

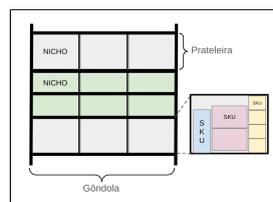
Um Modelo de Programação por Metas para Mensurar o Impacto de Nichos Compartilhados em Armazéns de *e-commerce*

Washington A. Oliveira¹
FCA/UNICAMP, Limeira, SP

O atual crescimento do comércio eletrônico (*e-commerce*) reflete na cadeia de suprimentos. Boas técnicas para a atribuição automatizada de locais de armazenamento de produtos, separação e coleta de pedidos têm contribuído para a gestão do *e-commerce*, incluindo diversos desafios como alto grau de personalização do produto. Plataformas tecnológicas de vendas são usadas pelas empresas para comunicação e monitoramento do comportamento de consumo [1]. A solução do *Storage Location Assignment Problem* (SLAP) surge como uma alternativa de gerenciamento, o qual envolve a alocação de produtos e a otimização do espaço disponível, inclui diversos parâmetros, o *layout* da área de armazenamento, características das prateleiras e produtos, chegada de produtos e variações da demanda [3]. O custo total de movimentação para a coleta/separação dos pedidos é item importante, pois representa a operação que mais consome tempo e mão de obra. Fig. 1(a) ilustra a complexidade de alocar diversos tipos de *Stock Keeping Units* (SKUs) em várias gôndolas. Fig. 1(b) ilustra alguns caminhos de alocação das SKUs nos nichos de uma gôndola.



(a) Layout. Fonte: www.frenet.com.br/blog



(b) Layout das gôndolas.

Figura 1: Representação da organização de um armazém de *e-commerce*. Fonte: dos autores.

Este artigo explora diferentes distribuições de SKUs nos nichos para melhorar o processo de coleta/separação dos pedidos e estuda o impacto global no espaço utilizado quando um mesmo nicho recebe mais de um SKU (compartilhamento), o que é interessante do ponto de vista da coleta das ordens, pois a maior dispersão dos SKUs em diferentes gôndolas tem potencial de produzir rotas de coleta mais eficientes e distribuir melhor o fluxo dos carros coletores. A volumetria, o compartilhamento e utilização de nichos são objetivos conflitantes na alocação de SKUs. A metodologia de programação por metas (PM) é uma ferramenta para lidar com objetivos conflitantes [2]. Ela envolve a minimização de variáveis de desvio indesejadas, que precisam ser combinadas por meio de uma função de realização de minimização para encontrar uma solução que seja “o mais próxima possível” dos alvos almeçados, modelando objetivos conflitantes por meio da compensação e ativação de alvos (*targets*). O modelo proposto envolve: p_{ij} é a quantidade de SKU i que cabe no nicho j ; v_i , D_i , w_i , c_i e h_i são o volume, demanda, peso, comprimento e altura do SKU i , resp.; M_i é a quantidade mínima de nichos que deve conter o SKU i ; C_j e H_j é o comprimento total e a altura máxima do nicho j , resp.; V_j é o volume total (alvo) do nicho j ; S_j é a quantidade de SKUs (alvo) que podem ser alocadas no nicho j ; N é a quantidade total (alvo) de nichos disponíveis; se NS é um subconjunto de nichos que pode receber produtos mais pesados, então W_{\max} é o maior peso permitido nos nichos selecionados; x_{ij} é a variável binária igual a 1 se o SKU i é alocado no nicho j ; y_j é a variável binária igual a 1 se o nicho j é usado; e n_j^a , n_j^b , n^c , p_j^a , p_j^b e p^c são as variáveis de desvio negativo e positivo em relação aos alvos, resp.; Segue a proposta de modelo para o SLAP.

¹waoliv@unicamp.br

$$\text{Minimizar} \quad \sum_j \left(\frac{n_j^a}{V_j} + \frac{n_j^b + p_j^b}{S_j} \right) + \frac{p^c}{0,8N} \quad (1)$$

$$\text{sujeito a} \quad \sum_i v_i x_{ij} + n_j^a - p_j^a = V_j \quad \forall j, \quad (2)$$

$$\sum_i x_{ij} + n_j^b - p_j^b = S_j \quad \forall j, \quad (3)$$

$$\sum_j \sum_i y_j + n^c - p^c = 0,8N, \quad (4)$$

$$\text{(a)} \sum_j p_{ij} x_{ij} \geq D_i \quad \forall i, \quad \text{(b)} \sum_j x_{ij} \geq M_i \quad \forall i, \quad \text{(c)} \sum_i x_{ij} \leq S_j y_j \quad \forall j, \quad (5)$$

$$\text{(a)} w_i x_{ij} \leq W_{\max} \quad \forall i, \forall j \in \{NS\}, \quad \text{(b)} \sum_i c_i x_{ij} \leq C_j \quad \forall j, \quad \text{(c)} h_i x_{ij} \leq H_j \quad \forall i, \forall j, \quad (6)$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall i, \quad \forall j. \quad (7)$$

A função de realização (1) do modelo minimiza o desvio relativo entre o conjunto de alvos. (2), (3) e (4) são os objetivos conflitantes para alcançar seus respectivos alvos, definidas no sentido “mais é melhor”, “igual é melhor” e “menos é melhor”, resp. As variáveis de desvio indesejáveis estão destacadas, normalizadas pelos alvos e minimizadas na função de realização. A demanda e a dispersão mínima de SKUs são atendidas em (5)(a) e (5)(b), resp. As variáveis são conectadas por (5)(c), limitando a quantidade de SKUs distintos por nicho. As restrições (6)(a)–(c) são de factibilidade das dimensões dos SKUs em cada nicho. Por fim, (7) fornece o domínio das variáveis.

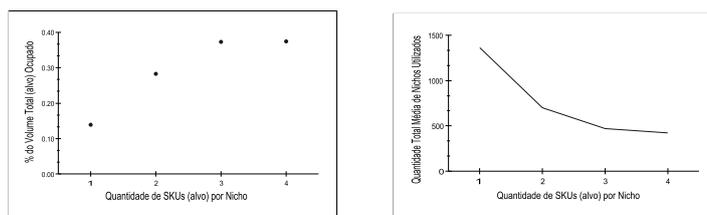


Figura 2: Impacto do aumento do número de SKUs/nicho (alvo na igualdade). Fonte: dos autores.

O modelo foi implementado no pacote JuMP do Julia (v1.10) e resolvido no Gurobi (v10.01). As exemplares foram retirados do banco de dados Olist, incluindo dados de produtos e pedidos de um *e-commerce* brasileiro. 3600s foi o tempo limite para cada teste. O conflito na Fig. 2(a) mostra que aumentar o número de SKUs em cada nicho melhora a volumetria global (esse comportamento limita-se 3 SKUs/nicho). Comportamento esperado uma vez que o número de SKUs por nicho é também limitado por suas dimensões. O conflito na Fig. 2(b) mostra que aumentar o número de SKUs por nicho reduz o número médio de nichos utilizados, o que melhora a utilização do armazém.

Agradecimentos

Ao financiamento do CNPq (nº 309925/2021-5) e FAPESP (nºs 2020/09838-0 e 2022/05803-3).

Referências

- [1] J. Jedrzejczak-Gas, A. Barska e M. Siničáková. “Level of development of e-commerce in EU countries”. Em: **Management** 23.1 (2019), pp. 209–224. DOI: 10.2478/manment-2019-0012.
- [2] D. Jones e M. Tamiz. **Practical goal programming**. Vol. 141. New York: Springer, 2010.
- [3] J. J. R. Reyes, E. L. Solano-Charris e J. R. Montoya-Torres. “The storage location assignment problem: A literature review”. Em: **Int. J. of Industrial Engineering Computations** 10.2 (2019), pp. 199–224. DOI: 10.5267/j.ijiec.2018.8.001.