OTIMIZAÇÃO MULTIOBJETIVO: TEORIA E APLICAÇÕES

Palavras-chave: Otimização multiobjetivo, Soma ponderada , $\varepsilon-$restrito

Matheus Queiroz Mota, IMECC - UNICAMP
Kelly Cristina Poldi, IMECC - UNICAMP

Introdução:

A otimização multiobjetivo envolve a otimização simultânea de várias funções, sujeitas a um conjunto de restrições formadas por equações e/ou inequações matemáticas. Devido aos conflitos entre essas funções (ou seja, otimizar uma função pode prejudicar o desempenho das outras), a solução "ótima"depende de qual objetivo é priorizado, pois, na maior parte das vezes, a solução que otimiza uma função específica não otimiza as demais. Portanto, existe um conjunto de soluções conhecido como soluções de Pareto [2].

Deste modo, o presente projeto visa estudar os conceitos de otimização multiobjetivo e alguns métodos exatos para obtenção de soluções para problemas de otimização multiobjetivo, implementando computacionalmente os métodos estudados; que são: o método da Soma Ponderada e o método $\varepsilon-$restrito. Para suas respectivas implementações foi utilizada da biblioteca PuLP, do Python [6]. Além disso, testamos essas implementações em aplicações específicas, como o Problema da Dieta, onde o objetivo é minimizar o custo e maximizar a proteína, e o Problema da Produção, onde buscamos maximizar o lucro e minimizar a emissão de CO2. Tais modelagens e soluções irão constar no relatório final.

Metodologia:

Problema de Otimização Multiobjetivo: De maneira geral, um Problema de Otimização Multiobjetivo (POM) é caracterizado por possuir mais de uma função objetivo, digamos p funções $(p > 1)$ que devem ser maximizadas, sujeitas a um conjunto de restrições matemáticas $\mathcal{X} \subseteq \mathbb{R}^n$: $\mathcal{X} \neq \varnothing$. Assim, o problema pode ser formulado da seguinte forma:

$$\begin{array}{ll} \max & \mathbf{z} = (f_1(\mathbf{x}), \cdots, f_p(\mathbf{x})) \\ \text{sujeito a} & x \in \mathcal{X} \end{array}$$

em que:

- $x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$ é vetor das variáveis de decisão;
- $z_i = f_i(\mathbf{x})$ é a $i-$ésima função objetivo a ser maximizada, $i = 1, \cdots, p$.

Assim, um Problema de Otimização Multiobjetivo (POM) busca otimizar (maximizar e/ou minimizar) várias funções objetivas dentro de determinadas restrições. Isso implica que não há uma solução única e perfeita, mas um conjunto de soluções ótimas chamadas soluções de Pareto [1]. Consequentemente, utilizamos métodos específicos para abordar problemas multiobjetivos. Os principais métodos explorados são a Soma Ponderada e o método ε-restrito, ambos denominados métodos de escalarização.

Método da Soma Ponderada: Esse método consiste em transformar um problema multiobjetivo em um problema mono-objetivo, através da formulação de uma função objetivo que é a soma ponderada das p funções originais. Para cada função f_k, é atribuído um peso $\lambda_k \geq 0$, de modo que, para ao menos um i, tenhamos $\lambda_i > 0$. Sendo assim, a formulação do problema pode ser descrita da seguinte forma:

$$\begin{aligned} &\max z = \sum_{k=1}^{p} \lambda_k f_k(x) \\ &\text{s.a} \begin{cases} x \in \mathcal{X} \end{cases} \end{aligned} \tag{1}$$

Método do $\varepsilon-$restrito: em um POM com p funções objetivas $(f_1(x), f_2(x), \ldots, f_p(x))$, selecionamos uma dessas funções (digamos $f_i(x)$) para otimização. As outras funções, $f_j(x)$ com $(j \neq i)$, são consideradas restrições no novo problema mono-objetivo, onde para cada j, tem-se $f_j(x) \geq \varepsilon_j$, de modo que cada ε_j é determinado pelo tomador de decisões do problema. Deste modo, tal problema pode ser modelado como:

$$\begin{aligned} &\max f_i(x) \\ &\text{s.a} \begin{cases} x \in X \\ f_k(x) \geq \varepsilon_k, \quad k = 1, ..., i-1, i+1, ..., p \end{cases} \end{aligned} \tag{2}$$

Vale ressaltar que ambos os métodos retornam soluções fracamente eficientes, e, sob certas condições, podem retornar soluções eficientes do problema original. Para mais detalhes, consulte as referências [3] e [4].

Implementação Computacional: Durante a pesquisa, desenvolvemos seis algoritmos computacionais. Estes incluem os algoritmos para o método da soma ponderada e o $\varepsilon-$restrito, além de duas variantes desses métodos que permitem ajustar e variar os parâmetros. Adicionalmente, criamos dois algoritmos, um para cada método exato, que suportam a leitura de arquivos .csv, facilitando a aplicação em problemas de grande escala.

Para a implementação, utilizamos a biblioteca PuLP [6], que nos ajudou a modelar os métodos e utilizar um solver para obter soluções factíveis. Além disso, utilizamos a biblioteca pandas [5] para ler os dados dos arquivos .csv. Todos os códigos estão disponíveis no (**GitHub**).

Pagina Web: Visando divulgar a parte teórica de forma mais leve e dinâmica, fizemos, por meio da biblioteca *streamlit* [7] do Python, uma página web que contém explicações sintetizadas dos principais conceitos da otimização multiobjetivo, os métodos exatos, bem como figuras e gráficos interativos. Para acessar página web, (**Clique Aqui**), caso a página esteja inativa, basta clicar na opção *"Yes, get this app back up!"*.

Aplicações

Vamos apresentar neste texto um exemplo prático da otimização multiobjetivo, escolhemos um exemplo de problema de transporte, que é apenas um dos vários casos que exploramos durante a Iniciação Científica. Outros exemplos, que abordam temas como dieta, produção e outros aspectos relevantes, são reportados no relatório final.

Considere dois armazéns (Armazéns 1 e 2) responsáveis por armazenar produtos alimentícios destinados a três mercados (Mercados A, B e C) de uma determinada empresa. Um dos objetivos da empresa é minimizar os custos de transporte de itens dos armazéns até os mercados. No entanto, visando emitir a quantidade mínima de CO2 por toda sua frota, a empresa estimou a produção de CO2, em média por valor unitário de produto transportado, de cada armazém para cada mercado.

Variáveis de decisão:

$$x_{mn} : \text{quantidade de mercadorias transportadas do Armazém } m \text{ para o Mercado } n.$$

Funções-objetivo: neste problema, a função custo é representada por $z_1(x)$ e a função que representa a quantidade de CO2 é $z_2(x)$. Os coeficientes correspondentes são dados por:

$$\min z_1(x) = 2x_{1A} + 4x_{1B} + 5x_{1C} + 3x_{2A} + 1x_{2B} + 2x_{2C} \quad \text{(Custo de transporte em R\$)}$$
$$\min z_2(x) = 9x_{1A} + 4x_{1B} + x_{1C} + 2x_{2A} + 5x_{2B} + 8x_{2C} \quad \text{(Emissão de CO2)}$$

Restrições:

- <u>Capacidade de armazenamento:</u> cada armazém possui uma capacidade máxima de produtos que podem ser estocados, ou seja, a quantidade de mercadorias que vai do armazém para todas as cidades não deve exceder a capacidade do armazém. Os respectivos valores de capacidade são dados a seguir:

$$x_{1A} + x_{1B} + x_{1C} \leq 100 \quad \text{(Capacidade do Armazém 1)}$$
$$x_{2A} + x_{2B} + x_{2C} \leq 150 \quad \text{(Capacidade do Armazém 2)}$$

- <u>Demanda:</u> cada mercado possui uma demanda mínima de produtos que deve ser respeitada, ou seja, a produção de cada armazém para a cidade deve ser maior ou igual a essa demanda mínima. Vale ressaltar que, nas restrições de demanda, estamos trabalhando com desigualdades do tipo "$\geq$", pois restrições do tipo "="podem ocasionar infactibilidade. As restrições são dadas a seguir:

$$x_{1A} + x_{2A} \geq 80 \quad \text{(Demanda do Mercado A)}$$
$$x_{1B} + x_{2B} \geq 70 \quad \text{(Demanda do Mercado B)}$$
$$x_{1C} + x_{2C} \geq 100 \quad \text{(Demanda do Mercado C)}$$

<u>Solução:</u> Por meio dos algoritmos desenvolvidos e disponíveis para conusulta no (**GitHub**), obtivemos a curva apresentada na Figura [1] representando o conflito das duas funções objetivos:

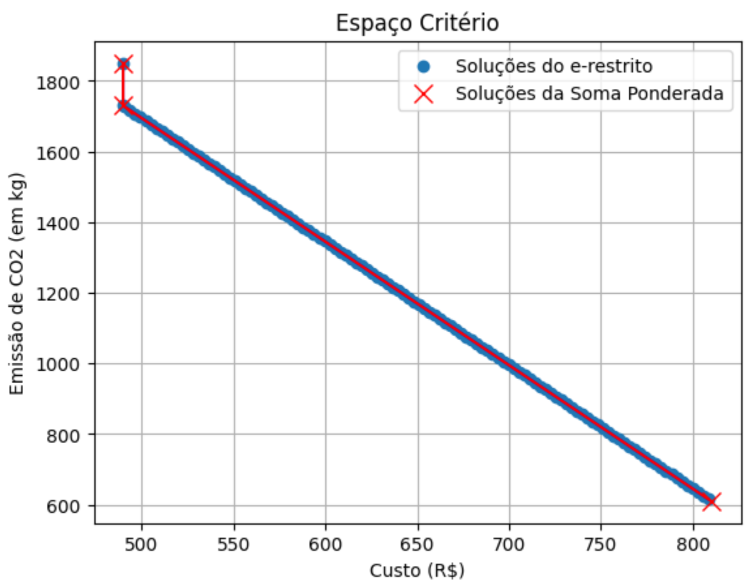

Figura 1: Gráfico da fronteira de Pareto do exemplo do transporte, para os métodos exatos.

Pode-se notar, na Figura [1], que ao diminuir o custo de transporte, aumentamos a quantidade de CO2 emitida.

Conclusão:

Em conclusão, este trabalho abordou tanto os aspectos teóricos quanto práticos da Otimização Multiobjetivo, explorando desde os fundamentos conceituais até a implementação computacional de métodos exatos.

Espera-se que as ferramentas desenvolvidas nesta pesquisa, incluindo os códigos, e a página web, possam não apenas auxiliar aqueles que estudam a área de otimização multiobjetivo, mas também despertar o interesse daqueles que desejam aprofundar-se neste campo da matemática.

Agradecimento

Os autores agradecem o apoio financeiro PIBIC/CNPq UNICAMP, cota 2023-2024.

Referências

[1] C. H. Antunes, M. J. Alves e J. N. Clímaco. *Multiobjective linear and integer programming*. Switzerland: Springer, 2016.

[2] V. Chankong e Y. Y. Haimes. *Multiobjective decision making: Theory and methodology*. New York: Elsevier Science Publishing Co, 1983.

[3] K. Deb. *Multiobjective optimization using evolutionary algorithms*. John Wiley & Sons, 2005.

[4] M. Ehrsgott. *Multicriteria optimization*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 2005.

[5] *Pandas Documentation*. URL: `https://pandas.pydata.org/pandas-docs/stable/` (acesso em 22/07/2024).

[6] Python Optimization Tools (PyOpt). *PuLP: LP Modelling in Python*. URL: `https://github.com/coin-or/pulp` (acesso em 19/07/2024).

[7] *Streamlit Documentation*. URL: `https://docs.streamlit.io/library/api-reference` (acesso em 19/07/2024).