSTO DE PREPARAÇÃO

Aluna: Beatriz Aria Valladão

Supervisora: Dra. Kelly Cristina Poldi

Resumo

O problema de corte de estoque tem grande importância para indústrias que lidam com a transformação de materiais (objetos) em itens de tamanho menor. Um dos grandes interesses dessas indústrias é a minimização das perdas obtidas durante o corte, visto que estas perdas normalmente são onerosas para as indústrias além da preocupação com o excesso de resíduos gerados. Em alguns casos, a preparação da máquina para a execução de um padrão de corte pode ser custosa; assim, a redução do número de padrões de corte diferentes (*setup*) pode aumentar a produtividade nestas indústrias diminuindo o tempo de preparação das máquinas de corte. O objetivo deste projeto é estudar modelos matemáticos de programação linear inteira e métodos de solução existentes na literatura para o problema de corte de estoque unidimensional, com minimização da quantidade de objetos cortados para atender a demanda e do número de diferentes tipos de padrões de corte usados. A partir disso, implementações do modelo matemático foram feitas usando o pacote de otimização *CPLEX*, e testes computacionais básicos foram realizados para validar a implementação. Os resultados obtidos são apresentados e analisados.

1 Introdução

O Problema de Corte de Estoque (PCE) consiste em produzir itens (peças menores) a partir do corte de objetos (peças maiores) disponíveis em estoque, atendendo as demandas exigidas para cada tipo de item. É desejado montar um plano de corte que, além de atender as demandas, minimize a quantidade de material e também o número de diferentes padrões de corte utilizados (diminuindo assim o tempo de preparação das máquinas).

Definimos um padrão de corte como uma combinação possível dos itens sobre um objeto do estoque. A Figura 1, retirada de [14], ilustra esses conceitos.

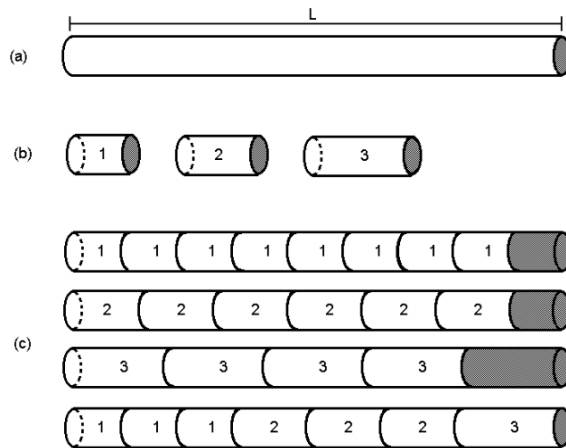


Figura 1: (a) representa um dos objetos em estoque; (b) indica os itens a serem cortados; e (c) mostra alguns possíveis padrões de corte desses itens sobre o objeto. Fonte: [14].

1.1 Modelo matemático

Estudamos o PCE unidimensional, ou seja, os itens e objetos são analisados em seu comprimento. Sejam:

- m a quantidade de tipos de itens;
- n a quantidade de padrões de corte factíveis;
- L o comprimento do objeto;
- ℓ_i o comprimento do item tipo i , $i = 1, \dots, m$;
- d_i a demanda do item tipo i , $i = 1, \dots, m$;
- α_j o vetor associado ao padrão de corte j , $j = 1, \dots, n$;
- x_j a quantidade de vezes que o padrão de corte j é utilizado, $j = 1, \dots, n$.

Um padrão de corte pode ser representado por um vetor α_j , no qual cada componente α_{ij} armazena a quantidade de itens do tipo i presentes neste padrão de corte j . Isto é, o vetor $\alpha_j = (\alpha_{1j}, \dots, \alpha_{mj})^t$ representa um padrão de corte se e somente se satisfizer

$$\begin{aligned} \ell_1 \alpha_{1j} + \ell_2 \alpha_{2j} + \dots + \ell_m \alpha_{mj} &\leq L, \\ 0 \leq \alpha_{ij} \leq d_i, \quad \alpha_{ij} \in \mathbb{Z}^+, \quad i &= 1, \dots, m, \quad j = 1, \dots, n. \end{aligned} \quad (1)$$

Queremos determinar α_j e x_j ($j = 1, \dots, n$) que minimize a **perda total de material** e a **quantidade de diferentes padrões de corte usados**. O modelo matemático que representa esse problema é dado por:

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x})) \\ \text{sujeito a:} \quad & \sum_{j=1}^n \alpha_{ij} x_j \geq d_i \quad i = 1, \dots, m, \\ & x_j \in \mathbb{Z}^+ \quad j = 1, \dots, n. \end{aligned} \quad (2)$$

em que $f_1(x) = \sum_{j=1}^n x_j$, representando o total de objetos usados nos cortes e $f_2(\mathbf{x})$ é a função referente à quantidade de padrões de corte, onde:

$$f_2(\mathbf{x}) = \sum_{j=1}^n \delta_j(x_j), \quad \text{onde } \delta_j(x_j) = \begin{cases} 1, & \text{se } x_j > 0, \quad j = 1, \dots, n, \\ 0, & \text{caso contrário.} \end{cases} \quad (3)$$

Assim, as restrições de (2) garantem o cumprimento da demanda de cada item e a integralidade da solução, e a função objetivo $f_2(\mathbf{x})$, definida por (3), indica se o padrão de corte j foi ou não utilizado.

1.2 Revisão bibliográfica

Alguns métodos de solução para o problema proposto foram apresentadas por Arroyo [2], Araújo *et al.* [1], Foerster e Wäscher [8], Yanasse e Limeira [17], Kobersztajn [12] e Vanderbeck [16]. Uma revisão dessa literatura foi feita, captando os pontos importantes para o projeto. Além dessas, outras referências importantes sobre otimização multiobjetivo são Ehrgott [7] e Cui *et al.* [4].

1.2.1 Arroyo

Em Arroyo [2] é apresentado um panorama sobre problemas de otimização multiobjetivos e alguns métodos mais tradicionais de resolução desse tipo de problema.

Nos problemas multiobjetivos temos normalmente objetivos conflitantes, e portanto diferentes soluções eficientes são encontradas, todas com igual importância para o decisor. Arroyo [2] introduz os conceitos de **dominância** e **otimalidade** de Pareto.

Sejam \mathbf{v}_1 e \mathbf{v}_2 soluções para um problema de otimização. É dito que a solução \mathbf{v}_1 domina \mathbf{v}_2 se $\mathbf{f}(\mathbf{v}_1) \leq \mathbf{f}(\mathbf{v}_2)$ com $\mathbf{f}(\mathbf{v}_1) \neq \mathbf{f}(\mathbf{v}_2)$. Assim, uma solução é chamada *eficiente* ou *Pareto-ótima* se

ela não é dominada por qualquer outra solução. A partir disso, o conjunto de todas as soluções eficientes forma o conjunto Pareto-ótimo, e sua imagem (conjunto dos valores de função objetivo) é denominada **fronteira Pareto-ótima**. A Figura 2, adaptada de Arroyo [2], ilustra os conceitos definidos.

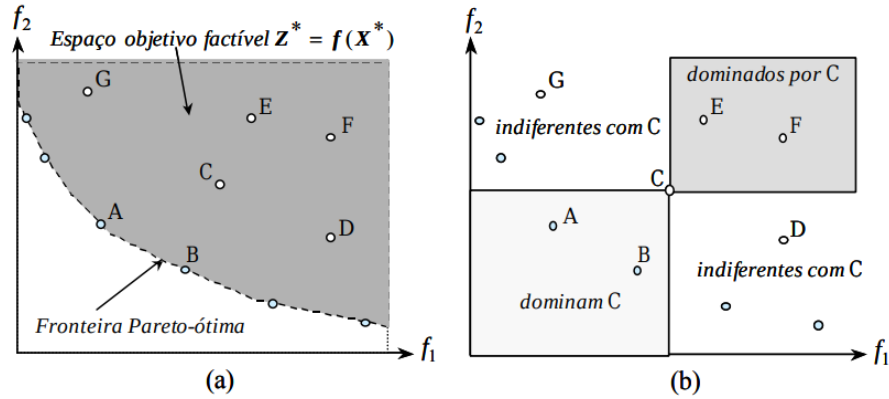


Figura 2: (a) Fronteira eficiente ou Pareto-ótima; (b) Relação de dominância entre as soluções do problema. Fonte:[2].

Além disso, dois métodos de resolução para esse problema foram apresentados:

- **Método da Soma Ponderada**

Transforma o problema multiobjetivo em um problema mono-objetivo, fazendo uma soma ponderada de todas as funções objetivos:

$$\begin{aligned}
 &\text{minimizar} && f(\mathbf{x}) = \sum_{k=1}^r w_k f_k(\mathbf{x}) \\
 &\text{sujeito a:} && \sum_{j=1}^n \alpha_{ij} x_j \geq d_i && i = 1, \dots, m, \\
 &&& x_j \in \mathbb{Z}^+ && j = 1, \dots, n.
 \end{aligned} \tag{4}$$

onde r é o número de funções objetivos que deseja-se minimizar, e w_k é o peso/importância que a função k recebe. Normalmente, adota-se $\sum_{k=1}^r w_k = 1$, com $w_k \geq 0$ ($k = 1, \dots, r$).

Ou seja, o decisor pode priorizar os objetivos da maneira conveniente. Entretanto, quando a região factível é não-convexa, o método não consegue gerar todas as soluções eficientes.

- **Método ϵ -restrito**

Esse método minimiza a função objetivo de maior prioridade, enquanto restringe as outras funções por um limitante:

$$\begin{aligned}
& \text{minimizar} && f_\ell(\mathbf{x}) \\
& \text{sujeito a:} && f_k(\mathbf{x}) \leq \epsilon_k \quad k = 1, \dots, r, \quad k \neq \ell, \\
& && \sum_{j=1}^n \alpha_{ij} x_j = d_i \quad i = 1, \dots, m, \\
& && x_j \in \mathbb{Z}^+ \quad j = 1, \dots, n.
\end{aligned} \tag{5}$$

Com isso, é possível também resolver o PCE para regiões viáveis não-convexas. Ademais, o limitante superior deve ser escolhido de forma que não elimine pontos eficientes.

1.2.2 Araújo et al.

Araújo *et al.* [1] propõem um Algoritmo Genético (AG) para solucionar o PCE unidimensional com custo de preparação.

A função objetivo apresentada por Araújo *et al.* [1] determina, a partir do conjunto de características (genótipo) de um indivíduo, a sua habilidade (fenótipo) para paternidade/maternidade. Cada indivíduo representa uma **solução factível** do problema, ou seja, cada gene representa um *padrão de corte* α_j e também a *frequência* x_j que o padrão é cortado. O objetivo é determinar um conjunto de indivíduos (população) com habilidade máxima para paternidade/maternidade.

Elementos são eliminados a cada iteração, simulando o processo de seleção natural teorizado por Darwin, e o critério de remoção desses indivíduos é arbitrário. A partir dos elementos restantes, recombinações são aplicadas, gerando filhos com genes misturados, e mutações são feitas, resultando em uma nova geração de indivíduos.

O algoritmo genético se resume à: inicialmente, cria-se uma população aleatória cujos indivíduos tem diferentes características (soluções factíveis variadas); para formar uma nova geração (nova iteração), os indivíduos são avaliados, a partir dessas características, por suas respectivas habilidades em serem pais (valor na função objetivo); indivíduos são escolhidos e eliminados, de maneira que uma nova geração possa ser formada com os indivíduos ainda vivos, a partir de recombinações genéticas entre eles e mutações; obtemos com isso uma nova população (novo conjunto de soluções).

O processo se repete até que uma certa quantidade de gerações seja determinada, ou até que a demanda de cada item seja atendida.

1.2.3 Foerster e Wäscher

Foerster e Wäscher [8] propõem o método KOMBI para redução de padrões de corte em PCE unidimensional. A ideia é fazer combinações de padrões que resultem em padrões com frequências mais altas, diminuindo assim a quantidade de padrões diferentes usados, e mantendo o número de itens produzidos.

Primeiramente, foi apresentada a combinação de dois padrões de corte para um (2-1) com respectivas frequências x_1 e x_2 . Para cada item $i = 1, \dots, m$ demandado, $s_i = \alpha_{i1}x_1 + \alpha_{i2}x_2$ é o comprimento total produzido por esses padrões. O padrão de corte resultante da combinação dos dois originais é dado por $*$ $= (\alpha_{1*}, \dots, \alpha_{m*})$, onde $\alpha_{i*} = s_i/x_*$ e $x_* = x_1 + x_2$, se todas as razões s_i/x_* são **inteiras**. Com isso, é garantida a obtenção do mesmo número de itens, usando apenas um padrão de corte.

A extensão do problema para combinações 3-2, 3-1, 4-3, 4-2, e 4-1 pode ser feita recursivamente, tendo como caso base a combinação 2-1. Toda vez que a recursão é chamada, a condição de factibilidade deve ser conferida, ou seja, o processo para se $\sum_{i=1}^m l_i \alpha_{i*} > L$ ou $\sum_{i=1}^m l_i \alpha_{i**} > L$.

1.2.4 Yanasse e Limeira

Uma heurística híbrida é estudada por Yanasse e Limeira [17], visando reduzir o número de padrões de corte no PCE. A proposta é gerar padrões (cuja frequência seja suficiente para atender a demanda de pelo menos dois itens) com perda limitada, e a partir disso, usa técnicas para reduzir os padrões de corte.

Ao reduzirmos os padrões, talvez ocorra o aumento da frequência dos padrões de corte para que as demandas continuem a ser atendidas. É dito que, padrões com frequência X devem conter itens de demandas maiores ou iguais a X , caso contrário haverá superprodução dos itens.

Propuseram então o uso do método RPET (*Repeated Pattern Exhaustion Technique*), que busca padrões de corte compostos por itens com demandas próximas ou que sejam múltiplas entre si. Um padrão é inserido no plano de corte se preenche os requisitos a seguir, e o método para se não encontrar mais padrões de corte desse tipo.

- Frequência G : Com um padrão formado por itens de demanda parecidas (ou múltiplas), G é escolhido de forma a ser um submúltiplo da demanda de alguns dos itens, para que, em G cortes, a demanda de pelo menos 2 itens seja atendida (sem superprodução).
- Eficiência de pelo menos β : β é calculado a partir de δ , que representa a perda total de material obtida resolvendo o PCE atual.
- A demanda de mais de um item é atendida ao final dos G cortes

A heurística é dividida em partes. Primeiramente, padrões de corte são gerados e os bons (citados acima) são selecionados, usados para construir um problema reduzido. Depois de resolver o problema residual, são usadas as técnicas de redução de padrões de corte, como a citada em Foerster e Wäscher [8].

1.2.5 Kobersztajn

Kobersztajn [12] apresenta uma resolução do PCE unidimensional inteiro com dois objetivos, com objetos de diferentes tamanhos. Primeiro, resolve o problema de minimização do comprimento

total de objetos cortados, e em seguida reduz a quantidade de padrões de corte usados.

Primeiramente, alguns modelos matemáticos para o PCE (objetos de tamanho único), tendo como função objetivo minimizar a quantidade de objetos cortados, são apresentados:

Gilmore e Gomory [9, 10] formulam o problema para ser resolvido usando o método simplex, relaxando as restrições de integralidade. A solução encontrada pode não ser inteira, porém os limitantes obtidos são fortes, o que possibilita uma boa aproximação. Além disso, devido a alta variedade de padrões de cortes factíveis, Gilmore e Gomory [9, 10] propõem o método de *Geração de Colunas* para obter um bom padrão de corte (aquele que diminui o valor da função objetivo) a cada iteração, e substituí-lo na matriz básica do problema. O problema auxiliar que deve ser resolvido para encontrar a melhor coluna a ser substituída é um *Problema da Mochila* associado às variáveis duais do PCE. Essa técnica será revisada na seção (2).

Valério de Carvalho [5, 6] cria um modelo de fluxo em arcos, onde cada unidade de comprimento do objeto representa um nó no grafo, e o comprimento de um item é um arco entre dois nós. Além disso, arcos de perda são adicionados e normalmente tem comprimento unitário. Assim, um conjunto de arcos que formam um caminho entre os nós representa um padrão de corte factível. O PCE é definido como encontrar o fluxo mínimo entre os nós extremos do grafo, isto é, caminhos cuja frequência de uso de tal padrão de corte seja mínima. Um fluxo mínimo e um conjunto de caminhos mínimos determina a solução ótima. Valério de Carvalho [5, 6] mostra alguns critérios de redução, que visam remover padrões de corte redundantes, como: ordenar os arcos pelo seu comprimento, de maneira não crescente; inserir arcos de perda no final do objeto; garantir que a quantidade de arcos do mesmo item (em um único objeto) não ultrapasse a demanda desse item.

Gilmore e Gomory [9, 10] propuseram uma extensão da resolução já citada, porém tendo como objetivo minimizar a perda total em comprimento de material para todos os padrões de corte. Além da restrição de atendimento da demanda, também foi inserida uma restrição sobre a disponibilidade dos objetos em estoque.

Valério de Carvalho [5, 6] também propôs um modelo estendido. Dessa vez, a ordem (número de nós) do grafo é definida pelo tamanho do maior objeto em estoque, também sendo discretizado em unidades de comprimento. Analogamente à formulação anterior, deseja-se minimizar o fluxo pelo grafo, respeitando as demandas e também a disponibilidade dos objetos.

Após encontrar uma solução ótima pelo modelo de Valério de Carvalho [5, 6], Kobersztajn [12] apresenta um método para reconstruir os padrões de corte obtidos, já que este modelo fornece somente o valor do fluxo mínimo e dos arcos que compõem essa solução.

Para isso, um algoritmo de busca é apresentado. O algoritmo faz a reconstrução de padrões de corte (para um ou mais tipos de objetos) mapeando iterativamente caminhos formados pelos arcos da solução ótima, *ie*, arcos cujos fluxos não são nulos.

1.2.6 Vanderbeck

Vanderbeck [16] apresenta uma reformulação do problema, que é decomposto em múltiplos problemas da mochila e resolvido por um algoritmo do tipo *branch-and-bound*.

No problema de minimização de padrões de corte, temos um **limitante superior** K dado, que representa o número de objetos máximo à serem cortados. Assim, queremos encontrar padrões de corte para usar em (no máximo) K objetos, satisfazendo o atendimento das demandas e minimizando a quantidade de padrões de corte usados. Normalmente K é determinado resolvendo o PCE clássico (somente minimizando as perdas de material).

Assim, é fácil ver que a quantidade máxima de padrões de corte válidos para o problema será K , e definindo z_k como a quantidade de padrões do tipo k usados, y_k indicando se o objeto k é cortado, o que nos permite escrever o problema:

$$\begin{aligned}
 Z = \min \quad & \sum_{k=1}^K y_k \\
 \text{sujeito a:} \quad & \sum_{k=1}^K x_k \alpha_{ik} = d_i \quad i = 1, \dots, m, \\
 & \sum_{k=1}^K x_k \leq K, \\
 & x_k \leq K y_k \quad k = 1, \dots, K, \\
 & \sum_{i=1}^m l_i \alpha_{ik} \leq L y_k \quad k = 1, \dots, K, \\
 & \alpha_{ik} \in \mathbb{Z}^+ \quad (i = 1, \dots, m) \quad k = 1, \dots, K, \\
 & y_k \in \{0, 1\} \text{ e } x_k \in \mathbb{Z}^+ \quad k = 1, \dots, K.
 \end{aligned} \tag{6}$$

A ideia do método é aplicar um algoritmo de *branch-and-price-and-cut*, que consiste em inicialmente usar uma relaxação linear forte com auxílio da Geração de Colunas, e então, adicionando planos de corte ao problema, obter um bom **limitante inferior**. A partir disso, o processo de *branch-and-bound* é aplicado ao modelo.

2 Geração de Colunas

2.1 Método

Em Poldi [14], é revisado o método de Geração de Colunas para o PCE (Gilmore e Gomory [9, 10]). Considere um problema de otimização na *forma padrão* com a condição de integralidade relaxada, reescrito na forma matricial

$$\begin{aligned}
 \text{minimizar} \quad & f(\mathbf{x}) = \mathbf{c}^t \mathbf{x} \\
 \text{sujeito a:} \quad & \mathbf{Ax} \geq \mathbf{d} \\
 & \mathbf{x} \in \mathbb{R}^+
 \end{aligned} \tag{7}$$

onde $\mathbf{A} \in \mathbb{R}^{m \times n}$ é a matriz formada pelos padrões de corte e que pode ser reescrita pela partição básica $\mathbf{A} = [\mathbf{B}, \mathbf{N}]$, com $\mathbf{B} \in \mathbb{R}^{m \times m}$ sendo a *matriz básica* e $\mathbf{N} \in \mathbb{R}^{m \times (n-m)}$ a *matriz não-básica*. Além disso, os vetores \mathbf{x} e \mathbf{c} também são particionados, em $\mathbf{x} = [\mathbf{x}_B, \mathbf{x}_N]$ (*vetor das variáveis básicas e não-básicas, respectivamente*) e $\mathbf{c} = [\mathbf{c}_B, \mathbf{c}_N]$ (*vetor de custos básicos e não-básicos, respectivamente*).

Portando, obtemos

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) = \mathbf{c}_B^t \mathbf{x}_B + \mathbf{c}_N^t \mathbf{x}_N \\ \text{sujeito a:} \quad & \mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{d}, \\ & \mathbf{x}_B, \mathbf{x}_N \in \mathbb{R}^+. \end{aligned} \tag{8}$$

Manipulando o primeiro conjunto de restrições, temos que $BfBx_B + Nx_N = d \Rightarrow x_B = B^{-1}d - B^{-1}Nx_N$. Uma solução factível é da forma $\mathbf{x}^0 = [\mathbf{x}_B^0, \mathbf{x}_N^0] = [\mathbf{B}^{-1}\mathbf{d}, \mathbf{0}]$ com $\mathbf{B}^{-1}\mathbf{d} \geq \mathbf{0}$, então $f(\mathbf{x}^0) = \mathbf{c}_B^t \mathbf{B}^{-1}\mathbf{d}$. Definindo o *vetor das variáveis duais* (associadas ao primeiro conjunto de restrições) como $\pi \in \mathbb{R}^m$, tal que $\pi = \mathbf{c}_B^t \mathbf{B}^{-1}$, temos uma solução básica *dual factível* se $(c_j - \pi^t \alpha_j) \geq 0$, $j = 1, \dots, n$, com α_j sendo a j -ésima coluna de \mathbf{A} .

A função objetivo, em termos das variáveis não-básicas, fica então $f(\mathbf{x}) = f(\mathbf{x}^0) + \sum_{j \in N} (c_j - \pi^t \alpha_j) x_j$, e os coeficientes de x_j , $(c_j - \pi^t \alpha_j)$, são chamados **custos reduzidos/relativos**. A estratégia simplex é encontrar, a cada iteração, o padrão de corte j de *menor* custo reduzido, que melhora o valor da função objetivo.

Para resolver esse problema pelo *método simplex*, teríamos que enumerar todos os padrões de cortes factíveis. Porém, na maioria dos casos, essa quantidade é muito grande, tornando o processo inviável. Então, o método de Geração de Colunas foi criado, e a partir de alguns padrões conhecidos, resolve pelo método simplex o problema (2) **relaxado**, e a cada iteração do método usamos as variáveis duais associadas para montar o subproblema $(c_k - \pi^t \alpha_k) = \min \{c_j - \pi^t \alpha_j\}$ ($j = 1, \dots, n$) que gera o melhor padrão de corte k para aquela iteração, adicionando-o ao plano de corte do problema mestre.

Esse subproblema é um *Problema da Mochila*, que usa o padrão de corte k (que seria a k -ésima coluna de \mathbf{A}) como a variável de decisão:

$$\begin{aligned} g_k(\pi) = \text{minimizar} \quad & c_k - \sum_{i=1}^m \pi_i \alpha_{ik} \\ \text{sujeito a:} \quad & \sum_{i=1}^m \alpha_{ik} \ell_i \leq L, \\ & \alpha_{ik} \in \mathbb{Z}^+ \quad i = 1, \dots, m. \end{aligned} \tag{9}$$

Quando o método não encontra mais um bom padrão de corte, *ie*, $g_k(\pi) < 0$, a Geração de Colunas para e resolvemos por fim o problema (7). A seguir, consideramos todos os padrões de corte gerados e então, resolvemos o problema (2) que determina uma solução inteira para o problema.

2.2 Algoritmo

Primeiro, foram criados os programas que resolvem o modelo do problema original (*PCE* 1D Inteiro) e o subproblema associado (o respectivo problema da mochila *GC*). Com isso, temos o arquivo de controle de fluxo, programa que relaxa a integralidade do problema para encontrar os padrões de corte candidatos à solução, como no esquema (1).

2.3 Implementação

Consideramos dois modelos: **modelo 1** que representa o PCE com objetivo de minimizar somente f_1 (mono-objetivo) e **modelo 2** que representa o PCE com ambos os objetivos, minimizando f_1 e f_2 .

- **Modelo 1**

No primeiro caso, o problema mestre é:

$$\begin{aligned} z_1 = \text{minimizar} \quad & \sum_{j=1}^n x_j \\ \text{sujeito a:} \quad & \sum_{j=1}^n \alpha_{ij} x_j \geq d_i \quad i = 1, \dots, m, \\ & x_j \in \mathbb{Z}^+ \quad j = 1, \dots, n. \end{aligned} \tag{10}$$

Como o custo de uso dos padrões de corte é $c_j = 1$ ($j = 1, \dots, n$), o subproblema associado para encontrar o padrão de corte k é

$$\begin{aligned} g_{1k}(\pi) = \text{minimizar} \quad & 1 - \sum_{i=1}^m \pi_i \alpha_{ik} \\ \text{sujeito a:} \quad & \sum_{i=1}^m \alpha_{ik} \ell_i \leq L, \\ & \alpha_{ik} \in \mathbb{Z}^+ \quad i = 1, \dots, m. \end{aligned} \tag{11}$$

- **Modelo 2**

Ao invés de usar a função $\delta(\mathbf{x})$ explicitamente, adicionamos o segundo conjunto de restrições para controlar seu valor, onde M é um valor tão grande quanto se queira (limitante superior da frequência de corte). Portanto, para esse modelo, problema mestre é:

$$\begin{aligned} z_2 = \text{minimizar} \quad & \sum_{j=1}^n x_j + \sum_{j=1}^n \delta_j(x_j) \\ \text{sujeito a:} \quad & \sum_{j=1}^n \alpha_{ij} x_j \geq d_i \quad i = 1, \dots, m, \\ & x_j \leq M \delta_j(x_j) \quad j = 1, \dots, n, \\ & x_j \in \mathbb{Z}^+, \delta_j(x_j) \in \{0, 1\} \quad j = 1, \dots, n. \end{aligned} \tag{12}$$

Algoritmo 1: GERAÇÃO DE COLUNAS

Entrada: Arquivo com os dados do problema: m, L, ℓ_i, d_i

Saída: Padrões de corte e as respectivas frequências

início

INICIALIZAÇÃO:

Leitura dos dados de entrada;

Monta o problema mestre *PCE*;

Monta o subproblema *GC*;

Adiciona os dados a *PCE*;

best = 0;

curr = ∞ ;

$\varepsilon = 10^{-7}$;

repita

 best = curr;

 RESOLVE O PCE RELAXADO;

se tem solução então

 | curr = solução do PCE;

senão

 | O programa para;

fim

 Adiciona os dados do *PCE* ao *GC*;

\leftrightarrow Inclui as variáveis **duais** associadas às restrições do *PCE*

 RESOLVE *GC*;

se tem solução então

 | Adiciona o padrão encontrado (solução de *GC*) aos já existentes em *PCE*;

senão

 | O programa para;

fim

se solução do GC > - ε então

 | O programa para;

fim

até best = curr;

RESOLVE O *PCE* ORIGINAL;

se tem solução ótima então

 | Imprime os padrões de corte e as frequências;

fim

fim

Nesse modelo, as variáveis duais são $\pi = [\eta \ \mu]^t$, onde η e μ se associam ao primeiro e segundo conjunto de restrições, respectivamente. De forma análoga ao item anterior, para o primeiro conjunto de restrições temos $c_{\eta j} = 1$ ($j = 1, \dots, n$) e $\alpha_{\eta i} = \alpha_i$ ($i = 1, \dots, m$). Já para o segundo conjunto, não temos custo associado ao padrão de corte (segunda parcela da função objetivo), *ie*, $c_j = 0$ ($j = 1, \dots, n$) e $\alpha_{\eta j} = \delta_j$ ($j = 1, \dots, n$). Dessa forma, foi construído o subproblema, para encontrar o padrão de corte k :

$$\begin{aligned}
g_{2k}(\eta, \mu) = \quad & \text{minimizar} \quad 1 - \sum_{i=1}^m \eta_i \alpha_{ik} - \sum_{j=1}^n \mu_j \\
\text{sujeito a:} \quad & \sum_{i=1}^m \alpha_{ik} \ell_i \leq L, \\
& \alpha_{ik} \in \mathbb{Z}^+ \quad \quad \quad i = 1, \dots, m.
\end{aligned} \tag{13}$$

3 O Método das Somas Ponderadas

O método das Somas Ponderadas (Sampaio [15], Grodzevich e Romanko [11]) consiste em transformar um problema multiobjetivo em um problema mono-objetivo, atribuindo pesos diferentes a cada uma das funções objetivos, e analisando a soma delas. Aplicando esse método para o caso do PCE biobjetivo (2), queremos encontrar as soluções ótimas (padrões de corte α_j e suas respectivas frequências x_j) para o problema

$$\begin{aligned}
\text{minimizar} \quad & \omega f_1(\mathbf{x}) + (1 - \omega) f_2(\mathbf{x}) \\
\text{sujeito a:} \quad & \sum_{j=1}^n \alpha_{ij} x_j \geq d_i \quad \quad \quad i = 1, \dots, m, \\
& x_j \leq M \delta_j(x_j) \quad \quad \quad j = 1, \dots, n, \\
& x_j \in \mathbb{Z}^+, \delta_j(x_j) \in \{0, 1\} \quad j = 1, \dots, n,
\end{aligned} \tag{14}$$

onde ω é o peso (importância) associado.

Porém as funções objetivos podem ter magnitudes diferentes, prejudicando a confiabilidade da solução, sendo então necessária a aplicação de uma normalização, que é construída usando os chamados pontos **Nadir** e **Utopia**.

Para o PCE multiobjetivo, o ponto Utopia z_k^U de um objetivo é o valor de $f_k(\mathbf{x})$ encontrado ao resolver:

$$\begin{aligned}
\text{minimizar} \quad & f_k(\mathbf{x}) + \sum_{r \neq k} \rho_r f_r(\mathbf{x}) \\
\text{sujeito a:} \quad & \sum_{j=1}^n \alpha_{ij} x_j \geq d_i \quad \quad \quad i = 1, \dots, m, \\
& x_j \leq M \delta_j(x_j) \quad \quad \quad j = 1, \dots, n, \\
& x_j \in \mathbb{Z}^+, \delta_j(x_j) \in \{0, 1\} \quad j = 1, \dots, n,
\end{aligned} \tag{15}$$

onde ρ_r ($\forall r \neq k$) é um número muito pequeno. Ou seja, é o menor valor da função objetivo $f_k(\mathbf{x})$ desconsiderando a influência das outras funções (limitante inferior). E ponto Nadir z_k^N de um objetivo é o maior valor que $f_k(\mathbf{x})$ pode assumir quando calculamos o ponto Utopia das outras funções (limitante superior).

Com isso, sabemos que $z_k^U \leq f_k(\mathbf{x}) \leq z_k^N$, e ao definirmos as funções $h_k(\mathbf{x}) = \frac{f_k(\mathbf{x}) - z_k^U}{z_k^N - z_k^U}$, é fácil ver que a relação $0 \leq h_k(\mathbf{x}) \leq 1$ é sempre válida, e portanto podemos usá-las como novas funções objetivos para o problema, dado que têm a mesma magnitude.

Foi criada uma função recursiva (Algoritmo 2) para enumerar todos os padrões de corte factíveis, e então resolvemos o problema (15) exato, obtendo os valores de z_1^U , z_1^N , z_2^U e z_2^N .

Algoritmo 2: PADROES_DE_CORTE (ITEM_ATUAL, ESPAÇO_VAZIO, PADRAO)

início

se *espaço_vazio* < *tamanho do menor item* **então**

retorna;

fim

prox_item = *item_atual*;

repita

 se *espaço_vazio* - *tamanho do prox_item* ≥ 0 **então**

 Adiciona o *prox_item* ao *padrao*;

\hookrightarrow PADROES_DE_CORTE (**PROX_ITEM**, **ESPAÇO_VAZIO - TAMANHO DO PROX_ITEM**, **PADRAO**);

 Remove o *prox_item* do *padrao*;

fim

prox_item = *prox_item* + 1;

até *prox_item* > *último item da lista*;

retorna;

fim

3.1 Geração de Colunas

Montamos então as funções normalizadas $h_1(\mathbf{x}) = \frac{f_1(\mathbf{x}) - z_1^U}{z_1^N - z_1^U}$ e $h_2(\mathbf{x}) = \frac{f_2(\mathbf{x}) - z_2^U}{z_2^N - z_2^U}$, e determinamos o **modelo 3**, que resolve o PCE usando o método das Somas Ponderadas com as funções normalizadas:

$$\begin{aligned}
& \text{minimizar} && \omega h_1(\mathbf{x}) + (1 - \omega)h_2(\mathbf{x}) \\
& \text{sujeito a:} && \sum_{j=1}^n \alpha_{ij}x_j \geq d_i && i = 1, \dots, m, \\
& && x_j \leq M\delta_j(x_j) && j = 1, \dots, n, \\
& && x_j \in \mathbb{Z}^+, \delta_j(x_j) \in \{0, 1\} && j = 1, \dots, n,
\end{aligned} \tag{16}$$

com ω sendo o respectivo peso/importância da função objetivo. Variando $\omega \in (0, 1)$, obtemos diferentes soluções para o problema.

Devido ao custo computacional de se trabalhar com todas as colunas (padrões de corte) do problema, utilizamos o método de Geração de Colunas. Portanto, o Problema de Mochila associado ao **modelo 3** para calcular o padrão de corte k é:

$$\begin{aligned}
g_{3k}(\eta, \mu) = \text{minimizar} && \frac{\omega}{z_1^N - z_1^U} - \sum_{i=1}^m \eta_i \alpha_{ik} - \sum_{j=1}^n \mu_j \\
&& \text{sujeito a:} && \sum_{i=1}^m \alpha_{ik} \ell_i \leq L, \\
&& && \alpha_{ik} \in \mathbb{Z}^+ && i = 1, \dots, m.
\end{aligned} \tag{17}$$

4 ε -restrito

No método ε -restrito (Bérubé *et al.* [3]), escolhemos uma função para ser priorizada, e as demais funções são transformadas em restrições. Para o PCE biobjetivo (2), temos o **modelo 4**:

$$\begin{aligned}
& \text{minimizar} && f_r(\mathbf{x}) \\
& \text{sujeito a:} && f_s(\mathbf{x}) \leq \varepsilon_s \\
& && \sum_{j=1}^n \alpha_{ij}x_j \geq d_i && i = 1, \dots, m, \\
& && x_j \leq M\delta_j(x_j) && j = 1, \dots, n, \\
& && x_j \in \mathbb{Z}^+, \delta_j(x_j) \in \{0, 1\} && j = 1, \dots, n.
\end{aligned} \tag{18}$$

Resolvendo o modelo para $(r, s) = (1, 2)$ e $(r, s) = (2, 1)$, utilizando todos os padrões de corte do problema, obtemos os pontos na curva de Pareto-ótima.

As constantes ε_1 e ε_2 são limitantes superiores para as funções, e a ideia do método é resolver o modelo (5) para diferentes valores dos limitantes, obtendo soluções variadas. Conhecendo os pontos Nadir das funções e sabendo que a solução é inteira, iniciamos os limitantes em $\varepsilon_i = z_i^N - \Delta$ e a cada iteração diminuímos esses valores, fazendo $\varepsilon_i \leftarrow z_i^* - \Delta$, com $\Delta = 1$, onde z_i^* é o valor de $f_i(x)$ na solução ótima encontrada. Esse processo está esquematizado no Algoritmo (3).

Algoritmo 3: ε -RESTRITO

Entrada: Arquivo com os dados do problema + pontos Nadir e Utopia das funções

Saída: Padrões de corte e respectivas frequências

$\Delta = 1;$

PARA $(r, s) = (1, 2):$

início

$\varepsilon_2 = z_2^N - \Delta;$

repita

$\hookrightarrow (z_1^*, z_2^*) = \text{SOLUÇÃO DO PCE EXATO};$

Imprime a solução encontrada;

$\varepsilon_2 \leftarrow z_2^* - \Delta;$

até $\varepsilon_2 < z_2^U;$

fim

PARA $(r, s) = (2, 1):$

início

$\varepsilon_1 = z_1^N - \Delta;$

repita

$\hookrightarrow (z_1^*, z_2^*) = \text{SOLUÇÃO DO PCE EXATO};$

Imprime a solução encontrada;

$\varepsilon_1 \leftarrow z_1^* - \Delta;$

até $\varepsilon_1 < z_1^U;$

fim

5 Testes computacionais

Com o pacote de otimização *IBM ILOG CPLEX Optimization Studio* versão 12.7.1.0 [13], resolvemos: os **modelos 1** (10) e **2** (12) usando o método de Geração de Colunas, revisado na Seção (2); o **modelo 3** (16) pelo método das Somas Ponderadas, utilizando o método exato e o método de Geração de Colunas, revisado na Seção (3); e o **modelo 4** (18) de maneira exata, usando o método ε -restrito, revisado na Seção (4).

5.1 Modelos 1 e 2

O exemplo numérico 1 apresentado a seguir foi retirado de [14]. Considere $L = 1000$ o tamanho dos objetos em estoque. A Tabela 1 contém os demais parâmetros do problema.

A Tabela 2 expõe os resultados encontrados no teste realizado para cada modelo proposto, mostrando que quando o **modelo 2** é aplicado, temos um aumento do número de objetos usados

Item i	1	2	3	4	5	6	7	8	9	10
Tamanho ℓ_i	173	194	97	118	22	163	184	67	88	133
Demanda d_i	8	10	5	12	8	10	12	9	3	15

Tabela 1: *Dados do exemplo 1 testado.*

e uma diminuição na quantidade de diferentes padrões de corte, se comparado ao **modelo 1**. Comparando os modelos em um contexto de produção, significa que o **modelo 1** encontrou uma solução que gasta menos material, e o **modelo 2** uma solução que tem um menor custo de prepação.

Modelo	1	2
Total de objetos cortados	13	15
Quantidade de padrões de corte usados	9	6

Tabela 2: *Resultados obtidos em cada caso, para o exemplo teste 1.*

5.2 Modelo 3

O exemplo numérico 2 apresentado a seguir considera $L = 200$ o tamanho dos objetos em estoque. A Tabela 3 contém os demais dados do problema, e a Tabela 4 a configuração usada para os testes.

Item i	1	2	3	4	5	6	7
Tamanho ℓ_i	101	71	53	44	42	28	7
Demanda d_i	5	6	15	4	10	9	12

Tabela 3: *Dados do exemplo de teste 2.*

Parâmetro	Valor
cplex.epgap	10^{-2}
cplex.tilim	3600
ρ_1	10^{-5}
ρ_2	10^{-5}

Tabela 4: *Parâmetros usados para testar o exemplo 2.*

Resolvendo primeiramente a minimização de $f_1(\mathbf{x}) + \rho_2 f_2(\mathbf{x})$, referente ao problema (15), encontramos a solução $f_1(\mathbf{x}) = z_1^U = 14$ e $f_2(\mathbf{x}) = z_2^N = 7$, apresentada na Tabela 5.

x_j	Item i							PC j
	1	2	3	4	5	6	7	
5	1	1	0	0	0	1	0	1
1	0	1	2	0	0	0	3	2
4	0	0	2	1	1	0	1	3
1	0	0	2	0	2	0	1	4
1	0	0	1	0	3	0	3	5
1	0	0	1	0	2	2	1	6
1	0	0	1	0	0	5	1	7

Tabela 5: Solução obtida na minimização de $f_1(\mathbf{x}) + \rho_2 f_2(\mathbf{x})$.

Então, resolvendo a minimização de $\rho_1 f_1(\mathbf{x}) + f_2(\mathbf{x})$, referente ao problema (15), encontramos a solução $f_1(\mathbf{x}) = z_1^N = 25$ e $f_2(\mathbf{x}) = z_2^U = 2$, apresentada na Tabela 6.

x_j	Item i							PC j
	1	2	3	4	5	6	7	
15	1	0	1	0	0	1	2	1
10	0	1	0	1	1	1	2	2

Tabela 6: Solução obtida na minimização de $\rho_1 f_1(\mathbf{x}) + f_2(\mathbf{x})$.

Encontrando os pontos Utopia e Nadir das funções objetivos (pontos extremos da curva de Pareto-ótima) $(z_1^U, z_2^N) = (14, 7)$ e $(z_1^N, z_2^U) = (25, 2)$, resolvemos então o **modelo 3** (16) para $\omega \in (0, 1)$ com passo 0.1, obtendo os resultados apresentados a seguir nas Tabelas 7 a 12, e ilustrados na Figura 3.

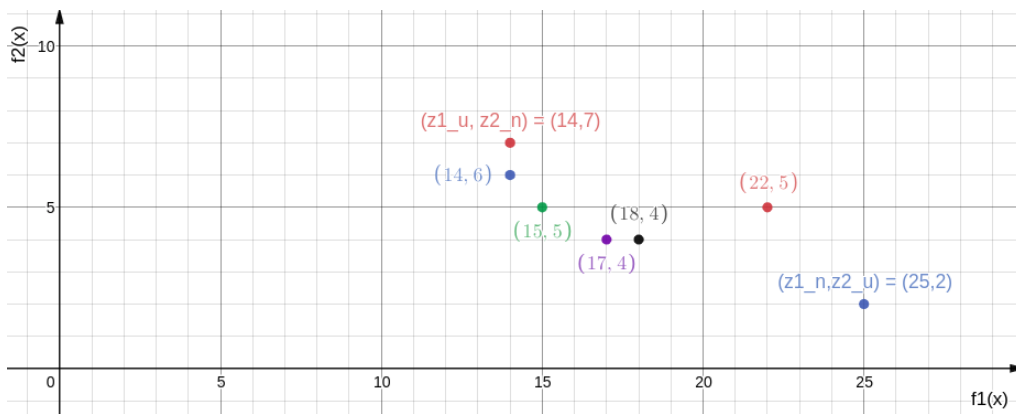


Figura 3: Soluções do exemplo 2 pelo método das Somas Ponderadas. Eixo x representa os valores de solução de $f_1(\mathbf{x})$, e o eixo y de $f_2(\mathbf{x})$.

		Item i							
x_j	1	2	3	4	5	6	7	PC j	
5	1	0	0	0	0	0	0	1	
6	0	1	0	0	0	0	0	2	
3	0	0	0	0	4	1	0	3	
2	0	0	0	4	0	0	3	4	
6	0	0	3	0	0	1	1	5	

Tabela 7: Solução para $\omega = 0.1$.

		Item i							
x_j	1	2	3	4	5	6	7	PC j	
5	1	0	0	0	0	0	0	1	
3	0	0	0	0	4	1	0	2	
6	0	0	3	0	0	1	1	3	
4	0	2	0	1	0	0	2	4	

Tabela 8: Solução para $\omega = 0.2$.

		Item i							
x_j	1	2	3	4	5	6	7	PC j	
3	0	0	0	0	4	1	0	1	
5	0	0	3	0	0	1	1	2	
4	0	2	0	1	0	0	2	3	
5	1	0	0	0	0	3	2	4	

Tabela 9: Solução para $\omega = 0.3$ e $\omega = 0.4$.

		Item i							
x_j	1	2	3	4	5	6	7	PC j	
6	0	0	3	0	0	1	1	1	
3	0	2	0	1	0	0	2	2	
5	1	0	1	0	1	0	0	3	
3	0	0	0	2	2	1	0	4	

Tabela 10: Solução para $\omega = 0.5$.

	Item i							
x_j	1	2	3	4	5	6	7	PC j
4	0	0	3	0	0	1	1	1
3	0	2	0	1	0	0	2	2
1	1	0	0	0	0	3	2	3
4	1	0	1	0	1	0	0	4
3	0	0	0	2	2	1	0	5

Tabela 11: *Solução para $\omega = 0.6$.*

	Item i							
x_j	1	2	3	4	5	6	7	PC j
1	0	0	0	0	4	1	0	1
4	0	0	3	0	0	1	1	2
3	0	2	0	1	0	0	2	3
1	1	0	0	0	0	3	2	4
4	1	0	1	0	1	0	0	5
1	0	0	0	2	2	1	0	6

Tabela 12: *Solução para $\omega = 0.7$, $\omega = 0.8$ e $\omega = 0.9$.*

5.3 Modelo 4

Utilizando os pontos Utopia e Nadir já calculados $(z_1^U, z_2^N) = (14, 7)$ e $(z_1^N, z_2^U) = (25, 2)$, resolvemos o problema (5) para $(r, s) = (1, 2)$ e para $(r, s) = (2, 1)$, com $\Delta = 1$. Os resultados encontrados estão expostos a seguir nas Tabelas 13 a 23, e a Figura 4 mostra as soluções obtidas pelo método.

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	1	0	0	0	1	0	1
1	0	1	2	0	0	0	3	2
4	0	0	2	1	1	0	1	3
1	0	0	2	0	2	0	1	4
1	0	0	2	0	0	2	5	5
2	0	0	1	0	2	2	1	6

Tabela 13: *Solução para $\varepsilon_2 = 6$.*

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	1	0	0	0	1	0	1
2	0	1	2	0	0	0	3	2
4	0	0	2	1	1	0	1	3
1	0	0	2	0	2	0	1	4
2	0	0	1	0	2	2	1	5
2	0	0	1	0	2	2	1	6

Tabela 14: Solução para $\varepsilon_2 = 5$.

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	1	0	0	0	1	0	1
2	0	1	2	0	0	0	3	2
4	0	0	2	1	1	0	1	3
3	0	0	1	0	2	2	1	4

Tabela 15: Solução para $\varepsilon_2 = 6$.

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	0	0	0	2	0	2	1
6	0	1	1	1	0	1	0	2
3	0	0	3	0	0	1	1	3

Tabela 16: Solução para $\varepsilon_2 = 3$.

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	0	0	1	0	0	4	1
15	0	1	1	0	1	1	0	2

Tabela 17: Solução para $\varepsilon_2 = 2$.

	Item i							
x_j	1	2	3	4	5	6	7	PC j
6	1	1	0	0	0	1	0	1
15	0	0	1	1	1	1	4	2

Tabela 18: Solução para $\varepsilon_1 = 24$.

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	0	0	1	0	1	3	1
15	0	1	1	0	1	1	0	2

Tabela 19: *Solução para $\varepsilon_1 = 20$.*

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	0	0	0	2	0	1	1
4	0	2	0	1	0	0	1	2
9	0	0	2	0	0	1	2	3

Tabela 20: *Solução para $\varepsilon_1 = 19$.*

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	0	0	0	2	0	1	1
6	0	1	0	2	0	1	1	2
6	0	0	3	0	0	1	1	3

Tabela 21: *Solução para $\varepsilon_1 = 17$.*

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	0	0	0	2	0	1	1
6	0	1	0	1	0	2	1	2
5	0	0	3	0	0	1	1	3

Tabela 22: *Solução para $\varepsilon_1 = 16$.*

	Item i							
x_j	1	2	3	4	5	6	7	PC j
5	1	0	1	0	0	1	1	1
6	0	1	1	0	1	1	0	2
4	0	0	1	1	1	0	2	3

Tabela 23: *Solução para $\varepsilon_1 = 15$.*

x_j	Item i							PC j
	1	2	3	4	5	6	7	
5	1	0	0	0	2	0	2	1
6	1	0	0	0	2	0	2	2
3	0	0	3	0	0	1	1	3

Tabela 24: Solução para $\varepsilon_1 = 14$.

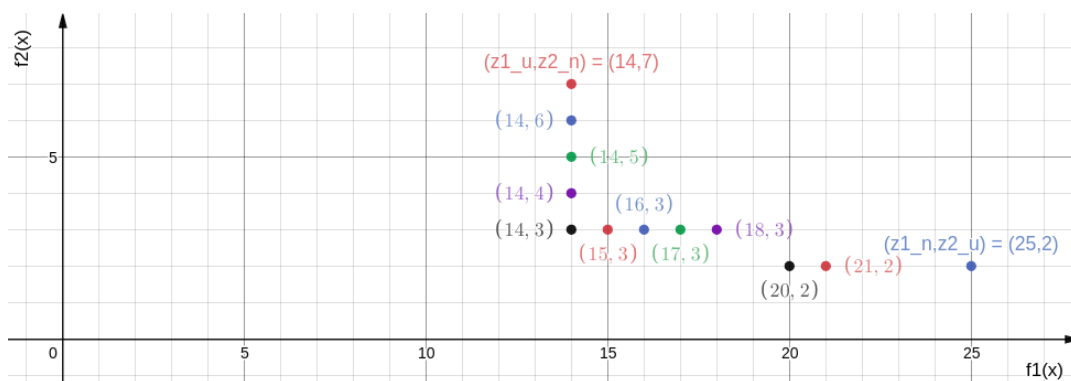


Figura 4: Soluções do exemplo 2 pelo método de ε -restrito. Eixo x representa os valores de solução de $f_1(\mathbf{x})$, e o eixo y de $f_2(\mathbf{x})$.

6 Conclusões

O projeto abordou, além de um panorama geral sobre PCE multiobjetivo, um estudo sobre alguns métodos de solução para este problema na sua versão biobjetiva. A partir disso, diferentes modelos matemáticos foram implementados utilizando o pacote de otimização *CPLEX*, e com isso foi possível compreender as vantagens e desvantagens de cada um dos métodos expostos na literatura.

O método de Geração de Colunas permite resolver o problema de maneira menos custosa computacionalmente, incrementando bons padrões de corte no problema ao longo das iterações. Já no método das Somas Ponderadas, o decisor tem controle sobre a importância de cada objetivo, além de utilizar uma normalização das funções visando um cálculo mais equilibrado. Por fim, o método do ε -restrito busca por diferentes soluções mantendo um objetivo fixo e controlando o limitante superior das outras funções transformando-as em restrições do problema.

Após o estudo e a implementação, testes computacionais foram realizados para validá-la, e os resultados obtidos foram apresentados e brevemente analisados.

Referências

- [1] S. A. D. ARAUJO, K. C. POLDI, AND J. SMITH, *A genetic algorithm for the one-dimensional cutting stock problem with setups*, Pesquisa Operacional, 34 (2014), pp. 165–187.

- [2] J. E. C. ARROYO ET AL., *Heurísticas e metaheurísticas para otimização combinatória multiobjetivo*, (2002).
- [3] J.-F. BÉRUBÉ, M. GENDREAU, AND J.-Y. POTVIN, *An exact ϵ -constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits*, *European journal of operational research*, 194 (2009), pp. 39–50.
- [4] Y. CUI, C. ZHONG, AND Y. YAO, *Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost*, *European Journal of Operational Research*, 243 (2015), pp. 540–546.
- [5] J. V. DE CARVALHO, *Exact solution of bin-packing problems using column generation and branch-and-bound*, *Annals of Operations Research*, 86 (1999), pp. 629–659.
- [6] ———, *Lp models for bin packing and cutting stock problems*, *European Journal of Operational Research*, 141 (2002), pp. 253–273.
- [7] M. EHRGOTT, *Multicriteria optimization*, vol. 491, Springer Science & Business Media, 2005.
- [8] H. FOERSTER AND G. WASCHER, *Pattern reduction in one-dimensional cutting stock problems*, *International Journal of Production Research*, 38 (2000), pp. 1657–1676.
- [9] P. C. GILMORE AND R. E. GOMORY, *A linear programming approach to the cutting-stock problem*, *Operations research*, 9 (1961), pp. 849–859.
- [10] ———, *A linear programming approach to the cutting stock problem—part II*, *Operations research*, 11 (1963), pp. 863–888.
- [11] O. GRODZEVICH AND O. ROMANKO, *Normalization and other topics in multi-objective optimization*, (2006).
- [12] H. A. KOBERSZTAJN, *Técnicas para minimização do número de padrões de corte no problema de corte de estoque*, (2015).
- [13] C. U. MANUAL, *IBM ILOG CPLEX optimization studio*, Version, 12 (1987), pp. 1987–2018.
- [14] K. C. POLDI, *Algumas extensões do problema de corte de estoque*, PhD thesis, Universidade de São Paulo, 2003.
- [15] P. R. SAMPAIO, *Teoria, métodos e aplicações de otimização multiobjetivo*, PhD thesis, Universidade de São Paulo, 2011.
- [16] F. VANDERBECK, *Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem*, *Operations Research*, 48 (2000), pp. 915–926.

- [17] H. H. YANASSE AND M. S. LIMEIRA, *A hybrid heuristic to reduce the number of different patterns in cutting stock problems*, *Computers & Operations Research*, 33 (2006), pp. 2744–2756.